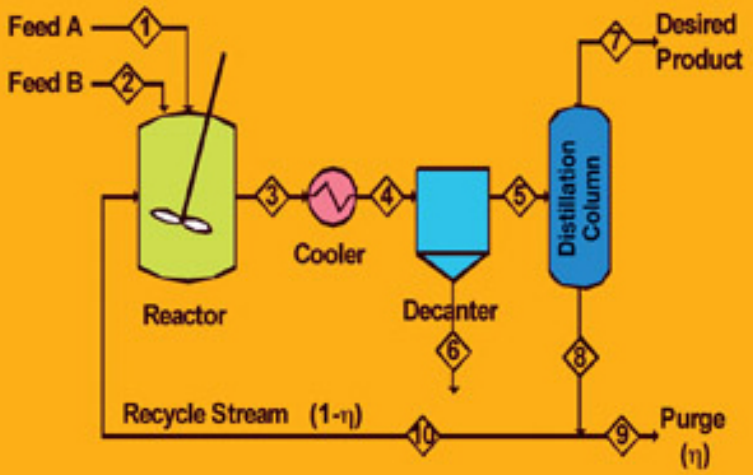
Advances in Process Systems Engineering – Vol. 1

MULTI-OBJECTIVE OPTIMIZATION

Techniques and Applications in Chemical Engineering



Gade Pandu Rangaiah

editor

World Scientific

Advances in Process Systems Engineering – Vol. 1

MULTI-OBJECTIVE OPTIMIZATION

Techniques and Applications in Chemical Engineering

Advances in Process Systems Engineering

Series Editor: Gade Pandu Rangaiah
(National University of Singapore)

Vol. 1: Multi-Objective Optimization:
Techniques and Applications in Chemical Engineering
ed: Gade Pandu Rangaiah

Advances in Process Systems Engineering – Vol. 1

MULTI-OBJECTIVE OPTIMIZATION

Techniques and Applications in Chemical Engineering

editor

Gade Pandu Rangaiah

National University of Singapore

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

Advances in Process Systems Engineering — Vol. 1
MULTI-OBJECTIVE OPTIMIZATION
Techniques and Applications in Chemical Engineering

Copyright © 2009 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the Publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

ISBN-13 978-981-283-651-9
ISBN-10 981-283-651-9

Desk Editor: Tjan Kwang Wei

Printed in Singapore.

Preface

Optimization is essential for reducing material and energy requirements as well as the harmful environmental impact of chemical processes. It leads to better design and operation of chemical processes as well as to sustainable processes. Many applications of optimization involve several objectives, some of which are conflicting. Multi-objective optimization (MOO) is required to solve the resulting problems in these applications. Hence MOO has attracted the attention of several researchers, particularly in the last ten years.

It is my pleasure and honor to edit this first book on MOO with focus on chemical engineering applications. Although process modeling and optimization has been my research interest since my doctoral studies around 1980, my interest and research in MOO began in 1998 when Prof. S.K. Gupta, Prof. A.K. Ray and I initiated collaborative work on the optimization of a steam reformer. Since then, we have studied optimization of many industrial reactors and processes that need to meet multiple objectives. I am thankful to both Prof. S.K. Gupta and Prof. A.K. Ray for the successful collaboration over the years.

The first chapter of the book provides an introduction to MOO with a realistic application, namely, the alkylation process optimization for two objectives. The second chapter reviews nearly 100 chemical engineering applications of MOO since the year 2000 to mid-2007. The next 5 chapters are on the selected MOO techniques; they include (1) review of multi-objective evolutionary algorithms in the context of chemical engineering, (2) multi-objective genetic algorithm and simulated

annealing as well as their jumping gene adaptations, (3) surrogate-assisted multi-objective evolutionary algorithm, (4) interactive MOO in process design, and (5) two methods for ranking the Pareto solutions.

The final 6 chapters present a broad range of MOO applications in chemical engineering including a few in biochemical engineering. They cover gas-phase refrigeration systems for liquefied natural gas, feed optimization to a residue catalytic cracker in a petroleum refinery, process design for multiple economic and/or environmental objectives, emergency response optimization around chemical plants, developing gene networks from gene expression data, and multi-product microbial cell factory. In these applications, the models employed are detailed, and the scenarios and data are realistic.

Each of the chapters in the book was contributed by leading researchers in MOO techniques and/or its applications. Brief resume and photo of each of the contributors to the book, are provided on the enclosed CD-ROM. Each chapter in the book was reviewed anonymously by at least two experts and/or other contributors. Of the submissions received, only those considered to be useful for education and/or research were revised by the respective contributor(s), and the revised submission was finally reviewed for presentation style by the editor or one of the other contributors. I am grateful to my colleague, Dr. S. Lakshminarayanan, who coordinated the anonymous review of chapters co-authored by me and also provided constructive comments on the first chapter.

The book will be useful to researchers in academic and research institutions, to engineers and managers in process industries, and to graduates and senior-level undergraduates. Researchers and engineers can use it for applying MOO to their processes whereas students can utilize it as a supplementary text in optimization courses. Each of the chapters in the book can be read and understood with little reference to other chapters. However, readers are encouraged to go through the introduction chapter first. Many chapters contain several exercises at the end, which can be used for assignments and projects. Some of these and the applications discussed within the chapters can be used as projects in

optimization courses at both undergraduate and postgraduate levels. The book comes with a CD-ROM containing many programs and files, which will be helpful to readers in solving the exercises and/or doing the projects.

I am thankful to all contributors to this book and anonymous reviewers for their collaboration and cooperation. In particular, I am grateful to Prof. S.K. Gupta and Prof. J. Thibault for several suggestions, which enhanced the book. Thanks are also due to Ms. H.L. Gow and Mr. K.W. Tjan from the World Scientific, for their suggestions and cooperation in preparing this book. It is my pleasure to acknowledge the contributions of my research fellow (Dr. A. Tarafder) and postgraduate students (J.K. Rajesh, P.P. Oh, B.S. Mohanakkannan, Y. Li, Y.M. Lee, N. Bhutani, N. Agrawal, F.C. Lee, Masuduzzaman, E.S.Q. Lee and N.M. Shah), to our studies on MOO applications in chemical engineering over the years and thus to this book in some way or other. I thank the Department of Chemical & Biomolecular Engineering and the National University of Singapore for encouraging and supporting my research over the years by providing ample resources including research scholarships.

Finally, and very importantly, I am grateful to my wife (Krishna Kumari) and daughters (Jyotsna and Madhavi) for their loving support, encouragement and understanding not only in preparing this book but in everything I pursue.

This page intentionally left blank

Contents

Preface v

Chapter 1 Introduction 1
Gade Pandu Rangaiah

1.1 Process Optimization 1
1.2 Multi-Objective Optimization: Basics 4
1.3 Multi-Objective Optimization: Methods 8
1.4 Alkylation Process Optimization for Two Objectives 13
1.4.1 Alkylation Process and its Model 13
1.4.2 Multi-Objective Optimization Results and Discussion 16
1.5 Scope and Organization of the Book 18
References 23
Exercises 25

Chapter 2 Multi-Objective Optimization Applications in Chemical Engineering 27
Masuduzzaman and Gade Pandu Rangaiah

2.1 Introduction 28
2.2 Process Design and Operation 29
2.3 Biotechnology and Food Industry 30
2.4 Petroleum Refining and Petrochemicals 40
2.5 Pharmaceuticals and Other Products/Processes 41
2.6 Polymerization 48
2.7 Conclusions 48
References 52

Chapter 3 Multi-Objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering 61
Antonio López Jaimes and Carlos A. Coello Coello

3.1 Introduction 61
3.2 Basic Concepts 62
3.2.1 Pareto Optimality 63
3.3 The Early Days 63
3.4 Modern MOEAs 65
3.5 MOEAs in Chemical Engineering 68
3.6 MOEAs Originated in Chemical Engineering 68
3.6.1 Neighborhood and Archived Genetic Algorithm 69
3.6.2 Criterion Selection MOEAs 70
3.6.3 The Jumping Gene Operator 72
3.6.4 Multi-Objective Differential Evolution 73
3.7 Some Applications Using Well-Known MOEAs 75
3.7.1 TYPE I: Optimization of an Industrial Nylon 6 Semi-Batch Reactor 75
3.7.2 TYPE I: Optimization of an Industrial Ethylene Reactor 76
3.7.3 TYPE II: Optimization of an Industrial Styrene Reactor 76
3.7.4 TYPE II: Optimization of an Industrial Hydrocracking Unit 77
3.7.5 TYPE III: Optimization of Semi-Batch Reactive Crystallization Process 78
3.7.6 TYPE III: Optimization of Simulated Moving Bed Process 79
3.7.7 TYPE IV: Biological and Bioinformatics Problems 80
3.7.8 TYPE V: Optimization of a Waste Incineration Plant 81
3.7.9 TYPE V: Chemical Process Systems Modelling 81
3.8 Critical Remarks 83
3.9 Additional Resources 84
3.10 Future Research 84
3.11 Conclusions 85
Acknowledgements 85
References 86

Chapter 4 Multi-Objective Genetic Algorithm and Simulated Annealing with the Jumping Gene Adaptations 91
Manojkumar Ramteke and Santosh K. Gupta

4.1 Introduction 92
4.2 Genetic Algorithm (GA) 93
4.2.1 Simple GA (SGA) for Single-Objective Problems 93
4.2.2 Multi-Objective Elitist Non-Dominated Sorting GA (NSGA-II) and its JG Adaptations 99
4.3 Simulated Annealing (SA) 106
4.3.1 Simple Simulated Annealing (SSA) for Single-Objective Problems 106
4.3.2 Multi-Objective Simulated Annealing (MOSA) 107
4.4 Application of the Jumping Gene Adaptations of NSGA-II and MOSA to Three Benchmark Problems 108
4.5 Results and Discussion (Metrics for the Comparison of Results) 110
4.6 Some Recent Chemical Engineering Applications Using the JG Adaptations of NSGA-II and MOSA 119
4.7 Conclusions 120
Acknowledgements 120
Appendix 121
Nomenclature 126
References 127
Exercises 129

Chapter 5 Surrogate Assisted Evolutionary Algorithm for Multi-Objective Optimization 131
Tapabrata Ray, Amitay Isaacs and Warren Smith

5.1 Introduction 132
5.2 Surrogate Assisted Evolutionary Algorithm 134
5.2.1 Initialization 135
5.2.2 Actual Solution Archive 136
5.2.3 Selection 136
5.2.4 Crossover and Mutation 136

5.2.5 Ranking 137
5.2.6 Reduction 137
5.2.7 Building Surrogates 138
5.2.8 Evaluation using Surrogates 140
5.2.9 k-Means Clustering Algorithm 140
5.3 Numerical Examples 141
5.3.1 Zitzler-Deb-Thiele's (ZDT) Test Problems 142
5.3.2 Osyczka and Kundu (OSY) Test Problem 145
5.3.3 Tanaka (TNK) Test Problem 146
5.3.4 Alkylation Process Optimization 146
5.4 Conclusions 147
References 148
Exercises 150

Chapter 6 Why Use Interactive Multi-Objective Optimization in Chemical Process Design? 153
Kaisa Miettinen and Jussi Hakanen

6.1 Introduction 154
6.2 Concepts, Basic Methods and Some Shortcomings 155
6.2.1 Concepts 155
6.2.2 Some Basic Methods 158
6.3 Interactive Multi-Objective Optimization 161
6.3.1 Reference Point Approaches 163
6.3.2 Classification-Based Methods 164
6.3.3 Some Other Interactive Methods 170
6.4 Interactive Approaches in Chemical Process Design 171
6.5 Applications of Interactive Approaches 171
6.5.1 Simulated Moving Bed Processes 172
6.5.2 Water Allocation Problem 176
6.5.3 Heat Recovery System Design 178
6.6 Conclusions 181
References 182
Exercises 187

Chapter 7 Net Flow and Rough Sets: Two Methods for Ranking the Pareto Domain 189
Jules Thibault

7.1 Introduction 190
7.2 Problem Formulation and Solution Procedure 193
7.3 Net Flow Method 196
7.4 Rough Set Method 203
7.5 Application: Production of Gluconic Acid 211
7.5.1 Definition of the Case Study 211
7.5.2 Net Flow Method 213
7.5.3 Rough Set Method 220
7.6 Conclusions 230
Acknowledgements 231
Nomenclature 231
References 232
Exercises 235

Chapter 8 Multi-Objective Optimization of Multi-Stage Gas-Phase Refrigeration Systems 237
Nipen M. Shah, Gade Pandu Rangaiah and Andrew F. A. Hoadley

8.1 Introduction 238
8.2 Multi-Stage Gas-Phase Refrigeration Processes 241
8.2.1 Gas-Phase Refrigeration 241
8.2.2 Dual Independent Expander Refrigeration Process for LNG 243
8.2.3 Significance of ΔT_{min} 245
8.3 Multi-Objective Optimization 246
8.4 Case Studies 247
8.4.1 Nitrogen Cooling using N_2 Refrigerant 248
8.4.2 Liquefaction of Natural Gas using the Dual Independent Expander Process 256
8.4.3 Discussion 267
8.5 Conclusions 267

Acknowledgements 269
Nomenclature 269
References 270
Exercises 271

Chapter 9 Feed Optimization for Fluidized Catalytic Cracking using a Multi-Objective Evolutionary Algorithm 277
Kay Chen Tan, Ko Poh Phang and Ying Jie Yang

9.1 Introduction 278
9.2 Feed Optimization for Fluidized Catalytic Cracking 279
9.2.1 Process Description 279
9.2.2 Challenges in the Feed Optimization 282
9.2.3 The Mathematical Model of FCC Feed Optimization 283
9.3 Evolutionary Multi-Objective Optimization 284
9.4 Experimental Results 288
9.5 Decision Making and Economic Evaluation 292
9.5.1 Fuel Gas Consumption of Reactor 72CC 293
9.5.2 High Pressure (HP) Steam Consumption of Reactor 72CC 295
9.5.3 Rate of Exothermic Reaction or Energy Gain 296
9.5.4 Summary of the Cost Analysis 297
9.6 Conclusions 298
References 298

Chapter 10 Optimal Design of Chemical Processes for Multiple Economic and Environmental Objectives 301
Elaine Su-Qin Lee, Gade Pandu Rangaiah and Naveen Agrawal

10.1 Introduction 302
10.2 Williams-Otto Process Optimization for Multiple Economic Objectives 304
10.2.1 Process Model 305
10.2.2 Objectives for Optimization 308
10.2.3 Multi-Objective Optimization 309

10.3 LDPE Plant Optimization for Multiple Economic Objectives 314
10.3.1 Process Model and Objectives 314
10.3.2 Multi-Objective Optimization 317
10.4 Optimizing an Industrial Ecosystem for Economic and Environmental Objectives 320
10.4.1 Model of an IE with Six Plants 322
10.4.2 Objectives, Results and Discussion 325
10.5 Conclusions 334
Nomenclature 335
References 335
Exercises 336

Chapter 11 Multi-Objective Emergency Response Optimization Around Chemical Plants 339
Paraskevi S. Georgiadou, Ioannis A. Papazoglou, Chris T. Kiranoudis and Nikolaos C. Markatos

11.1 Introduction 340
11.2 Multi-Objective Emergency Response Optimization 342
11.2.1 Decision Space 342
11.2.2 Consequence Space 343
11.2.3 Determination of the Pareto Optimal Set of Solutions 343
11.2.4 General Structure of the Model 345
11.3 Consequence Assessment 345
11.3.1 Assessment of the Health Consequences on the Population 345
11.3.2 Socioeconomic Impacts 349
11.4 A MOEA for the Emergency Response Optimization 349
11.4.1 Representation of the Problem 349
11.4.2 General Structure of the MOEA 349
11.4.3 Initialization 350
11.4.4 “Fitness” Assignment 350
11.4.5 Environmental Selection 352
11.4.6 Termination 352
11.4.7 Mating Selection 352
11.4.8 Variation 353

11.5 Case Studies 353
11.6 Conclusions 358
Acknowledgements 359
References 359

Chapter 12 Array Informatics using Multi-Objective Genetic Algorithms: From Gene Expressions to Gene Networks 363
Sanjeev Garg

12.1 Introduction 364
12.1.1 Biological Background 364
12.1.2 Interpreting the Scanned Image 367
12.1.3 Preprocessing of Microarray Data 368
12.2 Gene Expression Profiling and Gene Network Analysis 369
12.2.1 Gene Expression Profiling 370
12.2.2 Gene Network Analysis 371
12.2.3 Need for Newer Techniques? 377
12.3 Role of Multi-Objective Optimization 378
12.3.1 Model for Gene Expression Profiling 378
12.3.2 Implementation Details 380
12.3.3 Seed Population based NSGA-II 381
12.3.4 Model for Gene Network Analysis 382
12.4 Results and Discussion 386
12.5 Conclusions 395
Acknowledgments 396
References 396

Chapter 13 Optimization of a Multi-Product Microbial Cell Factory for Multiple Objectives – A Paradigm for Metabolic Pathway Recipe 401
Fook Choon Lee, Gade Pandu Rangaiah and Dong-Yup Lee

13.1 Introduction 402
13.2 Central Carbon Metabolism of *Escherichia coli* 405

13.3 Formulation of the MOO Problem 408
13.4 Procedure used for Solving the MIMOO Problem 410
13.5 Optimization of Gene Knockouts 413
13.6 Optimization of Gene Manipulation 415
13.7 Conclusions 422
Nomenclature 424
References 426

Index 429

This page intentionally left blank

Chapter 1

Introduction

Gade Pandu Rangaiah*
Department of Chemical & Biomolecular Engineering
National University of Singapore,
Engineering Drive 4, Singapore 117576

*Corresponding Author; e-mail: chegpr@nus.edu.sg

1.1 Process Optimization

Optimization refers to finding the values of decision (or free) variables, which correspond to and provide the maximum or minimum of one or more desired objectives. It is ubiquitous in daily life - people use optimization, often without actually realizing, for simple things such as traveling from one place to another and time management, as well as for major decisions such as finding the best combination of study, job and investment. Similarly, optimization finds many applications in engineering, science, business, economics, etc. except that, in these applications, quantitative models and methods are employed unlike qualitative assessment of choices in daily life. Without optimization of design and operations, manufacturing and engineering activities will not be as efficient as they are now. Even then, scope still exists for optimizing the current industrial operations, particularly with the ever changing economic, energy and environmental landscape.

Optimization has many applications in chemical, mineral processing, oil and gas, petroleum, pharmaceuticals and related industries. Not surprisingly, it has attracted the interest and attention from many chemical engineers in both the academia and industry for several decades. Optimization of chemical and related processes requires a mathematical model that describes and predicts the process behavior.

Process modeling and optimization along with control characterizes the area of process systems engineering (PSE), important in chemical engineering with a wide range of applications. The significant role of optimization in chemical engineering and contributions of chemical engineers to the field can be seen from the many books written by chemical engineering academicians (e.g., Lapidus and Luus, 1967; Beveridge and Schechter, 1970; Himmelblau, 1972; Ray and Szekely, 1973; Floudas, 1995 and 1999; Luus, 2000; Edgar *et al.*, 2001; Tawarmalani and Sahinidis, 2002; Diwekar, 2003; Reklaitis *et al.*, 2006). Besides these books devoted entirely to optimization, several books on process design cover optimization too (e.g., Biegler *et al.*, 1997; Peters *et al.*, 2003; Seider *et al.*, 2003).

The main focus of optimization of chemical processes so far has been optimization for one objective at a time (i.e., single objective optimization, SOO). However, practical applications involve several objectives to be considered simultaneously. These objectives can include capital cost/investment, operating cost, profit, payback period, selectivity, quality and/or recovery of the product, conversion, energy required, efficiency, process safety and/or complexity, operation time, robustness, etc. A few of these will be relevant for a particular application; for example, see Chapter 2 for the objectives (typically 2 to 4) used in each of the numerous chemical engineering applications summarized.

The appropriate objectives for a particular application are often conflicting, which means achieving the optimum for one objective requires some compromise on one or more other objectives. Some examples of sets of conflicting objectives are: capital cost and operating cost, selectivity and conversion, quality and conversion, profit and environmental impact, and profit and safety cost. These conflicting objectives can be handled by combining them suitably into one objective. A classic example of this practice is the use of total annual cost which includes operating cost and certain fraction of capital cost. The latter depends on plant life, expected return on investment and maintenance cost. Although the fraction of capital cost to be included in the total annual cost can be estimated, will it not be better to have a range of optimal solutions with varying capital and operating costs? Managers and engineers will then be able to choose one of the optimal solutions with the full knowledge on the variation of conflicting objectives besides

their own experience and other considerations which could not be included in the optimization problem.

Multi-objective optimization (MOO), also known as multi-criteria optimization, particularly outside engineering, refers to finding values of decision variables which correspond to and provide the optimum of *more than one objective*. Unlike in SOO which gives a unique solution (or several multiple optima such as local and global optima in case of non-convex problems), there will be many optimal solutions for a multi-objective problem; the exception is when the objectives are not conflicting in which case only one unique solution is expected. Hence, MOO involves special methods for considering more than one objective and analyzing the results obtained.

The relevance and importance of MOO in chemical engineering is increasing, which has been partly motivated by the availability of new and effective methods for solving multi-objective problems as well as increased computational resources. The study of Bhaskar *et al.* (2000) shows that there were around 30 journal publications on applications of MOO in chemical engineering before year 2000 (i.e., excluding those published in 2000). On the other hand, as will be shown in Chapter 2, nearly 100 MOO applications in chemical engineering have been studied and reported in more than 130 journals from the year 2000 to mid 2007. Another evidence of increasing interest and importance of MOO in chemical engineering is the inclusion of a chapter on MOO in the recent book on optimization by Diwekar (2003). Further, there have been several books on MOO outside the chemical engineering discipline (e.g., Cohon, 1978; Hwang and Masud, 1979; Chankong and Haimes, 1983; Sawaragi *et al.*, 1985; Stadler, 1988; Haimes *et al.*, 1990; Miettinen, 1999; Deb, 2001; Coello Coello *et al.*, 2002; Tan *et al.*, 2005). Understandably, their scope does not specifically include chemical engineering applications. The present book, that you are reading now, is the first book entirely devoted to MOO techniques and applications in chemical engineering.

The rest of this chapter is organized as follows. The next two sections respectively cover basics and methods for MOO. In the fourth section, optimization of a typical process for multiple objectives is described. The scope and organization of this book that includes an outline of individual chapters are presented in the last section.

1.2 Multi-Objective Optimization: Basics

In general, a MOO problem will have two or more objectives involving many decision variables and constraints. For illustration, consider an MOO problem with two objectives: $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, and several decision variables ($\mathbf{x}$). Such problems are known as two- or bi-objective optimization problems.

$$\begin{aligned} &\text{Minimize} && f_1(\mathbf{x}) && (1.1a)\\ &\text{Minimize} && f_2(\mathbf{x}) && (1.1b)\\ &\text{With respect to} && \mathbf{x} && \\ &\text{Subject to} && \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U && (1.1c)\\ & && h(\mathbf{x}) = \mathbf{0} && (1.1d)\\ & && g(\mathbf{x}) \le \mathbf{0} && (1.1e) \end{aligned}$$

Some applications may involve maximization of one or more objectives, which can be re-formulated by multiplying by -1 or taking the reciprocal (while ensuring that the denominator does not become zero) as the objective to be minimized. Hence, the above problem with two objectives to be minimized can be used for discussion without loss of generality.

The decision variables can either be all continuous within the respective lower and upper bounds ($\mathbf{x}^L$ and $\mathbf{x}^U$) or a mixture of continuous, binary (i.e., 0 or 1) and integer variables. In chemical engineering applications, the equality constraints, $h(\mathbf{x}) = 0$ arise from mass, energy and momentum balances - these can be algebraic and/or differential equations. The inequality constraints, $g(\mathbf{x})$ are due to equipment, material, safety and other considerations. Examples of inequality constraints are the requirement that the temperature in a reactor should be below a specified value to avoid reaction run-away, failure of the material used for equipment fabrication, undesirable side products and so on and so forth. The number of equality and inequality constraints can be none, a few or many depending on the application. The feasible region will be a multi-dimensional space satisfying bounds on variables, equality and inequality constraints. Besides $\mathbf{x}$, $f(\mathbf{x})$, $h(\mathbf{x})$ and $g(\mathbf{x})$ contain constants and/or parameters whose values are known. Note that MOO can also be used for estimating parameters in models – here, parameters are the decision variables.

The two objectives, $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are often conflicting. In such situations, there will be many optimal solutions to the MOO problem in equation 1.1. All these solutions are equally good in the sense that each

one of them is better than the rest in at least one objective. This implies that one objective improves while at least another objective becomes worse when one moves from one optimal solution to another. The solutions of an MOO problem are known as the Pareto-optimal solutions or, less commonly, Edgeworth-Pareto optimal solutions after the two economists, Edgeworth and Pareto, who developed the theory of indifference curves in the late 19th century. In the published literature, they are also referred to as non-dominated, non-inferior, efficient or simply Pareto solutions.

Definition: The set: $\mathbf{x}^P$, $f_1(\mathbf{x}^P)$ and $f_2(\mathbf{x}^P)$ is said to be a *Pareto-optimal solution* for the two-objective problem in equation 1.1, if and only if, no other feasible $\mathbf{x}$ exists such that $f_1(\mathbf{x}) \leq f_1(\mathbf{x}^P)$ and $f_2(\mathbf{x}) \leq f_2(\mathbf{x}^P)$ with strict inequality valid for at least one objective.

Pareto-optimal solutions can be represented in two spaces – objective space (e.g., $f_1(\mathbf{x})$ versus $f_2(\mathbf{x})$) and decision variable space. Definitions, techniques and discussions in MOO mainly focus on the objective space. However, implementation of the selected Pareto-optimal solution will require some consideration of the decision variable values. Multiple solution sets in the decision variable space may give the same or comparable objectives in the objective space; in such cases, the engineer can choose the most desirable solution in the decision variable space. See Tarafder *et al.* (2007) for a study on finding multiple solution sets in MOO of chemical processes.

The Pareto-optimal solutions of an MOO problem (namely, optimization of the classical Williams-Otto process for minimizing payback period, PBP, and maximizing the net present worth, NPW), described in detail in Chapter 10, are shown in Figure 1.1. For the present, in Figure 1.1, only the values of objectives and two decision variables (reactor temperature, T and reactor volume, V) are shown. In this figure, the first plot depicts the objective space and the other plots show the decision variables versus one objective. It is clear that the two objectives are conflicting since PBP increases with NPW. Further, optimal results in Figure 1.1a indicate that PBP increases gradually until NPW $\approx 7\times10^6$ US$ and then significantly. All of these are of interest to decision makers since both PBP and NPW are the popular economic criteria used for evaluating and selecting projects in industrial setting. As shown in Figures 1.1b and 1.1c, optimal values of decision variables

could vary with the objectives. Thus, the optimal solutions of MOO problems can be represented in two spaces: objective space (Figure 1.1a) and decision variable space (Figures 1.1b and 1.1c). In some publications, optimal values of objectives shown in Figure 1.1(a) are referred to as the Pareto-optimal (or simply Pareto) front. Besides the optimal values of the objectives, optimal values of the decision variables are of interest in selecting and implementing one of the optimal solutions in the industry.

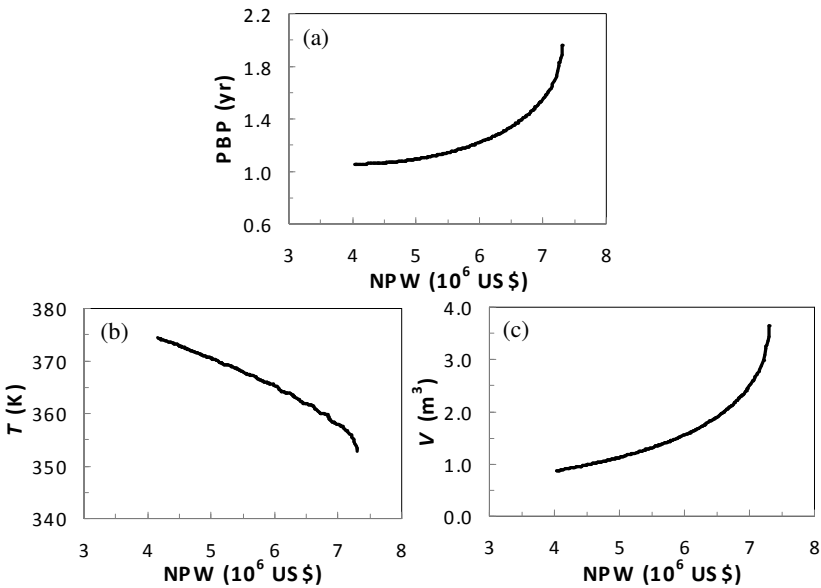


Fig. 1.1 Optimal results of Williams-Otto process for minimizing payback period, PBP and maximizing net present worth, NPW: (a) PBP versus NPW, (b) reactor temperature, *T*, versus NPW and (c) reactor volume, *V*, versus NPW.

Pareto-optimal solutions for another example – optimization of the dual independent expander refrigeration system for liquefaction of natural gas for minimizing the total capital cost (C_{total}) and the total shaftwork required (W_{total}), are shown in Figure 1.2. This example is one of the two cases described in Chapter 8. In this case, a kink is observed in the Pareto-optimal front when C_{total} is about 9.7 MM$ (Figure 1.2a), which corresponds to the point of discontinuity observed in optimal value of one decision variable (Figure 1.2b). The optimal value of another decision variable, on the other hand, is practically constant (Figure 1.2c). Although discrete points are shown in Figure 1.2, the

Pareto-optimal solutions are probably smooth curves except for the discontinuity in Figure 1.2b.

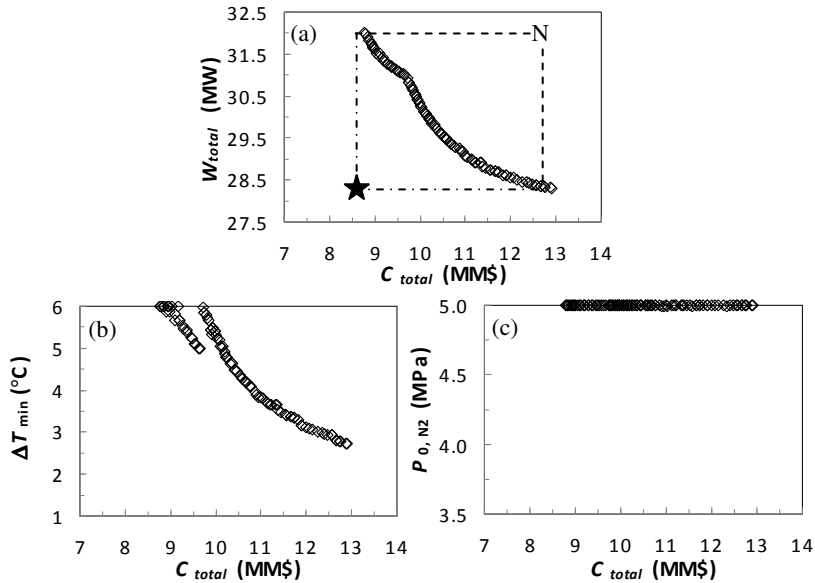


Fig. 1.2 Pareto-optimal solutions for the optimization of the dual independent expander refrigeration process: (a) objectives (W_{total} and C_{total}) to be minimized, and (b) and (c) two decision variables. See Chapter 8 for further details.

In MOO, ideal and nadir objective vectors are occasionally used. The ideal objective vector contains the optimum values of the objectives, when each of them is optimized individually disregarding the other objectives. The ideal objective vector denoted by superscript * (i.e., $[f_1^* \; f_2^*]$) is shown in Figure 1.2a along with the nadir objective vector denoted by superscript N (i.e., $[f_1^N \; f_2^N]$). Here, f_1^N is the value of $f_1(\mathbf{x})$ when $f_2(\mathbf{x})$ is optimized individually, and f_2^N is the value of $f_2(\mathbf{x})$ when $f_1(\mathbf{x})$ is optimized individually. Components of the nadir objective vector are the upper bounds (i.e., most pessimistic values) of objectives in the Pareto-optimal set. In case of two objectives, as shown in Figure 1.2a, they correspond to the value of one objective when the other is optimized individually. This may not be the case if there are more than two objectives (Weistroffer, 1985). The ideal objective vector is not realizable unless the objectives are non-conflicting in which case the MOO problem has only a unique solution, namely, ideal objective vector. However, it tells us the best possible value for each of the

objectives. On the other hand, nadir objective vector is not a desirable solution; further, it may or may not be feasible depending on the constraints. Components of both the ideal and nadir objective vectors are useful for normalizing the objectives in some MOO methods.

1.3 Multi-Objective Optimization: Methods

Many methods are available for solving MOO problems, and many of them involve converting the MOO problem into one or a series of SOO problems. Each of these problems involves the optimization of a 'scalarizing' function, which is a function of original objectives, by a suitable method for SOO. There are many ways of defining a scalarizing function, and therefore many MOO methods exist. Although the scalarization approach is conceptually simple, the resulting SOO problems may not be easy to solve.

Available methods for MOO can be classified in different ways. One of them is based on whether many Pareto-optimal solutions are generated or not, and the role of the decision maker (DM) in solving the MOO problem. This particular classification, adopted by Miettinen (1999) and Diwekar (2003), is shown in Figure 1.3. The DM can be one or more individuals entrusted with the task of selecting one of the Pareto-optimal solutions for implementation based on their experience and other considerations not included in the MOO problem. As shown in Figure 1.3, MOO methods are firstly divided into two main groups – generating methods and preference-based methods. As the names imply, the former methods generate one or more Pareto-optimal solutions without any inputs from the DM. The solutions obtained are then provided to the DM for selection. On the other hand, preference-based methods utilize the preferences specified by the DM at some stage(s) in solving the MOO problem.

The generating methods are further divided into three sub-groups, namely, no-preference methods, *a posteriori* methods using the scalarization approach and *a posteriori* methods using the multi-objective approach. No-preference methods, as the name indicates, do not require the relative priority of objectives whatsoever. Although a particular method gives only one Pareto-optimal solution, a few Pareto-optimal solutions can be obtained by using different no-preference methods (and so different metrics). Methods in this sub-group include

the method of global criterion and multi-objective proximal bundle method.

The ε-constraint and weighting methods belong to *a posteriori* methods using the scalarization approach. These methods convert an MOO problem into a SOO problem, which can then be solved by a suitable method to find one Pareto-optimal solution. A series of such SOO problems will have to be solved to find the other Pareto-optimal solutions. See Chapter 6 for a discussion of the weighting and ε-constraint methods, their properties and relative merits.

A posteriori methods using the multi-objective approach rank trial solutions based on objective values and finally find many Pareto-optimal solutions. They include population-based methods such as non-dominated sorting genetic algorithm and multi-objective differential evolution as well as multi-objective simulated annealing. In effect, all *a posteriori* methods provide many Pareto-optimal solutions to the DM, who will subsequently review and select one of them for implementation. Thus, the role of DM in these methods is after finding the Pareto-optimal solutions, which justifies their name - *a posteriori* methods. Classifications described in Miettinen (1999) and Diwekar (2003) had only one sub-group for *a posteriori* methods. Here, as shown in Figure 1.3, they are divided into two sub-groups – *a posteriori* methods using the scalarization approach and *a posteriori* methods using the multi-objective approach, for two reasons. Firstly, the methods in the two sub-groups employ different approaches for solving the MOO problems, and, secondly, several methods of this type have been developed and applied to many applications in the past ten years.

Preference-based methods have been divided into two sub-groups: *a priori* methods and interactive methods. In the former methods, preferences of the DM are sought and included in the initial formulation of a suitable SOO problem. Examples of *a priori* methods are value function methods, lexicographic ordering and goal programming. The approach of value function methods involves formulating a value function, which includes original objectives and preferences of the DM for optimization and then solving the resulting SOO problem. Weighting method is one particular case of value function methods. In lexicographic ordering, the DM must arrange the objectives according to their importance for subsequent solution by a SOO method. The DM provides an aspiration level for each of the objectives (whose achievement is the

goal) in goal programming; a suitable SOO problem is then formulated and solved.

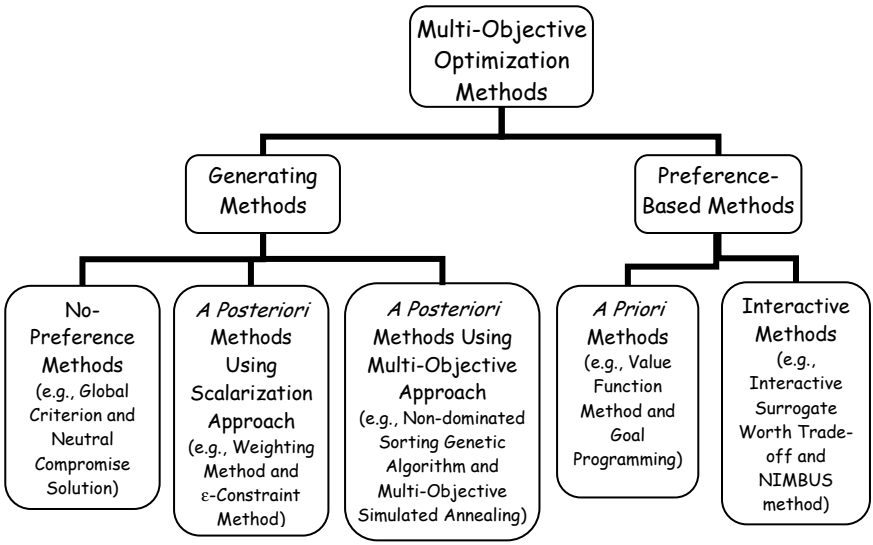


Fig. 1.3 Classification of multi-objective methods.

Interactive methods, as the name implies, requires interaction with the DM during the solution of the MOO problem. After an iteration of these methods, s/he reviews the Pareto-optimal solution(s) obtained and articulates, for example, further change (either improvement, compromise or none) desired in each of the objectives. These preferences of the DM are then incorporated in formulating and solving the optimization problem in the next iteration. At the end of the iterations, the interactive methods provide one or several Pareto-optimal solutions. Examples of these methods are interactive surrogate worth trade-off method and the NIMBUS method, which have been applied to several chemical engineering applications.

The classification in Figure 1.3 takes into account the recent developments and provides a good overview of available MOO methods. Relative merits and limitations of groups of methods are summarized in Table 1.1. A few of the MOO methods can be placed in another group. For example, weighting method in the *a posteriori* methods is a special case of value function methods in the *a priori* methods. The ε-constraint method from the *a posteriori* methods and goal programming from

a priori methods have been adopted for developing interactive methods. Thus, classification of MOO methods is somewhat subjective. For details on many methods including their theoretical properties and strengths, interested readers are referred to the comprehensive book by Miettinen (1999).

Table 1.1 Main Features, Merits and Limitations of MOO Methods

Methods	**Features, Merits and Limitations**
No Preference Methods (e.g., global criterion and neutral compromise solution)	These methods, as the name indicates, do not require any inputs from the decision maker either before, during or after solving the problem. Global criterion method can find a Pareto-optimal solution, close to the ideal objective vector.
A Posteriori Methods Using Scalarization Approach (e.g., weighting and ε-constraint methods)	These classical methods require solution of SOO problems many times to find several Pareto-optimal solutions. ε-constraint method is simple and effective for problems with a few objectives. Weighting method fails to find Pareto-optimal solutions in the non-convex region although modified weighting methods can do so. It is difficult to select suitable values of weights and ε. Solution of the resulting SOO problem may be difficult or non-existent.
A Posteriori Methods Using Multi-Objective Approach (many based on evolutionary algorithms, simulated annealing, ant colony techniques etc.)	These relatively recent methods have found many applications in chemical engineering. They provide many Pareto-optimal solutions and thus more information useful for decision making is available. Role of the DM is after finding optimal solutions, to review and select one of them. Many optimal solutions found will not be used for implementation, and so some may consider it as a waste of computational time.
A Priori Methods (e.g., value function, lexicographic and goal programming methods)	These have been studied and applied for a few decades. Their recent applications in chemical engineering are limited. These methods require preferences in advance from the DM, who may find it difficult to specify preferences with no/limited knowledge on the optimal objective values. They will provide one Pareto-optimal solution consistent with the given preferences, and so may be considered as efficient.
Interactive Methods (e.g., interactive surrogate worth trade-off and NIMBUS methods)	Decision maker plays an active role during the solution by interactive methods, which are promising for problems with many objectives. Since they find one or a few optimal solutions meeting the preferences of the DM and not many other solutions, one may consider them as computationally efficient. Time and effort from the DM are continually required, which may not always be practicable. The full range of Pareto optimal solutions may not be available.

Selected MOO methods are described in detail in later chapters of this book. Here, the weighting method and the ε-constraint method will be briefly described. These two classical methods have been used for solving several chemical engineering applications. Interestingly, a few studies reported in the literature have used the weighting or ε-constraint method without explicitly referring MOO. For example, Therdthai *et al.* (2002) optimized the bread oven temperature to minimize the weight loss during baking, for several values of baking times. Obviously, it is desirable to reduce the baking time, which is thus the second objective but considered as a constraint in Therdthai *et al.* (2002).

For solving the MOO problem in equation 1.1 by the weighting method, it is converted into the following SOO problem:

$$\text{Minimize} \qquad w\frac{f_1(\mathbf{x})-f_1^*}{f_1^N-f_1^*}+(1-w)\frac{f_2(\mathbf{x})-f_2^*}{f_2^N-f_2^*} \tag{1.2}$$

with respect to **x** **s**ubject to the bounds and constraints in equations 1.1(c) to 1.1(e). Here, $0 \le w \le 1$ is the weighting factor. Recall that the superscript * and N refer to the ideal and nadir objective vector respectively. These vector components are used for normalizing the objectives, which are likely to have significantly different magnitudes in applications. Although it is possible to define the objective in equation 1.2 without the normalizing factors, the solution of the resulting SOO problem will depend on w to a greater extent. The optimization problem in equation 1.2 will have to be solved several times, each time with a different w, in order to find several Pareto-optimal solutions. Note that $w = 1$ corresponds to minimizing $f_1(\mathbf{x})$ by itself whereas $w = 0$ correspond to minimizing $f_2(\mathbf{x})$ alone. Even though the weighting method is conceptually straightforward, choosing suitable w to find many Pareto-optimal solutions is difficult.

In the ε-constraint method, the MOO problem is transformed into a SOO problem by retaining only one of the objectives and converting all others into inequality constraints. For example, the MOO problem in equation 1.1 is transformed as:

$$\text{Minimize} \qquad f_2(\mathbf{x}) \tag{1.3}$$

with respect to **x** **s**ubject to the bounds and constraints in equations 1.1(c) to 1.1(e) as well as an additional constraint: $f_1(\mathbf{x}) \le \varepsilon$. Here, the second objective, $f_2(\mathbf{x})$ is retained but the objective, $f_1(\mathbf{x})$ is included as an inequality constraint such that its value is not more than ε at the optimal solution of the problem in equation 1.2. Obviously, the user will have to

select which objective to be retained and the value of ε. The problem in equation 1.3 will have to be solved for a range of ε values in order to find many Pareto-optimal solutions. The difficulties in the ε-constraint method are selection of ε value and solving the problem in equation 1.3. The additional constraint in this problem makes it more difficult to solve. Further, the SOO problem in the ε-constraint method may not have a feasible solution for some ε values. Values of the ideal and nadir objective vectors can be used for selecting suitable ε values.

1.4 Alkylation Process Optimization for Two Objectives

An important process in petroleum refining is the alkylation process, wherein a light olefin such as propene, butene or pentene reacts with isobutane in the presence of a strong sulfuric acid catalyst to produce the alkylate product (e.g., 2,2,4 tri-methyl pentane from butene and isobutane). The alkylate product is used for blending with refinery products such as gasoline and aviation fuel in order to increase their Octane Number. Jones (1996) provides a comprehensive overview of the alkylation process, its chemistry, design and operational aspects. Sauer *et al.* (1964) developed a nonlinear model for the alkylation process and used it for optimization via linear programming methods. Since then, many researchers (e.g., Bracken and McCormick, 1968; Luus and Jaakola, 1973; Rangaiah, 1985) employed this model in their optimization studies. Also, alkylation process optimization is a classic example included in the text-book on optimization by Edgar *et al.* (2001). To the best of our knowledge, only Luus (1978) reported alkylation process optimization for multiple objectives by the ε-constraint method; for this optimization, he modified the bounds on variables slightly compared to those in Sauer *et al.* (1964). Here, we will describe optimization of the alkylation process for two objectives by the ε-constraint method.

1.4.1 *Alkylation Process and its Model*

A simplified process flow diagram of the alkylation process is shown in Figure 1.4. The process has a reactor with olefin feed, isobutane makeup and isobutane recycle as the inlet streams. Fresh acid is added to catalyze the reaction and spent acid is withdrawn. The exothermic

reactions between olefins and isobutane occur at around room temperature, and excess isobutane is used. The hydrocarbon outlet stream from the reactor is fed into a fractionator from where isobutane is recovered at the top and recycled back to the rector, and the alkylate product is withdrawn from the bottom.

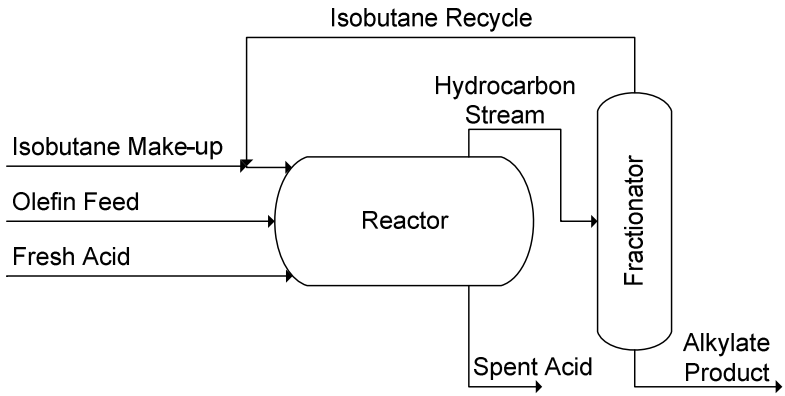


Fig. 1.4 Simplified schematic of the alkylation process.

Sauer *et al.* (1964) developed a model for this process based on a judicious combination of first principles, empirical equations and a number of simplifying assumptions. The resulting model has 10 variables and seven equality constraints. Bracken and McCormick (1968) have presented this model and the optimization problem in a different way. Noting that the four equality constraints derived by regression analysis need not be satisfied exactly, they converted them into eight inequality constraints. This optimization problem and its solution are concisely described by Edgar *et al.* (2001). Rangaiah (1985) studied both the problems - original one with seven equality constraints and the modified one with both equality and inequality constraints.

Variables involved in the alkylation process model of Sauer *et al.* (1964) and their bounds are summarized in Table 1.2. The SOO problem of this process is as follows.

Maximize Profit, P (\$/day) = $0.063\, x_4 x_7 - 5.04\, x_1 - 0.035\, x_2 - 10.0\, x_3 - 3.36\, x_5$ (1.4a)

With respect to x_1, x_7 and x_8

Subject to $0 \le x_1 \le 2{,}000$ (1.4b)

$90 \le x_7 \le 95$ (1.4c)

$$3 \le x_8 \le 12 \tag{1.4d}$$

$$0 \le [x_4 \equiv x_1(1.12 + 0.13167x_8 - 0.006667x_8^2)] \le 5{,}000 \tag{1.4e}$$

$$0 \le [x_5 \equiv 1.22x_4 - x_1] \le 2{,}000 \tag{1.4f}$$

$$0 \le [x_2 \equiv x_1 x_8 - x_5] \le 16{,}000 \tag{1.4g}$$

$$85 \le [x_6 \equiv 89 + (x_7 - (86.35 + 1.098x_8 - 0.038x_8^2))/0.325] \le 93 \tag{1.4h}$$

$$145 \le [x_{10} \equiv -133 + 3x_7] \le 162 \tag{1.4i}$$

$$1.2 \le [x_9 \equiv 35.82 - 0.222x_{10}] \le 4 \tag{1.4j}$$

$$0 \le [x_3 \equiv 0.001\ (x_4 x_6 x_9)/(98\text{-}x_6)] \le 120 \tag{1.4k}$$

The above problem will be referred to as Problem A in the following. The 7 inequality constraints in equations 1.4e to 1.4k are the bounds on the 7 variables (x_4, x_5, x_2, x_6, x_{10}, x_9 and x_3) in the original problem, and they arise from the elimination of these variables from the 7 equality constraints in the model thus making them dependent variables. The cost coefficients in the profit are alkylate product value ($0.063/octane-barrel), olefin feed cost ($5.04/barrel), isobutane recycle cost ($0.035/barrel), fresh acid cost ($10.0/thousand pounds) and isobutane feed cost ($3.36/barrel). The optimal solution for this SOO problem is also presented in Table 1.2. The reader can verify this using the Excel file: "Alkylation.xls" in the folder: Chapter 1 on the compact disk (CD) provided with the book.

Table 1.2 Variables, Bounds and Optimum Values in the Alkylation Process Optimization

Variables	**Lower Bound**	**Upper Bound**	**Optimum Value**
Olefin Feed, x_1 (barrels/day)	0	2,000	1,728
Isobutane Recycle, x_2 (barrels/day)	0	16,000	16,000
Acid Addition Rate, x_3 (thousand pounds/day)	0	120	98.14
Alkylate Production Rate, x_4 (barrels/day)	0	5,000	3,056
Isobutane Feed, x_5 (barrels/day)	0	2,000	2,000
Spent Acid Strength, x_6 (weight percent)	85	93	90.62
Octane Number, x_7	90	95	94.19
Isobutane to Olefin Ratio, x_8	3	12	10.41
Acid Dilution Factor, x_9	1.2	4	2.616
F-4 Performance Number, x_{10}	145	162	149.6
Profit ($/day)			**1,162**

1.4.2 *Multi-Objective Optimization Results and Discussion*

The SOO problem (Problem A) can be formulated as a two-objective problem by adding another objective. We consider two such problems:

Case A - Maximize Profit, P and Maximize Octane Number, x_7

Case B - Maximize Profit, P and Minimize Isobutane Recycle, x_2

Alkylate product with a higher octane number is better for blending with refinery products. Minimizing isobutane recycle helps to reduce fractionation and other costs associated with the recycle stream.

The ε-constraint method can be easily applied to solve both the two-objective problems since one objective (x_7 or x_2) is a variable whose range is known from the problem formulation. For this, the SOO problem for Case A is the same as Problem A except for the change in the lower bound of x_7 to ε (i.e., desired lowest value for the second objective) instead of 90. Since the optimum x_7 in Problem A is at 94.19, the solution of Case A by the ε-constraint for $\varepsilon \leq 94.19$ will be the same as that in Table 1.2. The Pareto-optimal solutions of Case A obtained by the ε-constraint method for several values of ε in the range 94.2 to 95.3 are presented in Figure 1.5. These and other results for the alkylation process optimization were obtained using the Solver tool in Excel. Note that the upper bound of x_7 will have to be changed for higher values of ε in order to have feasible region satisfying all bounds and constraints. Further, no feasible solution exists for ε above 95.3, which is understandable since the model is semi-empirical.

In Figure 1.5, the first plot shows the two objectives (P on the x-axis and x_7 on the y-axis) and the remaining plots show the optimal values of all other decision variables versus P. Increase in P from 1,000 to 1,106 \$/day is accompanied by x_7 decreasing from 95.3 to 94.3; thus, the two objectives, P and x_7 are contradictory leading to the optimal Pareto front in the first plot in Figure 1.5. All other decision variables with the exception of x_2 (isobutane recycle) which remains at its upper bound, contribute to the optimal Pareto front. Interestingly, each of them varies with P at certain rate until P is about \$1,110/day and then follows a different trend - x_1, x_4, x_5 and x_8 become constant, x_3 and x_6 start to decrease at P > \$,1110/day. Of these, the trend of x_3 is striking – it increases with P initially and then decreases for P > \$1,110/day.

In a similar way, the two-objective problem in Case B can also be solved by the ε-constraint method; the corresponding SOO problem is the same as Problem A except that the upper bound on x_2 (which is an

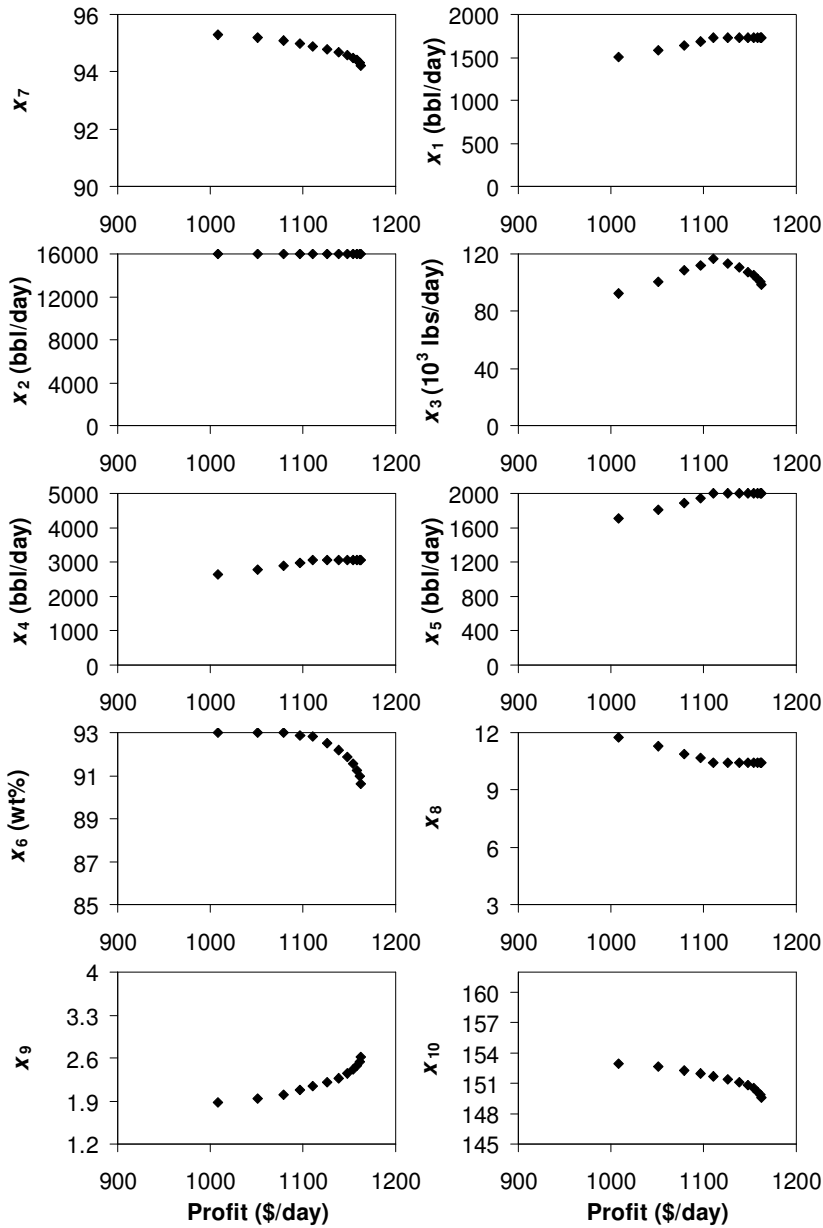


Fig. 1.5 Pareto-optimal solutions for maximizing profit and octane number (x_7) by the ε-constraint method; profit is shown on the x-axis in all plots.

inequality constraint in Problem A) is the ε. Pareto-optimal solutions resulting from solving a series of such SOO problems with ε in the range 12,000 to 17,500 barrels/day are shown in Figure 1.6. The optimal P increases from about 900 to 1,200 $/day as x_2 increases from 12,000 to 17,500 barrels/day. As in Case A, the two objectives – maximize P and minimize x_2, are contradictory. In Case B, all decision variables contribute to the optimal Pareto front and none of them is constant over the range of P shown in Figure 1.6. Similar to Case A, each of the decision variables show certain trend up to P = $1,050/day and then a different trend. Interestingly, x_3 increases with P initially and then starts decreasing beyond P = $1,050/day. Further, optimal values of decision variables in Case B are generally different from those in Case A.

Although an experienced engineer may be able to predict some trends of objectives and decision variables in Figures 1.5 and 1.6, it is impossible to foretell them accurately and correctly. On the other hand, MOO can give many optimal solutions, which in turn provide greater insight into and understanding of the process behavior. Knowing these solutions, their trends and additional considerations, the most suitable optimal solution can then be selected for implementation. The two-objective problem in Cases A can also be solved by the weighting method; this is given as an exercise at the end of this chapter.

1.5 Scope and Organization of the Book

This book, as implied by its title, focuses on both MOO techniques and their applications in chemical engineering. The chapters in the book can be divided into three groups. The first group consists of this introduction chapter and another chapter summarizing the MOO applications in chemical engineering reported from the year 2000 to mid 2007. The second group of 5 chapters is on the MOO techniques written by leading researchers in the field. The last group of 6 chapters is on a broad range of MOO applications in chemical engineering including a few on biochemical engineering. Each of the chapters in the book can be read and understood with little reference to other chapters. Many chapters contain several exercises at the end; these and the applications discussed within the chapters can be used as projects and/or assignments in optimization courses at both undergraduate and postgraduate levels. The programs and files on the enclosed CD will be helpful to readers in solving the exercises and/or doing the projects.

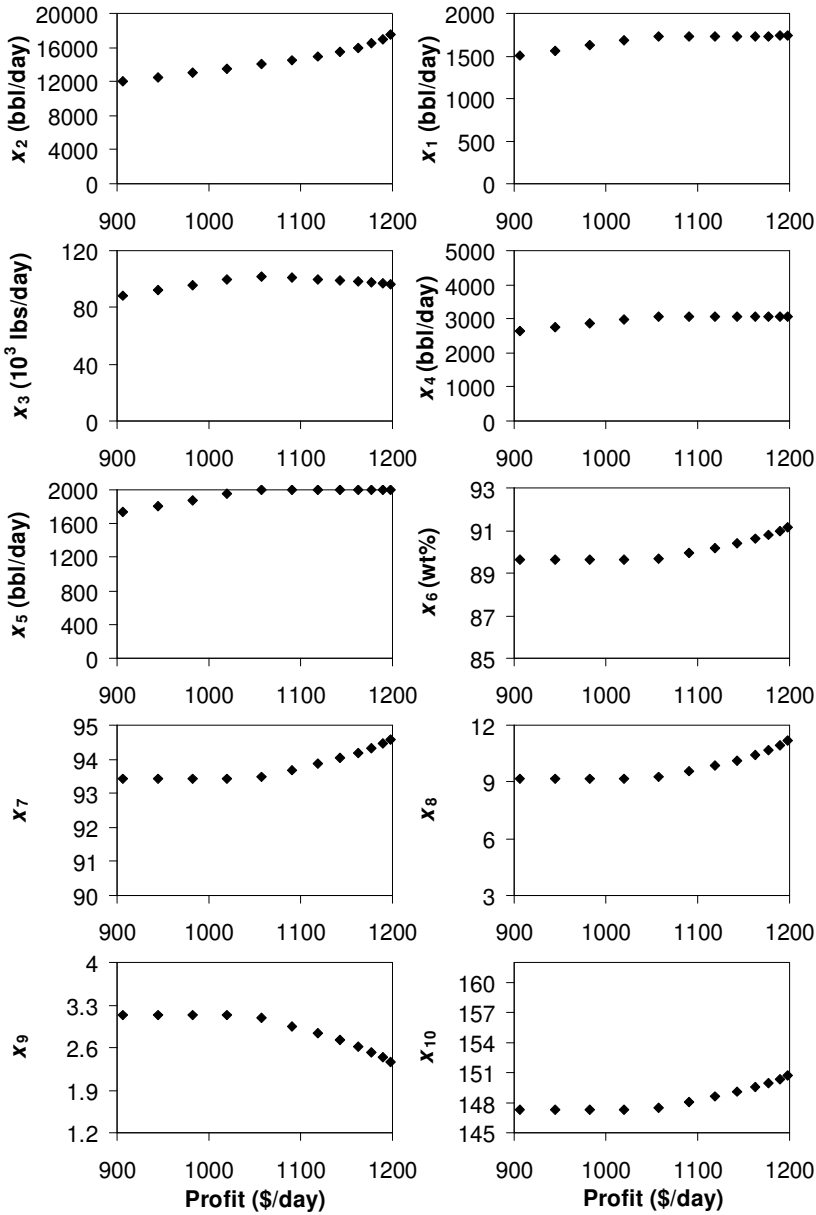


Fig. 1.6 Pareto-optimal solutions for maximizing profit and isobutane recycle (x_2) by the ε-constraint method; profit is shown on the x-axis in all plots.

Nearly 100 MOO applications in chemical engineering reported in journals from the year 2000 until mid 2007 are summarized by Masuduzzaman and Rangaiah in Chapter 2: *MOO Applications in Chemical Engineering*. These applications are categorized into five groups: (1) process design and operation, (2) biotechnology and food industry, (3) petroleum refining and petrochemicals, (4) pharmaceuticals and other products/processes, and (5) polymerization. However, the applications reported in this book are not included in this review. Many applications reviewed in Chapter 2 have employed detailed models of the processes for their optimization for a variety of objectives. Several others have used process simulators in their studies. Thus, Chapter 2 is a very rich source of reported MOO applications in chemical engineering as well as process models for many processes of importance.

In Chapter 3 entitled *Multi-objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering*, Jaimes and Coello Coello review development of evolutionary algorithms for MOO and some of their applications in chemical engineering. They also identify several contributions of chemical engineering researchers to the development of multi-objective evolutionary algorithms (MOEAs). Towards the end of the chapter, Jaimes and Coello Coello list available resources including websites and public-domain software on MOEAs, and then outline potential areas for future research on the use of MOEAs in chemical engineering.

Ramteke and Gupta, in Chapter 4: *The Jumping Gene Adaptations of Multi-objective Genetic Algorithm and Simulated Annealing*, describe genetic algorithms and simulated annealing, first for SOO, then for MOO and finally their jumping gene adaptations. The presented algorithms are tested on a few benchmark problems; the results are presented and discussed. Incorporating the macro-macro mutation operation, namely, jumping gene operator inspired from nature, leads to faster convergence of the algorithm. Towards the end of the chapter, Ramteke and Gupta review their more recent chemical engineering applications of the jumping gene adaptations of genetic algorithm and simulated annealing.

In Chapter 5 entitled *MOO using Surrogate Assisted Evolutionary Algorithm*, Ray describes a surrogate-assisted evolutionary algorithm for MOO in order to reduce the number of objective function and constraint evaluations. In this algorithm, a radial basis function (RBF) neural network is used as a surrogate/approximate model for the objectives and constraints, for certain number of generations instead of the original

objective functions and constraints before developing a new surrogate model. The results on several test functions show that the surrogate-assisted evolutionary algorithm provides better non-dominated solutions than the elitist non-dominated sorting genetic algorithm (NSGA-II) for the same number of actual function and constraint evaluations. This is particularly advantageous in chemical engineering applications which involve computationally-intensive process simulation for evaluating the objectives and/or constraints.

Chapter 6: *Why Use Interactive MOO in Chemical Process Design?* by Miettinen and Hakanen focuses on problems having several (i.e., more than 2) objective functions and interactive MOO. They describe weighting method, ε-constraint method and evolutionary MOO methods, their relative merits, and then a few interactive MOO methods including their NIMBUS method and its two implementations. In the later part of the chapter, Miettinen and Hakanen discuss the application of interactive MOO methods to chemical process design. Application of the NIMBUS method is then illustrated for optimizing a simulated moving bed process for separation of fructose and glucose for four objectives. Two other applications, namely, water allocation problem having three objectives and heat recovery system design with four objectives, are also outlined.

Many studies on MOO applications in chemical engineering focus on generating Pareto-optimal solutions. The important step of ranking these solutions is the subject of Chapter 7: *Net Flow and Rough Set: Two Methods for Ranking the Pareto Domain* by Thibault. In this chapter, net flow and rough sets methods for ranking Pareto-optimal solutions are described in detail. Both the methods require the preferences and knowledge of the decision maker. A few variants of rough sets method are also discussed. In the later part of the chapter, net flow and rough set methods are applied for ranking Pareto-optimal solutions for optimization of gluconic acid production for two, three and four objectives. The results obtained are presented and discussed.

In Chapter 8: *MOO of Gas Phase Refrigeration Systems for LNG*, Shah, Rangaiah and Hoadley describe the optimization of multi-stage gas-phase refrigeration systems for capital cost and energy efficiency simultaneously, for the first time. They employed a process simulator (namely, Hysys) for simulation, NSGA-II for MOO and an interface program for linking Hysys and NSGA-II. Design and optimization of two multi-stage gas-phase refrigeration systems (one for nitrogen cooling and another for liquefaction of natural gas) are discussed in detail.

Optimization of feed to an industrial fluidized catalytic cracker (FCC) important in petroleum refining, is the application described by Tan, Phang and Yang in Chapter 9 entitled *A Multi-objective Evolutionary Algorithm for Practical Residue Catalytic Cracking Feed Optimization*. In this particular refinery, there are seven different feed streams, each with its own flow range and characteristics, which can be used for the FCC. The feed optimization problem consists of three objectives subject to four constraints. It is solved using a multi-objective optimization evolutionary algorithm (MOEA) toolbox, and Pareto-optimal solutions are presented. Of particular interest in this application is the analysis and discussion on selecting a Pareto-optimal solution for implementation based on three performance indexes: fuel gas consumption, steam consumption and exothermic reaction rate.

Lee, Rangaiah and Agrawal report the optimization of three applications for multiple economic and/or environmental objectives using NSGA-II-aJG, in Chapter 10: *Optimal Design of Chemical Processes for Multiple Economic and Environmental Objectives*. The first two applications are the optimization of the classical Williams-Otto process and the optimal design of a low density polyethylene plant, both for two economic objectives simultaneously. Results of these two applications show that some economic criteria could be conflicting depending on the model equations and objectives. The third application is on the optimization of industrial ecosystems consisting of several plants for both economic and environmental criteria. As expected, the economic and environmental criteria are found to be conflicting leading to Pareto-optimal solutions for industrial ecosystem optimization.

An interesting application of MOO to emergency response around chemical plants is described by Georgiadou, Papazoglou, Kiranoudis and Markatos in Chapter 11 entitled *Multi-objective Emergency Response Optimization around Chemical Plants*. Problem formulation for emergency response optimization and an MOEA are described. They are then applied to two case studies: emergency response optimization for accidental ammonia release from a storage tank and for BLEVE (boiling liquid expanding vapor explosion) in a petroleum refinery. The results of MOO are presented and their use for emergency planning and land-use planning is highlighted.

The penultimate chapter: *Array Informatics using Multi-objective Genetic Algorithms: From Gene Expressions to Gene Networks* by Garg is on elucidating gene networks from microarray experimental data. This

application area is new to many chemical engineers. Hence, the chapter begins with a detailed introduction covering biological background, multiple microarray experiments to measure gene expressions[1] of several hundreds of genes, interpreting and pre-processing microarray data. The measured expression ratios are first analyzed to identify groups of genes with similar ratios via clustering techniques; the results of this step (known as gene expression profiling) are then used to model the complex interactions among genes (which is referred to as gene network analysis). MOO is applicable to both gene expression profiling and network analysis. This is successfully illustrated on two gene expression data sets: a synthetic data set and a real-life data set, to find the gene networks.

The last chapter entitled *Optimization of a multi-product microbial cell factory for multiple objectives – a paradigm for metabolic pathway recipe,* by Lee, Rangaiah and Lee reports a novel application of MOO to metabolic engineering. They present the optimization of gene manipulation (knockout, overexpression or repression) for two objectives in order to optimize production of desired amino acids by *Escherichia coli* (*E. coli*). The mixed-integer MOO problem in this application was successfully solved using the NSGA-II; this was particularly facilitated by the possibility of continuous and/or integer variables within NSGA-II. The MOO results show that fluxes of desired enzymes can be increased significantly by optimal manipulation of just three enzymes.

References

Beveridge, G. S. G. and Schechter, R. S. (1970). *Optimization: Theory and Practice*, McGraw Hill, New York.

Bhaskar, V., Gupta, S. K. and Ray, A. K. (2000). Applications of multi-objective optimization in chemical engineering, *Reviews in Chemical Engineering*, 16, pp. 1-54.

Biegler, L. T., Grossmann, I. E. and Westerberg, A. W. (1997). *Systematic Methods of Chemical Process Design*, Prentice Hall, New Jersey.

Bracken, J. and McCormick, G. P. (1968). *Selected Applications of Nonlinear Programming*, John Wiley, New York.

Chankong, V. and Haimes, Y. Y. (1983). *Multi-objective Decision Making Theory and Methodology*, Elsevier Science Publishing, New York.

Coello Coello, C. A., Veldhuizen, D. A. V. and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic, New York.

[1]Gene expression is the process by which the set of instructions is read by the cell and translated into proteins.

Cohon, J. L. (1978). *Multi-objective Programming and Planning*, Academic Press, New York.

Deb, K. (2001). *Multi-objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK.

Diwekar, U. M. (2003). *Introduction to Applied Optimization*, Kluwer Academic, Norwell, Mass.

Edgar, T. F., Himmelblau, D. M. and Lasdon, L. S. (2001). *Optimization of Chemical Processes*, Second Edition, McGraw-Hill, New York.

Floudas, C. A. (1995). *Nonlinear Mixed-integer Optimization: Fundamentals and Applications*, Oxford University Press, New York.

Floudas, C. A. (1999). *Deterministic Global Optimization: Theory, Methods and Applications,* Kluwer Academic, Boston.

Haimes, Y. Y., Tarvainen, K., Shima, T. and Thadathil, J. (1990). *Hierarchical Multi-objective Analysis of Large-Scale Systems*, Hemisphere Publishing, New York.

Himmelblau, D. M. (1972). *Applied Nonlinear Programming*, McGraw-Hill, New York.

Hwang, C. L. and Masud, A. S. M. (1979). *Multiple Objective Decision Making - Methods and Applications: A State-of-the-Art Survey*, Springer-Verlag, Lecture Notes in Economics and Mathematical Systems, Berlin.

Jones, D. S. J. (1996). *Elements of Petroleum Processing*, Chapter 14, John Wiley, New York.

Lapidus, L. and Luus, R. (1967). *Optimal Control in Engineering Processes*. Blaisdell, Waltham, Mass.

Luus, R. (2000). *Iterative Dynamic Programming*, Chapman & Hall, Boca Raton.

Luus, R. and Jaakola, T. H. I. (1973). Optimization by direct search and systematic reduction of the size of search region. *AIChE Journal*, 19, pp. 760-766.

Luus, R. (1978). Optimization of systems with multiple objective functions. *International Congress, European Federation of Chemical Engineering*, Paris, pp. 3-8.

Miettinen, K. (1999). *Nonlinear Multi-objective Optimization,* Kluwer Academic Publishers, Boston.

Peters, M. S., Timmerhaus, K. D. and West, R. E. (2003). *Plant Design and Economics for Chemical Engineers*, McGraw-Hill, Boston.

Ray, W. H. and Szekely, J. (1973). *Process Optimization with Applications in Metallurgy and Chemical Engineering*, Wiley, New York.

Rangaiah, G. P. (1985). Studies in constrained optimization of chemical processes. *Computers and Chemical Engineering*, 9, pp. 395-404.

Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M. (2006). *Engineering Optimization: Methods and Applications*, Second Edition, John Wiley, New Jersey.

Sauer, R. N., Colville, Jr., A. R. and Burwick, C. W. (1964). Computer Points the Way to More Profits, *Hydrocarbon Processing & Refiner*, 49, No. 2, pp. 84-92.

Sawaragi Y., Nakayama, H. and Tanino, T. (1985). *Theory of Multi-objective Optimization*, Academic Press, Orlando, Florida.

Seider, W. D., Seader, J. D. and Lewin, D. R. (2003). *Product and Process Design Principles: Synthesis, Analysis, and Evluation*, John Wiley, New York.

Stadler, W. (1988). *Multi-criteria Optimization in Engineering and in the Sciences*, Plenum Press, New York.

Tan, K. C., Khor, E. F. and Lee, T. H. (2005). *Multi-objective Evolutionary Algorithms and Applications*, Springer, London.

Tarafder, A., Rangaiah, G. P. and Ray, A. K. (2007). A Study of Finding Many Desirable Solutions in Multi-objective Optimization of Chemical Processes. *Computers and Chemical Engineering*, 31, pp. 1257-1271.

Tawarmalani, M. and Sahinidis, N. V. (2002). *Convexification and Global Optimization in Continuous and Mixed-integer Nonlinear Programming: Theory, Algorithms, Software and Applications*, Kluwer Academic, Dordrecht.

Therdthai, N. Zhou, W. and Adamczak, T. (2002). Optimization of the temperature profile in bread baking, *Journal of Food Engineering*, 55, pp. 41-48.

Weistroffer, H. R. (1985). Careful usage of pessimistic values is needed in multiple objectives optimization, *Operations Research Letters*, 4, pp. 23-25.

Exercises

1.1 Identify a daily-life situation (e.g., selection of a course of study, job and investment) requiring selection. What are the choices available? What are the objectives to be achieved? Are one or more objectives conflicting in nature? What are the constraints? Discuss these and any other related issues qualitatively. State the information and/or relations required if the optimization problem has to be solved quantitatively.

1.2 Optimize the alkylation process for two objectives (cases A and/or B) using the ε-constraint method and Solver tool in Excel. Are the results comparable to those in Figures 1.5 and 1.6?

1.3 Optimize the alkylation process for two objectives (cases A and/or B) using the weighting method. One can use the Solver tool in Excel for SOO. Try different weights to find as many Pareto-optimal solutions as possible. Compare and comment on the solutions obtained with those obtained by the ε-constraint method (Figures 1.5 and 1.6). Which of the two methods – the weighting and the ε-constraint method, is better?

1.4 Optimize the alkylation process for two objectives (cases A and/or B) using a MOO program (e.g., see Chapters 4 and 5 for two programs provided on the attached CD). Note the computational time taken for each of the two cases. Compare the results obtained with those presented in this chapter. Also, optimize the alkylation process for three objectives: maximize profit, maximize octane number and minimize isobutene recycle, using the same program. Compare and discuss the results obtained with those for cases A and B. Does three-objective optimization require comparable or more computational time than two-objective optimization?

This page intentionally left blank

Chapter 2

Multi-Objective Optimization Applications in Chemical Engineering

Masuduzzaman and G. P. Rangaiah*
Department of Chemical & Biomolecular Engineering
National University of Singapore,
Engineering Drive 4, Singapore 117576

*Corresponding Author; e-mail: chegpr@nus.edu.sg

Abstract

Multi-objective optimization (MOO) has received considerable attention from researchers in chemical engineering. Bhaskar *et al*. (2000a) have reviewed reported applications of MOO in chemical engineering until 2000. In this chapter, nearly hundred MOO applications in chemical engineering reported in journals from 2000 until mid 2007 are reviewed briefly. These are categorized into five groups: (1) process design and operation, (2) biotechnology and food industry, (3) petroleum refining and petrochemicals, (4) pharmaceuticals and other products/processes, and (5) polymerization. However, applications reported in this book are not included in this review.

Keywords: MOO Applications, Process Design, Process Operation, Biotechnology, Food Industry, Petroleum Refining, Petrochemicals, Pharmaceuticals, Polymerization.

2.1 Introduction

Optimization refers to obtaining the values of decision variables, which correspond to the maximum or minimum of one or more objective functions. Major part of research in optimization and its applications in chemical engineering considers only one objective function, probably due to the available computational resources including methods. However, most real world chemical engineering problems involve one or more objectives which are conflicting in nature. The way of finding solutions of such problem is known as multi-objective optimization (MOO). Over the last two decades, this field has grown significantly and many chemical engineering applications of MOO have been reported in the literature.

There are three reviews of MOO applications in chemical engineering. Bhaskar *et al*. (2000a) presented the background of MOO, different methods and their applications until the year 2000. These applications covered all areas in chemical engineering. MOO applications in polymerization are included in the review of genetic algorithm applications in polymer science and engineering by Kasat *et al*. (2003). Applications of non-dominated sorting genetic algorithm (NSGA), NSGA-II and its jumping gene adaptations in chemical reaction engineering were reviewed by Nandasana *et al*. (2003a). Although these two works in 2003 covered many MOO applications of interest to chemical engineers, there has been no comprehensive review of MOO applications in all areas of chemical engineering since the year 2000.

In this chapter, we summarize MOO applications in all areas of chemical engineering reported in journal publications from 2000 until mid 2007. This period is chosen so that it overlaps very slightly with the earlier, comprehensive review of Bhaskar *et al*. (2000a). Every effort is taken to include all reported MOO applications of chemical engineering in journals, in this chapter. Conference publications are not included in this summary for two reasons: their limited availability, and conference publications are often expanded and later published in journals. Figure 2.1 shows the trend of journal publications on MOO applications in chemical engineering from the year 2000 to mid 2007. Note that the number of publications in the year 2007 is for about half of the year only.

It can be seen from Figure 2.1 that MOO research and applications in chemical engineering are increasing: number of journal publications on MOO applications in chemical engineering is around 10 to 15 in the

years 2000 to 2002 with the minimum number in the year 2001, and it has increased to 20 to 25 in the subsequent years with the maximum number in the year 2003. It is interesting to see the sudden increase in 2003 as if to compensate for the minimum in the year 2001! Chemical engineering applications of MOO reported in the journals from the year 2000 to mid 2007 excluding those reported in this book are summarized under five groups: (1) process design and operation, (2) biotechnology and food industry, (3) petroleum refining and petrochemicals, (4) pharmaceuticals and other products/processes, and (5) polymerization. The first group contains applications of interest to several industries. Each of the next four groups focuses on one industry/area of interest to chemical engineers.

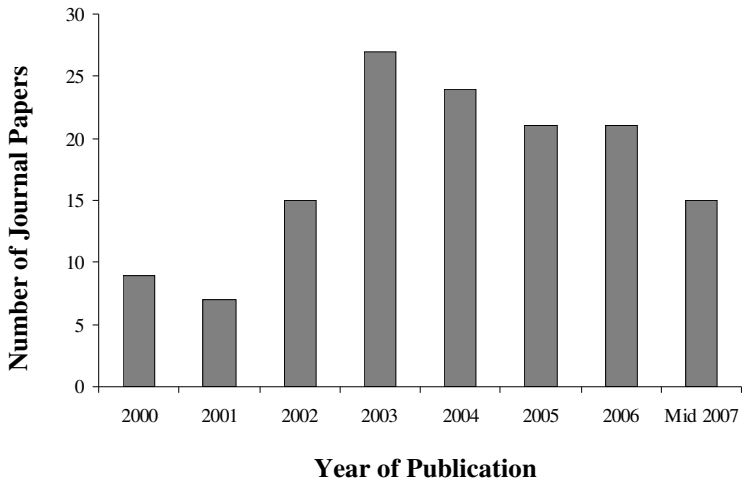


Fig. 2.1 Number of journal papers on chemical engineering applications of MOO from the year 2000 to mid 2007.

2.2 Process Design and Operation

Process design and operation, which are the central and important areas in chemical engineering, have attracted many applications of MOO since the year 2000. In all, there are 35 applications of MOO for process design and operation (Table 2.1). These cover fluidized bed dryer, cyclone separator, a pilot scale venturi scrubber, hydrogen cyanide production, heat exchanger network, grinding, froth floatation circuits, simulated moving bed (SMB) and related separation systems, thermal

and pressure swing adsorption, toluene recovery, heat recovery system, a co-generation plant, batch plants, safety-related decision making, system reliability, proportional-integral controller design, waste incineration, parameter estimation, industrial ecosystems, scheduling and supply chain networks etc. Several of these applications considered environmental objectives in addition to economic objectives (e.g., Chen *et al.*, 2002, 2003a; Hoffmann *et al.*, 2001, 2004; Kim and Diwekar, 2002).

A few MOO methods were also developed and tested as part of the above applications (Table 2.1). These include modified sum of weighted objective function method (Ko and Moon, 2002), multi-objective GA (Dedieu *et al*., 2003), jumping gene adaptations of NSGA-II, and multi-criteria branch and bound (MCBB) algorithm. In particular, NSGA-II and its jumping gene adaptations have been applied to many applications in process design and operation (e.g., Guria *et al*., 2005a, 2005b, 2006; Kurup *et al*., 2006a, 2006b; Sarkar *et al*., 2006 and 2007). Hakanen *et al*. (2005 and 2006) reported the application of an interactive MOO method, namely, NIMBUS to the heat recovery system and a co-generator plant. Several other methods – analytic hierarchical process, ant colony method, single and dual population evolutionary algorithm were applied to reactor-regeneration system, process synthesis, crystallization, controller design and scheduling problems (Table 2.1).

2.3 Biotechnology and Food Industry

Both biotechnology and food industry are closely related to chemical engineering, and of interest to many chemical engineers. There are 16 applications of MOO in these two areas since the year 2000 to mid 2007 (Table 2.2). These include food processing, beer dialysis, wine filtration, glucose-fructose separation, fermentation, and production of lipid, lysine, proteins and penicillin. First principles models were employed in many of the 16 applications reported since 2000. Different MOO methods, which include both the classical methods and evolutionary algorithms, were used in solving the applications in biotechnology and food industry; NSGA and its adaptations were used in 8 of these applications. Pareto-optimal solutions were successfully obtained and discussed in these studies. In addition to this, Halsall-Whitney *et al*. (2003), Muniglia *et al*. (2004) and Halsall-Whitney and Thibault (2006) rank the Pareto-optimal solutions by the net flow method taking into account the preferences of the decision maker.

Table 2.1 MOO Applications in Process Design and Operation

	Application	Objectives	Method	Comments	Reference(s)
1	Fluidized bed dryer	Minimization of product color deterioration and unit cost of final product.	No-preference method	Application is a dehydration plant for sliced potato. Pareto-optimal solutions were found from the single objective contours.	Krokida and Kiranoudis (2000)
2	Industrial cyclone separator	Two problems: maximization of overall collection efficiency while minimizing (a) pressure drop and (b) cost.	NSGA	Pareto-optimal solutions of the two problems are similar although their ranges are different.	Ravi *et al*. (2000)
3	Parameter estimation for a fermentation process	Two or four objectives, each of which corresponds to sum of squares of errors in a batch or fed-batch experiment.	Hybrid differential evolution (HDE)	Weighted min-max method was used to scalarize the problem, which was then solved by HDE.	Wang and Sheu (2000)
4	Process alternatives for hydrogen cyanide production	Maximization of economic benefit and minimization of environmental impact.	A preference-based approach	Hoffmann *et al.* (2001) considered total annualized profit per service unit (TAPPS) and material intensity per service (MIPS) as economic and environmental indicator respectively, while Hoffmann *et al.* (2004) considered Eco-indicator 99 (EI99) for environmental objective as well as uncertainty in model parameters.	Hoffmann *et al.* (2001)
			ε-constraint method		Hoffmann *et al.* (2004)
5	Plant-wide waste management	Simultaneous minimization of both cost and environmental impact.	Goal Programming	The proposed methodology consisting of superstructure generation and optimization for multiple objectives, is illustrated for optimal solvent recovery from a mixture of acetone, benzene, ethylene dichloride and toluene. Chakraborty and Linninger (2003) considered uncertainty in parameters and degree of flexibility in the design of plant-wide management policies.	Chakraborty and Linninger (2002) Chakraborty and Linninger (2003)
6	Volatile organic compounds (VOC) recovery	Maximization of net present value and minimization of a composite environmental index.	Analytic hierarchy process (AHP)	Seven environmental indices were combined into a single normalized and weighted environmental index. AHP aggregated the economic and environmental objective into a single objective function. Chen *et al.* (2002) and (2003) used exhaustive search and the genetic algorithm respectively, to solve the single objective optimization problem.	Chen *et al.* (2002) Chen *et al.* (2003a)

Table 2.1 MOO Applications in Process Design and Operation (Continued)

	Application	Objectives	Method	Comments	Reference(s)
7	Heat exchanger network	Simultaneous minimization of both the total annual cost and the composite environmental index.	Analytic hierarchy process	Chen *et al.* (2002) studied only one case, whereas Wen and Shonnard (2003) studied three cases with different stream data.	Chen *et al.* (2002); Wen and Shonnard (2003)
8	Solvent selection for acetic acid recovery	Maximization of acetic acid recovery and process flexibility, and minimization of environmental impact based on lethal-dosage (LD_{50}) and lethal-concentration (LC_{50})	Constraint multi-objective programming (MOP) method	Aspen Plus was employed to simulate the process, and uncertainty was also considered. The proposed MOP method is similar to the ε-constraint method.	Kim and Diwekar (2002)
9	Cyclic adsorption processes	Two examples: (a) thermal swing adsorption - maximization of total adsorption efficiency and minimization of consumption rate of regeneration energy, and (b) rapid pressure swing adsorption - maximization of both purity and recovery of the desired product for RPSA.	Modified Sum of Weighted Objective Function (SWOF) method	Modified SWOF method is superior to the conventional SWOF as it was able to find the non-convex part of the Pareto-optimal set.	Ko and Moon (2002)
10	Toluene recovery process	Maximization of economic benefit and minimization of environmental indicator. In addition to these two objectives, Kheawhom and Hirao (2002) considered process robustness measures (failure probability and deviation ratio) also.	Normal boundary intersection method	Sustainable process index was used as environmental indicator. Product revenue less capital and operating costs was the economic indicator	Kheawhom and Hirao (2002)
			NSGA-II	Kheawhom and Hirao (2004) proposed and used a two-layer methodology. Inner layer consists of single objective optimization to minimize operating cost. The outer layer involves multi-objective optimization.	Kheawhom and Hirao (2004)
11	A pilot-scale venturi scrubber	Maximization of overall collection efficiency and minimization of pressure drop.	NSGA	Ravi *et al*. (2003) considered a design variable besides operation variables in the optimization by Ravi *et al*. (2002).	Ravi *et al*. (2002 and 2003)
12	Reactor-separator-recycle system	Minimization of total cost and maximization of controllability.	ε-constraint method	Eigen-value optimization approach was used along with the ε-constraint method, to solve the design problem for two different control strategies.	Blanco and Bandoni (2003)

Table 2.1 MOO Applications in Process Design and Operation (Continued)

	Application	Objectives	Method	Comments	Reference(s)
13	System reliability	Four problems with two or three objectives from: (1) maximization of system reliability, (2) minimization of system cost, and (3) minimization of system weight for optimum redundancy allocation.	Simulated Annealing-based MOO methods	Five simulated annealing-based algorithms were tested, and their performance was found to be problem-specific. Simultaneous use of all five algorithms is suggested to generate many optimal solutions.	Suman (2003)
14	Supply chain networks	Simultaneous maximization of (1) participants' expected profits, (2) average safe inventory level (for plants, distribution centers and retailers), (3) average customer service levels (for retailers), (4) robustness of selected objectives to demand uncertainties and fair profit distribution.	A two-phase fuzzy decision-making method	Chen and Lee (2004) extended the study of Chen *et al*. (2003) by including uncertainty in product demands and prices.	Chen *et al*. (2003b) Chen and Lee (2004)
15	Multi-product batch plant	Two cases: (a) minimization of both investment and number of different sizes for each unit operation, and (b) minimization of investment, number of different sizes for each unit operation and number of campaigns to reach steady state or oscillatory regime.	Multi-Objective GA (MOGA)	Both design and retrofit problems were studied.	Dedieu *et al*. (2003)
16	Tubular reactor-regenerator system	Simultaneous maximization of (1) profit, (2) reactant conversion and (3) selectivity of the desired product.	Ant colony method	The method is based on the cooperative search behavior of ants.	Shelokar *et al*. (2003)
17	Simulated moving bed (SMB) and Varicol processes	Simultaneous maximization of the purity of the extract and productivity of the unit.	Genetic algorithm	SMB and Varicol processes for a model chiral separation were optimized for multiple objectives and their comparative performance was discussed.	Zhang *et al*. (2003)
18	Multi-purpose batch plant	Three examples, each with objectives: (1) maximize the throughput, (2) minimize the number of equipment units, and (3) minimize the number of floors the reaction mixture has to be pumped up.	Tabu search	The three objectives were prioritized for evaluation. Performance of tabu search was compared with a multi-start steepest descent method, and found to be superior for the examples tested.	Cavin *et al*. (2004)
19	Solvent for acrylic acid-water separation by extraction	Simultaneous minimization of total annualized cost and Eco-indicator 99.	ε-constraint method	Results show that solvent substitution improves both the process economics and environmental impact of the entire plant despite its adverse effect on the extractor unit alone.	Hugo *et al.* (2004)

Table 2.1 MOO Applications in Process Design and Operation (Continued)

	Application	Objectives	Method	Comments	Reference(s)
20	Safety related decision making in chemical processes	Simultaneous minimization of (1) total safety activity cost, (2) total accident consequence, (3) number of accident scenarios for unreasonable frequency, and (4) non-operating time.	Goal programming	Example considered has 30 accident scenarios.	Kim *et al*. (2004)
21	Industrial grinding operation	Simultaneous maximization of the grinding product throughput and percent passing of one of the most important size fractions.	NSGA-II	Tournament-based constraint handling technique was used instead of penalty function.	Mitra and Gopinath (2004)
22	Waste incineration plant	Maximization of waste feed rate and minimization of carbon content in ash.	Multi-Objective GA (MOGA)	The plant was modeled using a radial basis function neural network.	Anderson *et al*. (2005)
23	Process design incorporating demand uncertainty	Simultaneous minimization of (1) capital and operating cost, (2) variance in operating cost, and (3) demand infeasibility penalty.	Line search method	Case studies considered are reactor-separator system and multi-product batch plant design. The mixed-integer nonlinear programming problems involved were solved using GAMS/SBB solver. Goyal and Ierapetritou (2004) combined the objective functions using weighted parameters.	Goyal and Ierapetritou (2004 and 2005)
24	Froth floatation circuits for mineral processing	Maximization of both the recovery of the concentrated ore and valuable mineral content in the concentrated ore.	NSGA-II with modified Jumping Gene operator	Equality constraint was imposed on total floatation cell volume.	Guria *et al*. (2005a)
		Four problems: (a) maximization of recovery of the concentrate stream and the number of non-linking streams in the circuit (N*), (b) maximization of profit and N*, (c) maximization of recovery of valuable mineral in the concentrate stream and N*, and (d) maximization of solids hold-up and N*.		Four problems were considered. One aim of the study was to simplify floatation circuits.	Guria *et al*. (2005b)
		Several problems using two to four objectives from: (1) maximization of the overall recovery, (2) maximization of the number of non-linking streams, (3) maximization of the grade, and (4) minimization of the total cell volume.		More complex floatation circuits were optimized, and several simple circuits with slightly lower recoveries were found.	Guria *et al*. (2006)

Table 2.1 MOO Applications in Process Design and Operation (Continued)

	Application	Objectives	Method	Comments	Reference(s)
25	Supply chain of vinyl chloride monomer and ethylene glycol	Two problems with two or three objectives from (1) maximization of net present value (NPV), (2) minimization of Eco-indicator 99, (3) minimization of carcinogenic plant emissions, (4) minimization network resource depletion.	ε-constraint method	The design and planning problem considers site location, raw material availability, technology and markets for the two products. The resulting problem is a multi-objective mixed-integer linear programming problem. For the application studied, Pareto curve is discontinuous, and NPV can be improved by 25% by compromising only 0.5% in the environmental impact.	Hugo and Pistikopoulos (2005)
26	Heat recovery system design in a paper mill	Minimization of (1) steam needed in summer, (2) steam needed in winter, (3) area of heat exchangers and (4) cooling/heating needed for the effluent.	NIMBUS	The process was simulated using BALAS.	Hakanen *et al*. (2005 and 2006)
27	Optimal process synthesis	Two problems: (a) chemical process optimization for maximization of net present value (NPV) while minimizing uncertainty in the future demand of two products, and (b) utility system optimization for minimization of both total annual cost and CO_2 emission.	Multi-Criteria Branch and Bound (MCBB) Algorithm	The existing MCBB algorithm was modified to increase speed, reliability and suitability for a wide range of applications.	Mavrotas and Diakoulaki (2005)
28	A co-generation plant to produce shaft power and steam	Minimization of energy loss and total cost while maximizing shaft power.	NIMBUS	The process was simulated using BALAS.	Hakanen *et al*. (2006)
29	Proportional-integral (PI) controller design	Minimization of (1) integral of time weighted absolute error (ITAE), (2) integral of square of manipulated variable changes (ISDU) and (3) settling time of a controller.	SPEA, DPEA and GSA, each combined with net flow method, for generating Pareto-optimal solutions	Single and dual population evolutionary algorithms (SPEA and DPEA) were found to be more efficient than grid search algorithm (GSA) when the optimization problem has many decision variables. DPEA was found to be more robust and faster than the other two methods.	Halsall-Whitney and Thibault (2006)

Table 2.1 MOO Applications in Process Design and Operation (Continued)

	Application	Objectives	Method	Comments	Reference(s)
30	Separation of ternary mixtures using simulated moving bed (SMB) systems	Maximization of sum of purity A and purity C, and maximization of purity B.	NSGA-II-JG	Two modified SMB configurations were optimized and compared for several situations. Kurup *et al*. (2006b) optimized a pseudo SMB system.	Kurup *et al*. (2006a and 2006b)
31	Distillation Unit	Simultaneous minimization of total annual cost and potential environmental impact.	Goal Programming	Optimization was performed during design stage.	Ramzan and Witt (2006)
32	Seeded batch crystallization process	Three problems with two or three objectives from (1) maximization of the weight mean size of the crystal size distribution, (2) minimization of the nucleated product, (3) minimization of total time of operation, and (4) minimization of coefficient of variation.	NSGA-II	Dynamic optimization problems were solved to find the optimal temperature profile.	Sarkar *et al*. (2006)
33	Industrial Ecosystems	Maximizing the profitability while minimizing environmental impact.	Hierarchical Pareto optimization	The bi-objective optimization was solved using the linear weight method. See Chapter 10 in this book for the optimization of an industrial ecosystem using NSGA-II-aJG.	Singh and Lou (2006)
34	Scheduling problems in plants	Minimization of the (1) expected makespan, (2)expected unsatisfied demands and (3) solution robustness.	A new method based on normal boundary interaction (NBI) technique	Three examples (single product production line, two products produced through 5 processing stages, and crude oil unloading and mixing problem), and uncertain demand and processing time were studied.	Jia and Ierapetritou (2007)
35	Semibatch reactive crystallization process.	Maximization of weight mean size while minimizing coefficient of variation.	NSGA-II	Dynamic optimization problems were solved to find the optimal feed addition profile.	Sarkar *et al*. (2007)

Table 2.2 MOO Applications in Biotechnology and Food industry

	Application	Objectives	Method	Comments	Reference(s)
1	Food drying	Minimization of color deviation and the unit cost of final product.	Non-preference MOO method	First principles models were employed.	Kiranoudis and Markatos (2000)
		Maximization of final product quality and minimization of drying time.	ε constraint method with SQP	Optimal trajectories of air temperature and relative humidity for drying paddy rice were determined.	Olmos *et al.* (2002)
2	Dialysis of beer to produce low-alcohol beer using hollow-fiber membrane modules	Two cases: maximization of alcohol removal from beer while minimizing (a) removal of 'taste chemicals or extract', and (b) removal of 'taste chemicals or extract' as well as cost.	NSGA	Three-objective problem was formulated as a two-objective problem using ε-constraint approach; it was then solved using NSGA. A unique solution was obtained for each value of ε.	Chan *et al.* (2000)
3	Thermal processing of food by conduction heating	Simultaneous minimization of surface cook values (i.e. maximization of final product quality) and minimization of processing time.	GA	An artificial neural network model was developed based on simulated data from the first principles model, and then used in optimization.	Chen and Ramaswamy (2002)
		Maximization of the volume average retention of thiamine for two geometries: spherical and finite cylinder, for a given boundary condition.	Modified complex method	The modified complex method was combined with the weighting method and lexicographic ordering.	Erdogdu and Balaban (2003)
4	Membrane filtration of wine	Objectives are different quality parameters and permeate filtration flux.	Minimum loss (similar to weighting) method	Three applications: champagne and wine production from different sources, were studied.	Gergely *et al.* (2003)
5	Operating conditions of gluconic acid production	Maximization of overall production rate and the final concentration of the gluconic acid while minimizing the final substrate concentration at the end of fermentation process.	Net Flow Method (NFM)	Pareto-domain was first found by a procedure which includes an evolutionary algorithm.	Halsall-Whitney *et al.* (2003)
			Two evolutionary and one grid search algorithms for finding the Pareto-optimal solutions, followed by NFM	Single and dual population evolutionary algorithms (SPEA and DPEA) were found to be more efficient than grid search algorithm (GSA) when the optimization problem has many decision variables. DPEA was found to be more robust and faster than the other two methods.	Halsall-Whitney and Thibault (2006)
6	Glucose-Fructose separation using SMB and Varicol Processes	Two cases: (a) maximization of both purity and productivity of fructose, and (b) maximization of productivity of both glucose and fructose.	NSGA	Both operation and design optimization were studied. This is one of the three applications presented in Yu *et al.* (2004).	Subramani *et al.* (2003a) Yu *et al.* (2004)

Table 2.2 MOO Applications in Biotechnology and Food industry (Continued)

	Application	Objectives	Method	Comments	Reference(s)
		Maximization of throughput and minimization of desorbent consumption.	ε constraint method	A superstructure optimization problem for SMB process is considered. An interior point optimizer (IPOPT) is used to solve the single objective sub-problems.	Kawajiri and Biegler (2006)
		Four objectives: (1) maximization of throughput, (2) minimization of solvent consumption in desorbent stream, (3) maximizing product purity, and (4) maximizing recovery of valuable component in the product stream.	NIMBUS	This study includes more objectives than the previous studies on SMB where two or three objectives were considered. As in Kawajiri and Biegler (2006), IPOPT is used to solve the single objective sub-problems in this study too.	Hakanen *et al.* (2007)
7	SMB bioreactor for high fructose syrup by glucose isomerization	Maximization of productivity of fructose and minimizing desorbent used.	NSGA-II-JG	Both operation and design of the SMB bioreactor were optimized.	Zhang *et al.* (2004)
8	Lipid production	Maximizing the productivity and yield of lipid for an optimum composition of the culture medium.	Diploid Genetic Algorithm (DGA)	Net flow algorithm was used for ranking the Pareto-optimal solutions obtained by DGA.	Muniglia *et al.* (2004)
9	SMB bioreactors for sucrose inversion to produce fructose and glucose	Maximization of production of concentrated fructose while minimizing solvent consumption.	NSGA-II-JG	Optimization was done for both an existing system and at the design stage, as well as for modified SMB bioreactors.	Kurup *et al.* (2005a)
10	*Aspergilllus niger* fermentation for catalase and protease production	Two cases: (a) maximization of catalase enzyme while minimizing protease enzyme, and (b) maximization of protease enzyme while minimizing catalase enzyme.	ε-constraint method along with differential evolution (DE)	Penalty function approach was used for constraint handling.	Mandal *et al.* (2005)
11	Fed-batch bioreactors for (a) lysine and (b) protein by recombinant bacteria	The objectives are: (1) maximization of both productivity and yield of lysine, and (2) maximization of amount of protein produced while minimizing volume of inducer added.	NSGA-II	The two applications were solved as single objective optimization problems in the earlier studies.	Sarkar and Modak (2005)
12	Batch plant design for the production of four recombinant proteins	Four cases of 2 or 3 objectives from minimization of investment and environmental impact (EI) due to biomass and EI due to solvent.	Multi-Objective GA (MOGA)	Discrete event simulator for simulating and checking the feasibility of the batch plant.	Dietz *et al.* (2006)

Table 2.2 MOO Applications in Biotechnology and Food industry (Continued)

	Application	Objectives	Method	Comments	Reference(s)
		Three cases with one or more objectives from maximization of net present value (NPV) and optimizing two other criteria: (1) production delay/advance and (2) flexibility criteria.	Multi-Objective GA (MOGA)	A fuzzy approach was proposed to account for uncertain demand in the optimization of batch plant design for multiple objectives.	Dietz *et al.* (2007)
13	Bioreactor for growing *Saccharomyces cerevisiae* in sugar cane molasses	Maximization of profit while minimizing fixed capital investment.	NSGA-II, NBI and NNC	Performance of NSGA-II, normalized boundary intersection (NBI) and normalized normal constraint (NNC) and the use of bifurcation analysis in decision making are discussed.	Sendin *et al.* (2006)
14	Penicillin V Bioreactor Train	Three cases: maximization of (a) both penicillin yield and concentration at the end of fermentation, (b) penicillin yield and batch cycle time, and (c) penicillin yield and concentration at the end of fermentation as well as profit.	NSGA-II	Glucose feed concentration is the decision variable contributing to the Pareto-optimal front. Multiple solution sets producing the same Pareto-optimal front were observed.	Lee *et al.* (2007)
15	Cattle feed manufacture.	Simultaneous minimization of (1) moisture content, (2) friability of the product and (3) energy consumption in the process.	Diploid Genetic Algorithm (DGA)	The decision variables are two operating conditions.	Mokeddem and Khellaf (2007)
16	SMB and column chromatography for plasmid DNA purification and Troger's base enantiomer separation	Maximization of productivity and minimization of solvent consumption.	NSGA	Optimization results show that SMB is better than column chromatography for the two applications studied.	Paredes and Mazzotti (2007)

2.4 Petroleum Refining and Petrochemicals

Petroleum refining and petrochemicals are the most energy and capital intensive industries in the world. Petroleum refining has evolved from a relatively simple distillation to a highly complex and integrated distillation and conversion process. It is now facing several challenges, e.g. stringent fuel quality requirements, increasing and more volatile energy prices, increasing environmental and safety concerns. Similarly, petrochemicals industry has multiple objectives, e.g. maximization of productivity, maximization of purity, minimization of utility consumption etc. These objectives are often conflicting and require some trade-off. Therefore, MOO has been successfully applied to petroleum refining and petrochemical industries for generating the trade-off solutions.

The reported applications of MOO in petroleum refining since the year 2000 include fluidized bed catalytic cracking, crude distillation, hydrocracking, heavy fuel oil blending, naphtha catalytic reforming, scheduling of refinery processes and production of gasoline (Table 2.3). Several catalytic reactors have been optimized by using NSGA-II and/or its jumping gene adaptations (Kasat *et al*., 2002; Bhutani *et al*., 2006; Sankararao and Gupta, 2007a). Hou *et al*. (2007) optimized naphtha catalytic reformer using their neighborhood and archived GA (NAGA). Song *et al*. (2002) used the ε-constraint method to optimize scheduling of refinery processes for maximum profit and minimum environmental impact.

Steam reforming for hydrogen production was optimized for multiple objectives in several studies. The first of them was the study by Rajesh *et al*. (2000) on MOO of an existing side-fired steam reformer. It was later extended to the entire hydrogen plant consisting of a steam reformer, two shift converters and other units (Rajesh *et al*., 2001). Oh *et al*. (2001) added a third objective: minimization of total heat duty of the reformer, and also considered heat flux profiles as a decision variable. Oh *et al*. (2002a and 2002b) considered design changes and a hydrogen plant using refinery off-gas as feed. Nandasana *et al*. (2003b) and Sankararao and Gupta (2006) optimized the operation of a steam reformer under dynamic conditions. Mohanty (2006) optimized synthesis gas production from combined carbon dioxide reforming and oxidation of natural gas. All of the above works used NSGA except for Nandasana

et al. (2003b) (NSGA-II) and Sankararao and Gupta (2006) (MOSA and its jumping gene adaptation).

In the petrochemicals area, styrene production was optimized in several studies (Table 2.3); Li *et al*. (2003) and Yee *et al*. (2003) optimized styrene reactors for multiple objectives by NSGA. Later, Tarafder *et al*. (2005a) optimized styrene reactors and plant by NSGA-II. Tarafder *et al*. (2007) explored several techniques to find many multiple solution sets having the same or similar objective trade-off, taking styrene and ethylene reactors as examples. Other applications include: (1) terephthalic acid production (Mu *et al*., 2003 and 2004), (2) conversion of methane and CO_2 into synthesis gas and C_2 hydrocarbons (Istadi and Amin, 2005 and 2006), (3) recovery of *p*-xylene (Kurup *et al*., 2005b), (4) ethylene production (Tarafder *et al*., 2005b), and (5) separation of ternary mixtures of xylene isomers (Kurup *et al*., 2006b and 2006c). Many of these works used NSGA-II and/or its jumping gene adaptations.

2.5 Pharmaceuticals and Other Products/Processes

Though MOO has been of interest to chemical engineers over the last two decades, their application to pharmaceuticals is relatively recent. The reported MOO applications to pharmaceuticals include separation of racemate mixtures and production of vitamin C (Table 2.4) as well as a few others included in Table 2.2. MOO applications to other products/processes include production of methyl ethyl ketone, synthesis of methyl tertiary butyl ether, pulping process, synthesis of methyl acetate ester, acetic acid recovery from aqueous waste mixture, hydrolysis of methyl acetate, desalination of brackish and sea water, and air separation by pressure swing adsorption (Table 2.4). Several of these applications involve simulated moving bed (SMB) and related technologies. Almost all studies in Table 2.4 employed first principles models including a process simulator for simulating the complete process for methyl ethyl ketone production (Lim *et al*., 2001). The MOO method used in these studies is mostly NSGA or its modification.

Table 2.3 MOO Applications in Petroleum Refining and Petrochemicals

	Application	Objectives	Method	Comments	Reference(s)
1	Industrial steam reformer	Minimization of methane feed rate and maximization of the flow rate of carbon monoxide at the reformer exit.	NSGA	Side-fired steam reformer operation was optimized.	Rajesh *et al.* (2000)
		Minimization of the cumulative deviations of flow rates of both hydrogen and steam flow rate from their steady state values.	NSGA-II	Reformer operation was simulated under dynamic conditions, and then optimized for three disturbances. Sankararao and Gupta (2006) solved the problem by MOSA and its jumping gene adaptations.	Nandasana *et al.* (2003b) Sankararao and Gupta (2006)
2	Vinyl chloride monomer (VCM) manufacture	Maximization of VCM production and minimization of environmental burden, environmental impact and operating cost simultaneously.	ε-constraint method	A design methodology consisting of 4 steps was proposed and applied to VCM plant. The steps are: (1) life cycle analysis of the process, (2) formulation of the design problem, (3) MOO, and (4) multi-criteria decision-making to find best compromise solutions.	Khan *et al.* (2001)
3	Hydrogen production	Maximization of hydrogen production and export steam flow rate.	NSGA	Oh *et al*. (2002a) studied several design and operational changes of the plant, for two objectives.	Rajesh *et al.* (2001) Oh *et al*. (2002a)
		Two problems: (a) maximization of hydrogen and export steam flow rate, and (b) maximization of hydrogen and export steam flow rate, and minimization of total heat duty of the reformer.		Oh *et al*. (2001) considered heat flux profile as a decision variable instead of furnace gas temperature in Rajesh *et al*. (2001). Oh *et al*. (2002b) optimized an industrial hydrogen plant based on refinery off-gas.	Oh *et al*. (2001) Oh *et al*. (2002b)
4	Fluidized bed catalytic cracking unit	Four problems with two or three objectives from (1) maximization of gasoline yield, (2) minimization of air flow rate, and (3) minimization of percent carbon monoxide in the flue gas.	NSGA-II	Sankararao *et al*. (2007a) solved two problems using two jumping gene adaptations of Multi-Objective Simulated Annealing (MOSA).	Kasat *et al.* (2002) Sankararao and Gupta (2007a)
		Maximization of gas yield and minimization of coke formed on the catalyst.	NSGA-II-JG	The study developed NSGA-II-JG and showed it to be better than NSGA-II for examples studied.	Kasat and Gupta (2003)

Table 2.3 MOO Applications in Petroleum Refining and Petrochemicals (Continued)

	Application	Objectives	Method	Comments	Reference(s)
5	Propylene glycol production	Maximization of economic benefit and minimization of environmental indicator. In addition process robustness measures (deviation ratio) were also considered.	Normal boundary intersection method	Sustainable process index was used as environmental indicator. Product revenue less capital and operating costs was the economic indicator	Kheawhom and Hirao (2002)
6	Scheduling of refinery processes	Maximization of total profit while minimizing total environmental impact.	ε-constraint method along with a MILP method	Scheduling of refinery processes was modeled as a mixed-integer linear programming (MILP) model.	Song *et al.* (2002)
7	Terephthalic acid (TA) production	Maximization of feed flow rate while minimizing concentration of 4-carboxy-benzaldehyde intermediate in the crude TA.	NSGA-II and Neighborhood and Archived GA (NAGA)	Mu *et al.* (2003) employed NSGA-II whereas Mu *et al.* (2004) used NAGA for four cases of operation optimization with 1 to 6 decision variables.	Mu *et al.* (2003) Mu *et al.* (2004)
8	Styrene production	Five cases using two or three objectives from (1) maximization of styrene produced, (2) maximization of styrene selectivity, (3) maximization of styrene yield, and (4) minimization of amount of steam used.	NSGA	Operation of both adiabatic and steam-injected reactors was optimized by Yee *et al.* (2003) whereas Li *et al.* (2003) optimized their design.	Li *et al.* (2003) Yee *et al.* (2003)
			Multi-objective differential evolution (MODE)	The work Babu *et al.* (2005) is very similar to that of Yee *et al.* (2003) except for different values for some model parameters (which affect the results).	Babu *et al.* (2005)
		Two cases: (a) maximization of styrene flow rate and selectivity, and (b) maximization of styrene flow rate and selectivity while minimizing total heat duty.	NSGA-II	Adiabatic, steam-injected and double-bed reactors were optimized for two objectives followed by three-objective optimization of the entire process. NSGA-II and constraint domination criterion (for constraint handling) were found to be better than NSGA and penalty function respectively. The study of Tarafder *et al.* (2007) focuses on finding many multiple solution sets to achieve the same or similar objective trade-off.	Tarafder *et al.* (2005a) Tarafder *et al.* (2007)

Table 2.3 MOO Applications in Petroleum Refining and Petrochemicals (Continued)

	Application	Objectives	Method	Comments	Reference(s)
9	Industrial crude distillation unit (CDU)	Three cases: (a) maximization of profit while minimizing energy cost, (b) maximization of total distillates produced while minimizing energy cost, and (c) maximization of profit while minimizing cumulative deviation of all the properties from the plant/desired values.	NSGA-II	CDU studied consists of the main column with four side strippers and produces six products.	Inamdar *et al.* (2004)
10	Ethanol Production	Simultaneous minimization of potential environmental impact, resource conservation and energy consumption.	Analytic hierarchy process	The three criteria were combined into a single objective: integrated environmental index, which needs to be minimized. Ethanol production via two alternative routes was compared.	Jia *et al.* (2004)
11	Gasoline production	Simultaneous maximization of total profit and negative of partial derivative of total profit with respect to three different parameters (in three cases).	Two algorithms based on simulated annealing (SA)	For the examples tested, Pareto-dominant based multi-objective SA with self-stopping gave better Pareto-optimal sets compared to an existing multi-objective SA but the former takes more iterations.	Suman (2005)
12	Catalytic CO_2 oxidative coupling of methane for the production of C_2 hydrocarbons	Three problems, each considers two objectives from maximization of (1) yield of C_2 hydrocarbons, (2) selectivity of C_2 hydrocarbons, and (3) methane conversion.	Weighted sum of squared objective functions method along with Nelder-Mead Simplex method	The process model was based on correlating experimental data on objectives with decision variables relating to catalyst composition and operating conditions. The decision variables for optimization were catalyst composition and reactor operating conditions.	Istadi and Amin (2005)
13	Recovery of *p*-xylene from a mixture of C_8 aromatics using SMB-based Parex process	Several cases of (a) maximization of recovery of *p*-xylene and minimization of desorbent consumption, and (b) maximization of both purity and recovery of *p*-xylene.	NSGA-II-JG	Varicol process was also optimized and compared with the Parex process.	Kurup *et al.* (2005b)
14	Ethylene production by steam cracking of ethane	Four cases with two or three objectives from (1) maximization of ethane conversion, (2) maximization of ethylene selectivity, and (3) maximization of ethylene flow rate.	NSGA-II	Reactor inlet temperature and length were observed to be the most important decision variables.	Tarafder *et al.* (2005b)

Table 2.3 MOO Applications in Petroleum Refining and Petrochemicals (Continued)

	Application	Objectives	Method	Comments	Reference(s)
15	Industrial hydrocracking unit	Three cases: (a) maximization of kerosene produced while minimizing hydrogen makeup, (b) maximization of diesel produced while minimizing hydrogen makeup, and (c) maximization of more valuable, heavy end products while minimizing light end products.	NSGA-II	Some parameters in the first principles model were estimated using the actual operating data.	Bhutani *et al*. (2006)
16	Dielectric barrier discharge reactor for conversion of methane and CO_2 into synthesis gas and C_{2+} hydrocarbons	Three cases: (a) maximization of methane conversion and C_{2+} selectivity, (b) maximization of methane conversion and C_{2+} yield, and (c) maximization of methane conversion and H_2 selectivity.	Weighted sum of squared objective functions method along with GA	An artificial neural network model of the process was developed based on experimental data, and then used for optimization.	Istadi and Amin (2006)
17	Separation of ternary mixtures of xylene isomers using modified simulated moving bed systems.	Maximization of sum of purities of the two streams (one containing *m*- and *o*-xylene and another containing *p*-xylene) and maximization of purity of ethyl benzene stream.	NSGA-II-JG	Several modified simulated moving bed (SMB) systems were optimized and compared in Kurup *et al*. (2006c). A pseudo SMB system was optimized by Kurup *et al*. (2006b).	Kurup *et al*. (2006b and 2006c)
18	Synthesis gas production from combined CO_2 reforming and partial oxidation of natural gas	Maximization of both methane conversion and carbon monoxide selectivity while maintaining the hydrogen to carbon monoxide ratio close to 1.	Real-coded NSGA with blend crossover	Empirical models were used for optimization.	Mohanty (2006)
19	Naphtha catalytic reforming process	Maximization of the aromatic yield and minimization of the yield of heavy aromatics.	Neighborhood and Archived GA (NAGA)	The process model is based on 20 lumps and 31 reactions. The frequency factors of 31 reactions were estimated by matching the predictions with the operating data.	Hou *et al*. (2007)
20	Heavy fuel oil blending in a petroleum refinery	Five cases with two or three objectives from: (1) maximization of profit, (2) maximization of profit/ton, (3) minimization of blend viscosity deviation from the desired value, (4) maximization of calorific value of the blend/ton, and (5) maximization of production.	NSGA-II, NSGA-II-JG and NSGA-II-aJG	The article shows that NSGA-II-aJG converges faster than NSGA-II and NSGA-II-JG for one problem.	Khosla *et al*. (2007)

Table 2.4 MOO Applications in Pharmaceuticals and Other Products/Processes

	Application	Objectives	Method	Comments	Reference(s)
1	Production of methyl ethyl ketone (MEK)	Simultaneous minimization of production cost and environmental impact (via minimization of effluent flow rates).	Combined NBI and SWOF method	The complete process for MEK production was simulated and optimized using the process simulator: ProSim.	Lim *et al.* (2001)
2	A high-yield pulping process	Maximization of both the brightness and breaking length while minimizing both specific refining energy and the amount of dichloromethane extractives.	Net flow method (NFM) and Rough set method (RSM)	NFM and RSM were compared by Renaud *et al.* (2007). Though they use the preferences provided by a decision maker separately for generating the Pareto-optimal results, the two methods gave comparable results.	Thibault *et al.* (2002)
					Thibault *et al.* (2003)
					Renaud *et al.* (2007)
3	Synthesis of MTBE using Simulated Countercurrent Moving Bed Chromatographic Reactor (SCMCR)	A few problems with two or three objectives from (1) maximization of purity and yield of methyl tertiary butyl ether (MTBE), (2) minimization of total amount of adsorbent requirement, and (3) minimization of eluent consumption.	NSGA	Subramani *et al.* (2003b) optimized both SCMCR and Varicol systems for three objectives, and compared them.	Zhang *et al.* (2002a) Subramani *et al.* (2003b)
4	Chiral separation of 1, 2, 3, 4-tetrahydro-1-naphthol racemate using simulated moving bed (SMB) and Varicol processes	Two cases: (a) maximization of purity of both the extract and raffinate streams, and (b) maximization of productivity and minimization of eluent consumption.	NSGA	This is one of the three applications described in Yu *et al.* (2004).	Zhang *et al.* (2002b) Yu *et al.* (2004)
5	Synthesis of methyl acetate ester using reactive SMB	Several problems with two or three objectives from (1) maximization of both purity and yield of methyl acetate, and (2) minimization of both eluent flow and adsorbent requirement.	NSGA	Optimization of both operation and design stages was performed, and some optimal results for operation were verified experimentally.	Yu *et al.* (2003a)
				Yu *et al.* (2003b) optimized and compared both Varicol and SMB processes. This application is one of the three presented in Yu *et al.* (2004).	Yu *et al.* (2003b) Yu *et al.* (2004).
6	Acetic acid recovery by batch distillation from aqueous waste mixtures in pharmaceutical industries	Maximization of total profit and minimization of potential environmental impact.	Parallel multi-objective steady-state GA (pMSGA)	MSGA uses a new fitness-sharing function based on Euclidean distances from an individual, and produces evenly distributed Pareto-optimal solutions. Three different feed compositions were considered.	Kim and Smith (2004)

Table 2.4 MOO Applications in Pharmaceuticals and Other Products/Processes (Continued)

	Application	Objectives	Method	Comments	Reference(s)
7	Enantioseparation of SB-553261 racemate using SMB technology	Three cases: (a) maximizing the purity and productivity of raffinate stream, (b) maximization of purity and productivity of extract stream, and (c) maximization of feed flow rate and minimization of desorbent flow rate.	NSGA-II-JG	Both SMB and Varicol processes were optimized, and the study found that the latter has superior performance.	Wongso *et al.* (2004)
8	Desalination of brackish and sea water using spiral wound or tubular module	Two cases: (a) maximization of the permeate throughput and minimization of cost of desalination, and (b) maximizing the permeate throughput while minimizing the cost as well as permeate concentration.	NSGA-II, NSGA-II-JG and NSGA-II-aJG	Both operation and design optimization were studied. NSGA-II-aJG was observed to be the fastest of the three algorithms.	Guria *et al.* (2005c)
9	Hydrolysis of methyl acetate for producing methanol and acetic acid using SMB and Varicol Process	Two cases: (a) maximization of purity of both raffinate and extract streams, and (b) maximization of yield of both raffinate and extract streams.	NSGA	The study found that the reactive Varicol performs better than SMB reactor for the application studied.	Yu *et al.* (2005)
10	Production of vitamin C	Maximization of both net present value and productivity while optimizing: (1) batch size, (2) no. of equipment units, (3) campaign costs, (4) floors-up indicator etc.	Tabu search	An abstract equipment model (namely, super-equipment) that can perform any physico-chemical batch operation is introduced and used in the study.	Mosat *et al.* (2007)
11	Air separation by pressure swing adsorption for producing oxygen- and nitrogen-rich mixtures	Three cases: (a) maximization of both purity and recovery of oxygen in raffinate, (b) maximization of both purity and recovery of nitrogen in extract, and (c) maximization of purity and recovery of oxygen in raffinate and of purity and recovery of nitrogen in extract.	MOSA-aJG	Two parameters in the process model were tuned using one set of experimental data. Then the model was employed in the MOO.	Sankararao and Gupta (2007b)

2.6 Polymerization

Polymerization processes are usually complex in nature, and have multiple and conflicting objectives. Understandably, MOO has found 11 applications for optimizing polymerization processes since the year 2000 (Table 2.5). These are wiped film poly(ethylene terephthalate) reactor, continuous casting process for poly(methyl methacrylate), styrene polymerization, epoxy polymerization, styrene-acrylonitrile copolymerization, ethylene-poly(oxyethylene terephthalate) copolymerization, poly(propylene terephthalate) and low-density polyethylene reactor. Decision variables in some of these applications are trajectories/profiles of operating variables. NSGA, NSGA-II and their jumping gene adaptations have been used in optimizing polymerization processes for multiple objectives. On the other hand, Silva and Biscaia (2003 and 2004), Massebeuf *et al*. (2003) and Fonteix *et al*. (2004) have proposed GA-based methods and used them for optimizing the selected polymerization applications.

2.7 Conclusions

MOO applications in chemical engineering included in this chapter show that nearly hundred applications were studied by researchers and reported in more than 130 journal publications since the year 2000. Hence, on average, about 15 new applications of MOO in chemical engineering have been reported every year since the year 2000. These applications are from several industry sectors and areas of interest to chemical engineers. Many of them were modeled using first principles models and employed two to three objectives. MOO applications in chemical engineering in future can be expected to be more complex – complete plants, dynamic optimization, many objectives, uncertain parameters/data etc. GA-based MOO (namely, NSGA, NSGA-II, NSGA-II-JG and MOGA) were the most popular for solving the chemical engineering applications. This may be because of their ready availability, effectiveness to find Pareto-optimal solutions and researchers' experience. Many studies in chemical engineering focused on finding Pareto-optimal solutions and only a few studies considered ranking and selecting one or a few Pareto-optimal solutions for implementation. More emphasis and studies on ranking and selection from among the Pareto-optimal solutions are expected in the future.

Table 2.5 MOO Applications in Polymerization

	Application	Objectives	Method	Comments	Reference(s)
1.	An industrial wiped film poly(ethylene terephthalate) (PET) reactor	Simultaneous minimization of the acid and the vinyl end groups concentration in the product.	NSGA	When the temperature is a decision variable, a global unique solution was obtained (Bhaskar *et al.*, 2000b) while Pareto-optimal solutions were obtained when the temperature is kept constant (Bhaskar *et al.*, 2001).	Bhaskar *et al.* (2000b and 2001)
2	Continuous casting process for poly(methyl methacrylate) (PMMA)	Maximization of average monomer conversion in the product while minimizing length of the film reactor.	NSGA	The optimization problem includes an end-point constraint on the number-average molecular weight of the polymer produced.	Zhou *et al.* (2000)
		Simultaneous minimization of (1) reaction time, (2) polydispersity index for desired value of monomer conversion and (3) degree of polymerization.	Weighting method	The resulting single objective optimization problem was solved by SQP, GA and a hybrid of the two. Based on more than 100 optimization runs, all the three methods were concluded to be trustworthy.	Curteanu *et al.* (2006)
3	Batch free radical polymerization of styrene	Two cases: (a) minimization of both the deviations from the desired conversion and number-average molecular weight, and (b) minimization of the initiator concentration in the product and the deviation from the desired conversion.	ε-constraint method	Three problems, each with a different set of desired values for conversion and number-average molecular weight, were solved.	Merquior *et al.* (2001)
		Minimization of the initiator concentration in the product and the deviation from the desired conversion.	An improved GA for MOO	Silva and Biscaia *et al.* (2004) optimized a semi-batch reactor.	Silva and Biscaia (2003 and 2004)
4	Emulsion homo-polymerization of styrene	Maximization of monomer conversion and minimization of deviation of average molecular weight and particle size from desired values.	Adapted GA for MOO	Model parameters were estimated based on experimental data. A decision support system was also developed to rank the Pareto-optimal solutions.	Massebeuf *et al.* (2003)
		Maximization of production while minimizing number of particles per liter and weight average molecular weight.	A new method based on diploid GA		Fonteix *et al.* (2004)
5	Polystyrene using the continuous tower process	Maximization of the monomer conversion and minimization of the polydispersity index of the product.	NSGA-II	A unique solution was obtained instead of a Pareto-optimal set.	Bhat *et al.* (2004)

Table 2.5 MOO Applications in Polymerization (Continued)

	Application	Objectives	Method	Comments	Reference(s)
6	Semi-batch epoxy polymerization process	Simultaneous maximization of number-average molecular weight and minimization of reaction time.	NSGA-II	The MOO problem includes a constraint on the desired polydispersity index.	Mitra *et al.* (2004a)
		Three cases: (a) maximization of number-average molecular weight and minimization of polydispersity index, (b) maximization of concentration of the most desired species and minimization of chain propagation, and (c) maximization of concentration of the most desired species and minimization of both chain propagation and the total amount of sodium hydroxide added.		Majumdar *et al*. (2005a) optimized the process for two advanced cases: (a) maximization of selective growth of a particular polymer species, minimization of both polymer processing time and chain length, and (b) minimization of total sodium hydroxide added, polymer processing time and chain length.	Mitra *et al.* (2004b) Majumdar *et al*. (2005a)
		Three cases using two or three objectives from (1) maximization of number-average molecular weight, (2) minimization of reaction time, and (3) minimization of polydispersity index.		In all three cases, Pareto-optimal front was found to be non-convex. Real-coded NSGA-II was observed to be slightly better than binary-coded NSGA-II.	Deb *et al*. (2004)
7	Styrene-acrylonitrile copolymerization in a semi-batch reactor	Two cases: (a) maximization of monomer conversion and minimization of polydispersity index at the end of reaction, and (b) maximization of monomer conversion at the end of reaction while minimizing polydispersity index at the end of reaction and molar ratio of unreacted monomer in the reactor at any time.	NSGA-II	The decision variables for optimization were trajectories of addition rate of a monomer-solvent-initiator mixture and reactor temperature.	Nayak and Gupta (2004)
8	Batch copoly (ethylene-polyoxyethylene terephthalate) reactor	Minimization of both reaction time and undesired side products.	NSGA-II, NSGA-II-JG and NSGA-II-aJG	At near-optimal solutions, NSGA-II-JG was observed to be faster than the other two methods.	Kachhap and Guria (2005)
9	Catalytic esterification of poly(propylene terephthalate)	Maximization of both degree of polymerization and the desired functional group while minimizing processing time for esterification.	NSGA- II	Kinetic parameters were estimated using a simple GA. Decision variables for MOO were trajectories of addition rates of terephthalic acid and propylene glycol.	Majumdar *et al*. (2005b)

Table 2.5 MOO Applications in Polymerization (Continued)

	Application	Objectives	Method	Comments	Reference(s)
10	Styrene polymerization	Minimization of total annual cost and integral of square of the error in the number-average molecular weight of the product during grade transition.	ε-constraint method with gPROMS/gOPT software	Design and control (during polymer grade transition) of a continuous stirred tank reactor for styrene polymerization was optimized. Three different scenarios were studied.	Asteasuain *et al*. (2006)
11	Industrial low-density polyethylene tubular reactor.	Maximization of monomer conversion and minimization of the concentration of the undesirable side products (methyl, vinyl and vinylidene groups).	NSGA-II, NSGA-II-JG and NSGA-II-aJG, all binary-coded	Agrawal *et al*. (2006) presented a detailed model with all parameter values. Instead of a hard equality constraint, a softer constraint was used for obtaining Pareto-optimal solutions. NSGA-II-aJG and NSGA-II-JG performed better than NSGA-II near the hard end-point constraints. Agrawal *et al*. (2007) optimized the reactor design, and also found that constrained-dominance principle is marginally better than penalty function for handling constraints.	Agrawal *et al*. (2006)
		Two cases: (a) two objectives same as above, and (b) two objectives same as above and minimization of compression power.			Agrawal *et al*. (2007)

References

Agrawal, N., Rangaiah, G. P., Ray, A. K. and Gupta, S. K. (2006). Multi-objective optimization of the operation of an industrial low-density polyethylene tubular reactor using genetic algorithm and its jumping gene adaptations, *Ind. Eng. Chem. Res.*, 45, pp. 3182-3199.

Agrawal, N., Rangaiah G.P., Ray A.K. and Gupta S.K. (2007). Design stage optimization of an industrial low-density polyethylene tubular reactor for multiple objectives using NSGA-II and its jumping gene adaptations, *Chem .Eng. Sci.*, 62, pp. 2346-2365.

Anderson, S.R., Kadirkamanathan, V., Chipperfield, A., Sharifi, V. and Swithenbank, J. (2005). Multi-objective optimization of operational variables in a waste incineration plant. *Comput. Chem. Eng.*, 29, pp.1121–1130.

Asteasuain, M., Bandoni A., Sarmoria, C and Brandolin, A. (2006). Simultaneous process and control system design for grade transition in styrene polymerization, *Chem .Eng. Sci.*, 61, pp. 3362-3378.

Babu, B. V., Chakole, P. G. and Mubeen, J. H. S. (2005). Multi-objective differential evolution (MODE) for optimization of adiabatic styrene reactor, *Chem .Eng. Sci.*, 60, pp. 4822-4837.

Bhaskar, V., Gupta, S. K. and Ray, A. K. (2000a). Applications of multi-objective optimization in chemical engineering, *Rev. Chem. Eng.*, 16, pp. 1-54.

Bhaskar, V., Gupta, S. K. and Ray, A. K. (2000b). Multi-objective optimization of an industrial wiped-film pet reactor, *AIChE J.*, 46, pp. 1046-1058.

Bhaskar, V., Gupta, S. K. and Ray, A. K. (2001). Multi-objective optimization of an industrial wiped film poly (ethylene terephthalate) reactor: some further insights, *Comput. Chem. Eng.*, 25, pp. 391-407.

Bhat, S. A., Sharma, R. and Gupta, S. K. (2004). Simulation and optimization of the continuous tower process for styrene polymerization, *J. Appl. Polym. Sci.*, 94, pp. 775-788.

Bhutani, N., Ray, A. K. and Rangaiah, G. P. (2006). Modeling, simulation, and multi objective optimization of an industrial hydrocracking unit, *Ind. Eng. Chem. Res.*, 45, pp. 1354-1372.

Blanco, A. M. and Bandoni, J. A. (2003). Design for operability: a singular-value optimization approach within a multi-objective framework, *Ind. Eng. Chem. Res.*, 42, pp. 4340-4347.

Cavin, L., Fischer, U., Glover, F. and Hungerbühler, K. (2004). Multi-objective process design in multi-purpose batch plants using tabu search optimization algorithm, *Comput. Chem. Eng.*, 28, pp. 459-478.

Chakraborty, A. and Linninger, A. A. (2002). Plant-wide waste management.1. Synthesis and multi-objective design, *Ind. Eng. Chem. Res.*, 41, pp. 4591-4604.

Chakraborty, A. and Linninger, A. A. (2003). Plant-wide waste management.2. Decision making under uncertainty, *Ind. Eng. Chem. Res.*, 42, pp. 357-369.

Chan, C. Y., Aatmeeyata, Gupta, S. K. and Ray, A. K. (2000). Multi-objective optimization of membrane separation modules using genetic algorithm, *J.Memb. Sci.*, 176, pp. 177–196.

Chen, C. R. and Ramaswamy, H. S. (2002). Modeling and optimization of variable retort temperature (VRT) thermal processing using coupled neural networks and genetic algorithms, *J.Food Eng.*, 53, pp. 209-220.

Chen, C. L. and Lee, W. C. (2004). Multi-objective optimization of multi echelon supply chain networks with uncertain product demands and prices, *Comput. Chem. Eng.*, 28, pp. 1131-1144.

Chen, H., Wen, Y., Waters, M. D. and Shonnard, D. R. (2002). Design guidance for chemical processes using environmental and economic assessments, *Ind. Eng. Chem. Res.*, 41, pp. 4503-4513.

Chen, H., Rogers, T. N., Barna, B. A. and Shonnard, D. R. (2003a). Automating hierarchical environmentally-conscious design using integrated software: VOC Recovery Case Study, *Environ. Prog.*, Vol. 22, No. 3, pp. 147-160.

Chen, C. L., Wang, B. W. and Lee, W. C. (2003b). Multiobjective optimization for a multienterprise supply chain network, *Ind. Eng. Chem. Res.*, 42, pp. 1879-1889.

Curteanu, S., Leon, F., Galea, D. (2006). Alternatives for multi-objective optimization of a polymerization process, *J. Appl. Polym. Sci.*, 100, pp. 3680-3695.

Deb, K., Mitra, K., Dewri, R. and Majumdar, S. (2004). Towards a better understanding of the epoxy-polymerization process using multi-objective evolutionary computation, *Chem .Eng. Sci.*, 59, pp. 4261-4277.

Dedieu, S., Pebouleau, L., Pantel, C. A. and Domenech, S. (2003). Design and retrofit of multi-objective batch plants via a multicriteria genetic algorithm, *Comput. Chem. Eng.*, 27, pp. 1723-1740.

Dietz, A., Pebouleau, L., Pantel, C. A. and Domenech, S. (2006). Multi-objective optimization for multiproduct batch plant design under economic and environmental considerations, *Comput. Chem. Eng.*, 30, pp. 599-613.

Dietz, A., Lasserre, A., Pantel, C., Pebouleau, L. and Domenech, S. (2007). A fuzzy multi-objective algorithm for multiproduct batch plant: Application to protein production, *Comput. Chem. Eng.*, in press.

Erdogdu, F and Balaban, M. O. (2003). Complex method for nonlinear constrained multi criteria (multi-objective function) optimization of thermal processing, *J. Food Process Eng.*, 26, pp. 357-375.

Fonteix, C., Massebeuf, S., Pla, F. and Kiss, L. N. (2004). Multicriteria optimization of an emulsion polymerization process, *Eur. J. Oper. Res.*, 153, pp. 350-359.

Gergely, S., Molnar, E. B. and Vatai, G. (2003). The use of multi-objective optimization to improve wine filtration, *J.Food Eng.*, 58, pp. 311-316.

Goyal, V. and Ierapetritou, M. G. (2004). Deterministic framework for robust modular design with integrated-demand data analysis, *Ind. Eng. Chem. Res.,* 43, pp. 6813-6821.

Goyal, V. and Ierapetritou, M. G. (2005). Multi-objective framework for modular design generation incorporating demand uncertainty, *Ind. Eng. Chem. Res.,* 44, pp. 3594-3606.

Guria, C., Verma, M., Mehrotra, S. P. and Gupta, S. K. (2005a). Multi-objective optimal synthesis and design of froth flotation circuits for mineral processing, using the jumping gene adaptation of genetic algorithm, *Ind. Eng. Chem. Res.*, 44, pp. 2621-2633.

Guria, C., Verma, M., Gupta, S. K. and Mehrotra, S. P. (2005b). Simultaneous optimization of the performance of flotation circuits and their simplification using the jumping gene adaptations of genetic algorithm, *Int. J. Miner. Process,* 77, pp. 165-185.

Guria, C., Bhattacharya, P. K. and Gupta, S. K. (2005c). Multi-objective optimization of reverse osmosis desalination units using different adaptations of the non-dominated sorting genetic algorithm (NSGA), *Comput. Chem. Eng.*,29, pp. 1977-1995.

Guria, C., Verma, M., Mehrotra, S. P. and Gupta, S. K. (2006). Simultaneous optimization of the performance of flotation circuits and their simplification using the jumping gene adaptations of genetic algorithm-II: More complex problems, *Int. J. Miner. Process,* 79, pp. 149-166.

Hakanen, J., Miettinen, K., Makela, M. M. and Manninen, J. (2005). On interactive multi-objective optimization with NIMBUS in chemical process design, *J. Multi-Crit. Decis. Anal.*, 13, pp. 125-134.

Hakanen, J., Hakala, J. and Manninen, J. (2006). An integrated multi-objective design tool for process design*, Appl. Therm. Eng.*, 26, 1393-1399.

Hakanen, J., Kawaziri, Y., Miettinen, K., Biegler, L. T. (2007). Interactive multi-objective optimization for simulated moving bed processes, *Control Cybern.*, 36, 1-20.

Halsall-Whitney, H. and Thibault, J. (2006). Multi-objective optimization for chemical processes and controller design: Approximating and classifying the Pareto domain, *Comput. Chem. Eng.*, 30, pp. 1155-1168.

Halsall-Whitney, H., Taylor, D. and Thibault, J. (2003). Multicriteria optimization of gluconic acid production using net flow, *Bioprocess Biosyst. Eng.*, 25, pp. 299-307.

Hoffmann, V. H., Hungerbühler, K. and McRae, G. J. (2001). Multi-objective Screening and Evaluation of Chemical Process Technologies, *Ind. Eng. Chem. Res*., 40, pp. 4513-4524.

Hoffmann, V. H., McRae, G. J. and Hungerbühler, K. (2004). Methodology for early-stage technology assessment and decision making under uncertainty: application to the selection of chemical processes, *Ind. Eng. Chem. Res*., 43, pp. 4337-4349.

Hou W.F., Su H.Y., Mu S.J. and Chu J. (2007). Multi-objective optimization of the industrial naphtha catalytic reforming process, *Chinese. J. Chem. Eng*., 15, pp. 75-80.

Hugo, A. and Pistikopoulos, E. N. (2005). Environmentally conscious long-range planning and design of supply chain networks, *J. Clean. Prod*, 13, pp. 1471-1491.

Hugo, A., Ciumei, C., Buxton, A. and Pistikopoulos, E. N. (2004). Environmental impact minimization through material substitution: a multi-objective optimization approach, *Green Chem.*, 6, pp. 407-417.

Inamdar, S. V., Gupta, S. K. and Saraf, D. N. (2004). Multi-objective optimization of an industrial crude distillation unit using the elitist nondominated sorting genetic algorithm, *Chem. Eng. Res. Des.*, 82-A, pp. 611-623.

Istadi and Amin, N. A. S. (2005). A hybrid numerical approach for multi-responses optimization of process parameters and catalyst compositions in CO_2 OCM process over CaO-MnO/CeO_2 catalyst, *Chem. Eng. J.,* 106, pp. 213-227.

Istadi and Amin, N. A. S. (2006). Hybrid Artificial Neural Network-Genetic Algorithm Technique for Modeling and Optimization of Plasma Reactor, *Ind. Eng. Chem. Res.*, 45, pp. 6655-6664.

Jia, Z. and Ierapetritou, M. G. (2007). Generate Pareto optimal solutions of scheduling problems using normal boundary intersection technique, *Comput. Chem. Eng.*, 31, pp. 268-280.

Jia, X. P., Han, F. Y. and Tan, X. S. (2004). Integrated environmental performance assessment of chemical processes, *Comput. Chem. Eng*., 29, pp. 243-247.

Kachhap, R. and Guria, C. (2005). Multi-objective optimization of a batch copoly (ethylene-polyoxyethylene terephthalate) reactor using different adaptations of nondominated sorting genetic algorithm, *Macromol. Theor. Simul*., 14, pp. 358-373.

Kasat, R. B. and Gupta, S. K. (2003). Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator, *Comput. Chem. Eng.*, 27, pp. 1785-1800.

Kasat, R. B., Kunzru, D., Saraf, D. N. and Gupta, S. K. (2002). Multi-objective optimization of industrial FCC units using elitist nondominated sorting genetic algorithm, *Ind. Eng. Chem. Res*., 41, pp. 4765-4776.

Kasat, R. B., Ray, A. K. and Gupta, S. K. (2003). Applications of genetic algorithm in polymer science and engineering, *Mater. Manuf. Processes*, 18, pp. 523-532.

Kawajiri, Y. and Biegler, L. T. (2006). Nonlinear programming superstructure for optimal dynamic operations of Simulated Moving Bed processes, *Ind. Eng. Chem. Res*., 45, pp. 8503-8513.

Khan, F. I., Natrajan, B. R. and Revathi, P. (2001). GreenPro: a new methodology for cleaner and greener process design, *J. Loss Prevent Proc.*, 14, pp. 307-328.

Kheawhom, S. and Hirao, M. (2002). Decision support tools for process design and selection, *Comput. Chem. Eng.*, 26, pp. 747-755.

Kheawhom, S. and Hirao, M. (2004). Decision support tools for environmentally benign process design under uncertainty, *Comput. Chem. Eng*., 28, pp. 1715-1723.

Khosla, D. K., Gupta, S. K. and Saraf, D. N. (2007). Multi-objective optimization of fuel oil blending using the jumping gene adaptation of genetic algorithm, *Fuel Process. Technol.*, 88, pp. 51-63.

Kim, K. J. and Diwekar, U. M. (2002). Integrated solvent selection and recycling for continuous processes, *Ind. Eng. Chem. Res*., 41, pp. 4479-4488.

Kim, K. J. and Smith, R. L. (2004). Parallel multi-objective evolutionary algorithms for waste solvent recycling, *Ind. Eng. Chem. Res*., 43, pp. 2669-2679.

Kim, D., Yeo, Y. and Moon, I. (2004). Multi-objective optimization for safety-related decision making in chemical processes, *J. Chem. Eng. Jpn.*, 37, pp. 332-337.

Kiranoudis, C. T and Markatos N. C. (2000). Pareto design of conveyor belt dryers, *J.Food Eng.*, 46, pp. 145-155.

Ko, D. and Moon, I. (2002). Multi-objective optimization of cyclic adsorption processes, *Ind. Eng. Chem. Res.*, 41, pp. 93-104.

Krokida, M. K. and Kiranoudis, C. T. (2000). Pareto design of fluidized bed dryers, *Chem. Eng. J.,* 79, pp. 1-12.

Kurup, S. K., Subramani, H. J., Hidajat, K. and Ray, A. K. (2005a). Optimal design and operation of SMB bioreactor for sucrose inversion, *Chem. Eng. J.,*108, pp. 19–33.

Kurup, A. S., Hidajat, K. and Ray, A. K. (2005b). Optimal operation of an industrial scale Parex process for the recovery of *p*-xylene from a mixture of C_8 aromatics, *Ind. Eng. Chem. Res*., 44, pp. 5703-5714.

Kurup, A. S., Hidajat, K. and Ray, A. K. (2006a). Comparative study of modified simulated moving bed systems at optimal conditions for the separation of ternary mixtures under nonideal conditions, *Ind. Eng. Chem. Res*., 45, pp. 3902-3915.

Kurup, A. S., Hidajat, K. and Ray, A. K. (2006b). Optimal operation of a Pseudo SMB process for ternary separation under non-ideal conditions, *Sep. Purif. Technol.*, 51, pp. 387-403.

Kurup, A. S., Hidajat, K. and Ray, A. K. (2006c). Comparative study of modified simulated moving bed systems at optimal conditions for the separation of ternary mixtures of xylene isomers, *Ind. Eng. Chem. Res*., 45, pp. 6251-6265.

Lee, F. C., Rangaiah, G. P. and Ray, A.K. (2007). Multi-objective optimization of an industrial Penicillin V bioreactor train using non-dominated sorting genetic algorithm, *Biotechnol. Bioeng*. 98, pp. 586-598.

Li, Y., Rangaiah, G. P. and Ray, A. K. (2003). Optimization of styrene reactor design for two objectives using a genetic algorithm, *Int. J. Chem. Reactor Eng.*, 1, pp. 1-14.

Lim, Y, I., Floquet, P. and Joulia, X. (2001). Efficient implementation of the normal boundary intersection (NBI) method on Multi-objective optimization problems, *Ind. Eng. Chem. Res*., 40, pp. 648-655.

Majumdar, S., Mitra, K. and Raha, S. (2005a). Optimized species growth in epoxy polymerization with real-coded NSGA-II, *Polymer*, 46, pp. 11858-11869.

Majumdar, S., Mitra, K., Sardar, G. (2005b). Kinetic analysis and optimization for the catalytic esterification step of PPT polymerization, *Macromol. Theor. Simul*., 14, pp. 49-59.

Mandal, C., Gudi, R. D. and Shuraishk, G. K. (2005). Multi-objective optimization in *Aspergillus niger* fermentation for selective product enhancement, *Bioprocess Biosyst. Eng*., 28, pp. 149–164.

Massebeuf, S., Fonteix, C., Hoppe, S. and Pla, F. (2003). Development of new concepts for the control of polymerization processes: multiobjective optimization and decision engineering. I. Application to emulsion homopolymerization of styrene, *J. Appl. Polym. Sci.*, 87, pp. 2383-2396.

Mavrotas, G. and Diakoulaki, D. (2005). Solving multiobjective MILP problems in process synthesis using multi-criteria branch and bound algorithm, *Chem. Eng. Technol*., 28, pp. 1500-1510.

Merquior, D. M., Fontoura, J. M. R., Pinto, J. C. and Lima, E. L. (2001). Studies of multiobjective optimization of batch free-radical polymerization process, *Latin Am. Appl. Res.*, 31, pp. 513-517.

Mitra, K. and Gopinath, R. (2004). Multi-objective optimization of an industrial grinding operation using elitist nondominated sorting genetic algorithm, *Chem .Eng. Sci.*, 59, pp. 385-396.

Mitra, K., Majumdar, S. and Raha, S. (2004a). Multi-objective dynamic optimization of a semi-batch epoxy polymerization process, *Comput. Chem. Eng.*, 28, pp. 2583-2594.

Mitra, K., Majumdar, S. and Raha, S. (2004b). Multi-objective optimization of a semibatch epoxy polymerization process using the elitist genetic algorithm, *Ind. Eng. Chem. Res*., 43, pp. 6055-6063.

Mohanty, S. (2006). Multi-objective optimization of synthesis gas production using non-dominated sorting genetic algorithm, *Comput. Chem. Eng.*,30, pp. 1019-1025.

Mokeddem, D. and Khellaf, A. (2007). Pareto-optimal solutions for multicriteria optimization of a chemical engineering process using a diploid genetic algorithm, *Comput. Chem. Eng.*, in press. doi:10.1016/j.compchemeng.2006.12.006.

Mosat, A., Cavin, L., Fischer, U. and Hungerbuhler, K. (2007). Multi-objective optimization of multipurpose batch plants using superequipment class concept, *Comput. Chem. Eng.*, in press. doi:10.1016/j.compchemeng.2007.03.017.

Mu, S. J., Su, H. Y., Gu, Y. and Chu, J. (2003). Multi-objective optimization of industrial purified terephthalic acid (PTA) oxidation process, *Chinese J. Chem. Eng.*, 11, pp. 536-541.

Mu, S. J., Su, H., Jia, T., Gu, Y. and Chu, J. (2004). Scalable multi-objective optimization of industrial purified terephthalic acid (PTA) oxidation process, *Comput. Chem. Eng*.,28, pp. 2219–2231.

Muniglia, L., Kiss, L. N., Fonteix, C. and Marc, I. (2004). Multicriteria optimization of a single-cell oil production, *Eur. J. Oper. Res.*, 153, pp. 360-369.

Nandasana, A. D., Ray, A. K. and Gupta, S. K. (2003a). Applications of the non-dominated sorting genetic algorithm (NSGA) in chemical reaction engineering, *Int. J. Chem. Reactor Eng.*, 1, R2, pp. 1-16.

Nandasana, A. D., Ray, A. K., Gupta, S. K. (2003b). Dynamic model of an industrial steam reformer and its use for multi-objective optimization, *Ind. Eng. Chem. Res*., 42, pp. 4028-4042.

Nayak, A. and Gupta, S. K. (2004). Multi-objective optimization of semi-batch copolymerization reactors using adaptations of genetic algorithm, *Macromol. Theor. Simul.*, 13, pp. 73-85.

Oh, P. P., Ray, A. K. and Rangaiah, G. P. (2001). Triple-objective optimization of an industrial Hydrogen plant, *J. Chem. eng. Jpn.,* 34, pp. 1341-1355.

Oh, P. P., Ray, A. K. and Rangaiah, G. P. (2002a). Optimal design and operation of an industrial hydrogen plant for multiple objectives, in "Recent developments in optimization and optimal control in chemical engineering" edited by R. Luus, *Research Signpost, Trivandrum, India*, pp. 289-306.

Oh, P. P., Rangaiah, G. P. and Ray, A. K. (2002b). Simulation and multi-objective optimization of an industrial hydrogen plant based on refinery off gas, *Ind. Eng. Chem. Res.*, 41, pp. 2248-2261.

Olmos, A., Trelea, I. C., Courtois, F., Bonazzi, C. and Trystram, G. (2002). Dynamic optimal control of batch rice drying process, *Dry. Technol.*, 20, pp. 1319-1345.

Paredes, J. and Mazzotti, M. (2007). Optimization of simulated moving bed and column chromatography for a plasmid DNA purification step and for a chiral separation, *J. Chromatogr. A*., 1142, pp. 56–68.

Rajesh, J. K., Gupta, S. K., Rangaiah, G. P. and Ray, A. K. (2000). Multi-objective optimization of steam reformer performance using genetic algorithm, *Ind. Eng. Chem. Res*., 39, pp. 706-717.

Rajesh, J. K., Gupta, S. K., Rangaiah, G. P. and Ray, A. K. (2001). Multi-objective optimization of industrial hydrogen plants, *Chem .Eng. Sci.*, 56, pp. 999-1010.

Ramzan, N. and Witt, W. (2006). Multi-objective optimization in distillation unit: a case study, *Can. J. Chem. Eng.,* 84, pp. 604-613.

Ravi, G., Gupta, S. K. and Ray, M. B. (2000). Multi-objective optimization of cyclone separators using genetic algorithm, *Ind. Eng. Chem. Res.*, 39, 4272-4286.

Ravi, G., Gupta, S. K., Viswanathan, S. and Ray, M. B. (2002). optimization of venturi scrubbers using genetic algorithm, *Ind. Eng. Chem. Res.*, 41, pp. 2988-3002.

Ravi, G., Viswanathan, S., Gupta, S. K and Ray, M. B. (2003). Multi-objective optimization of venture scrubbers using a three-dimensional model for collection efficiency, *J. Chem. Technol. Biot.*, 78, pp. 308-313.

Renaud, J., Thibault, J., Lanouette, R., Kiss, L. N., Zaras, K. and Fonteix., C. (2007). Comparison of two multicriteria decision aid methods: Net Flow and Rough Set methods in a high yield pulping process, *Eur. J. Oper. Res.*, 177. pp. 1418-1432.

Sankararao, B. and Gupta, S. K. (2006). Multi-objective optimization of the dynamic operation of an industrial steam reformer using the jumping gene adaptations of simulated annealing, *Asia Pac. J. Chem. Eng.*, 1, pp. 21-31.

Sankararao, B. and Gupta, S. K. (2007a). Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using two jumping gene adaptations of simulated annealing, *Comput. Chem. Eng.*, 31, pp. 1496-1515.

Sankararao, B. and Gupta, S. K. (2007b). Multi-objective optimization of pressure swing adsorbers for air separation, *Ind. Eng. Chem. Res.*, 46, pp. 3751-3765.

Sarkar, D. and Modak, J. M. (2005). Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm, *Chem .Eng. Sci.*, 60, pp. 481–492.

Sarkar, D., Rohani, S. and Jutan, A. (2006). Multi-objective optimization of seeded batch crystallization processes, *Chem .Eng. Sci.*, 61, pp. 5282-5295.

Sarkar, D., Rohani, S. and Jutan, A. (2007). Multi-objective optimization of semibatch reactive crystallization processes, *AIChE J.*, 53, pp. 1164-1177.

Sendin, J. O. H., Muras, I. O., Alonso, A. A. and Banga, J. R. (2006). Improved optimization methods for the multi-objective design of bioprocesses, *Ind. Eng. Chem. Res.*, 45, pp. 8594-8603.

Shelokar, P. S., Jayaraman, V. K. and Kulkarni, B. D. (2003). Multi-objective optimization of reactor–regenerator system using ant algorithm, *Pet. Sci. Technol.*, 21, pp. 1167-1184.

Silva, C. M. and Biscaia, E. C. (2003). Genetic algorithm development for multi-objective optimization of batch free-radical polymerization reactors, *Comput. Chem. Eng.*, 27, pp.1329-1344.

Silva, C. M. and Biscaia, E. C. (2004). Multi-objective dynamic optimization of semi-batch polymerization processes, *Macromol. Symp.*, 206, pp. 291-306.

Singh, A. and Lou, H. H. (2006). Hierarchical Pareto Optimization for the Sustainable Development of Industrial Ecosystems, *Ind. Eng. Chem. Res.*, 45, pp. 3265-3279.

Song, J., Park, H., Lee, D. Y., Park, S. (2002). Scheduling of actual size refinery processes considering environmental impacts with multi-objective optimization, *Ind. Eng. Chem. Res.*, 41, pp. 4794-4806.

Subramani, H. J., Hidajat, K. and Ray, A. K. (2003a). Optimization of simulated moving bed Varicol processes for glucose-fructose separation, *Trans IChemE.*, 81, pp. 549-567.

Subramani, H. J., Hidajat, K. and Ray, A. K. (2003b).Optimization of reactive SMB and Varicol systems, *Comput. Chem. Eng.*,27, pp. 1883-1901.

Suman, B. (2003). Simulated annealing based multi-objective algorithms and their application for system reliability, *Eng. Optimiz.*, 35, pp. 391-416.

Suman, B. (2005). Study of self-stopping PDMOSA and performance measure in multi-objective optimization, 29, *Comput. Chem. Eng.*, pp. 1131-1147.

Tarafder, A., Rangaigh, G. P. and Ray, A. K (2005a). Multi-objective optimization of an industrial styrene monomer manufacturing process, *Chem .Eng. Sci.*, 60, pp. 347-363.

Tarafder, A., Lee, B. C. S., Ray, A. K. and Rangaiah, G. P. (2005b). Multi-objective optimization of an industrial ethylene reactor using a nondominated sorting genetic algorithm, *Ind. Eng. Chem. Res.*, 44, pp. 124-141.

Tarafder, A., Rangaigh, G. P. and Ray, A. K (2007). A study of finding many desirable solutions in multi-objective optimization of chemical processes, *Chem .Eng. Sci.*, 31, pp. 1257-1271.

Thibault, J., Lanouette, R., Fonteix., C. and Kiss, L. N. (2002). Multicriteria optimization of a high-yield pulping process, *Can. J. Chem.* Eng., 80, pp. 897-902.

Thibault, J., Taylor, D., Yanofsky, C., Lanouette, R., Fonteix., C. and Zaras, K. (2003). Multicriteria optimization of a high yield pulping process with rough sets, *Chem.Eng. Sci.*, 58, pp. 203-213.

Wang, F. S. and Sheu, J. W. (2000). Multi-objective parameter estimation problems of fermentation processes using a high ethanol tolerance yeast, *Chem .Eng. Sci.*, 55, pp. 3685-3695.

Wen, Y. and Shonnard, D. R (2003). Environmental and economic assessments of heat exchanger networks for optimum minimum approach temperature, *Comput. Chem. Eng.*, 27, pp. 1577-1590.

Wongso, F., Hidajat, K. and Ray, A. K. (2004). Optimal operating mode for enantioseparation of SB-553261 racemate based on simulated moving bed technology, *Biotechnol. Bioeng.*, 87, pp. 714-722.

Yee, A. K. Y., Ray, A. K. and Rangaiah, G. P. (2003). Multi-objective optimization of an industrial styrene reactor, *Comput. Chem. Eng.*, 27, pp. 111-/130.

Yu, W., Hidajat, K. and Ray, A. K. (2003a). Application of multi-objective optimization in the design and operation of reactive SMB and its experimental verification, *Ind. Eng. Chem. Res.*, 42, pp. 6823-6831.

Yu, W., Hidajat, K. and Ray, A. K. (2003b). Optimal operation of reactive simulated moving bed and Varicol systems, *J. Chem. Technol. Biot.*, 78, pp. 287-293.

Yu, W., Hariprasad, J. S., Zhang, Z., Hidajat, K. and Ray, A. K. (2004). Application of multi-objective optimization in the design of SMB in chemical process industry, *J. Chin. Inst. Chem. Engrs.*, 35, pp. 1-8.

Yu, W., Hidajat, K. and Ray, A. K. (2005). Optimization of reactive simulated moving bed and Varicol systems for hydrolysis of methyl acetate, *Chem. Eng. J.*, 112, pp. 57-72.

Zhang, Z., Hidajat, K. and Ray, A. K. (2002a). Multi-objective optimization of simulated counter current moving bed chromatographic reactor (SCMCR) for MTBE synthesis, *Ind. Eng. Chem. Res.*, 41, pp. 3213-3232.

Zhang, Z., Hidajat, K., Ray, A. K. (2002b). Multi-objective optimization of SMB and Varicol process for chiral separation, *AIChE J.*, 48, pp. 2800-2816.

Zhang, Z., Mazzotti, M. and Morbidelli, M. (2003). Multi-objective optimization of simulated moving bed and Varicol processes using a genetic algorithm, *J. Chromatogr. A.*,989, pp. 95-108.

Zhang, Y., Hidajat, K., Ray, A. K. (2004). Optimal design and operation of SMB bioreactor: production of high fructose syrup by isomerization of glucose, *Biochem. Eng. J.*, 21, pp. 111–121.

Zhou, F., Gupta, S. K. and Ray, A. K. (2000). Multi-objective optimization of the continuous casting process for poly (methyl methacrylate) using adapted genetic algorithm, *J. Appl. Polym. Sci.*, 78, pp. 1439-1458.

This page intentionally left blank

Chapter 3

Multi-Objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering

Antonio López Jaimes and Carlos A. Coello Coello*
CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07360, MEXICO

*E-mail: ccoello@cs.cinvestav.mx

Abstract

In this chapter, we provide a general overview of evolutionary multi-objective optimization, with particular emphasis on algorithms in current use. Several applications of these algorithms in chemical engineering are also discussed and analyzed. We also provide some additional information about public-domain resources available for those interested in pursuing research in this area. In the final part of the chapter, some potential areas for future research are briefly described.

Keywords: Evolutionary Multi-objective Optimization, Chemical Engineering, Metaheuristics, Evolutionary Algorithms.

3.1 Introduction

The solution of problems having two or more (normally conflicting) objectives has become very common in the last few years in a wide variety of disciplines. Such problems are called “multi-objective”, and

can be solved using either mathematical programming techniques (Miettinen, 1999) or using metaheuristics (Coello Coello *et al*., 2002). In either case the concept of Pareto optimality is normally adopted. When using this concept, we aim to obtain the best possible trade-offs among all the objectives.

Of the many metaheuristics available, evolutionary algorithms (EAs) have become very popular because of their ease of implementation and high effectiveness. EAs are based on an emulation of the natural selection mechanism (Goldberg, 1989). EAs are particularly suitable for solving multi-objective problems because of their ability to handle a set of solutions in a simultaneous manner, and their capability to deal with problems of different types, without requiring any specific problem-domain information (e.g., derivatives) (Deb, 2001).

The first multi-objective evolutionary algorithms (MOEAs) were introduced in the 1980s (Schaffer, 1985), but they became popular only in the mid 1990s. Nowadays, the use of MOEAs in all disciplines has become widespread (see for example (Coello Coello and Lamont, 2004)), and chemical engineering is, by no means, an exception.

This chapter provides a short introduction to MOEAs, presented from a historical perspective. It also reviews some of the most representative work regarding their use in chemical engineering applications. Finally, it provides a short description of some of the main Internet resources currently available for those interested in pursuing research in this area.

3.2 Basic Concepts

We are interested in the solution of MOO problems (MOOPs) of the form:

$$\text{minimize } \left[f_1(\vec{x}), f_2(\vec{x}), ..., f_k(\vec{x}) \right] \tag{3.1}$$

subject to the *m* inequality constraints:

$$g_i(\vec{x}) \leq 0 \qquad i = 1,2,\ldots,m \tag{3.2}$$

and the *p* equality constraints:

$$h_i(\vec{x}) \leq 0 \quad i = 1, 2, ..., p \tag{3.3}$$

where k is the number of objective functions, $f_i : \Re^n \rightarrow \Re$. We call $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set, $\mathcal{F}$, of all vectors which satisfy Eqs. (3.2) and (3.3) the particular set of values, $x_1^*, x_2^*, ..., x_n^*$, which yield the optimum values of all the objective functions.

3.2.1 *Pareto Optimality*

The most common notion of optimality adopted in multi-objective optimization is the so-called *Pareto optimality* (Pareto, 1896).

We say that a vector of decision variables, $\vec{x}^* \in \mathcal{F}$, is *Pareto optimal* if there does not exist another $\vec{x} \in \mathcal{F}$ such that $f_i(\vec{x}) \leq f(\vec{x}^*)$ for all $i = 1, ..., k$ and $f_j(\vec{x}) < f(\vec{x}^*)$ for at least one j.

In words, this definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables, $\vec{x} \in \mathcal{F}$, which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not just a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors, $\vec{x}^*$, corresponding to the solutions included in the Pareto optimal set are called *nondominated*. The image of the Pareto optimal set under the objective functions is called the *Pareto front*.

3.3 The Early Days

Apparently, Rosenberg's PhD thesis (Rosenberg, 1967) contains the first reference regarding the possible use of an evolutionary algorithm in an MOOP. Rosenberg suggests the use of multiple *properties* (nearness to some specified chemical composition) in his simulation of the genetics and chemistry of a population of single-celled organisms. This is then, an MOOP. However, Rosenberg's actual implementation contained a single property and no actual MOEA is developed in his thesis.

Despite the existence of an early paper by Ito *et al.* (1983), the first actual implementation of an MOEA is normally attributed to David Schaffer, who developed the *Vector Evaluated Genetic Algorithm* (VEGA) in the mid 1980s (Schaffer, 1985).

During the period from the mid 1980s up to the first half of the 1990s, a few other MOEAs were developed. Most of these approaches had a clear influence of the mathematical programming techniques developed within the operations research community and their implementations were straightforward, since they required very few (and simple) changes in the original structure of their underlying EAs.

In his famous book on genetic algorithms, Goldberg (1989) analyzes VEGA and indicates its main limitations. Goldberg also proposes a ranking scheme based on Pareto optimality. Such a mechanism, which was called *Pareto ranking*, would soon become standard within modern MOEAs. The basic idea of Pareto ranking is to find the set of individuals in the population that are Pareto nondominated with respect to the rest of the population. These individuals are then assigned the highest rank and eliminated from further contention. Another set of individuals which are nondominated with respect to the remainder of the population is then determined and these individuals are assigned the next highest rank. This process continues until the population is suitably ranked. Goldberg also suggested the use of some kind of diversification technique to keep the EA from converging to a single Pareto optimal solution. A niching mechanism such as fitness sharing (Goldberg and Richardson, 1987) was suggested for this purpose. Three major MOEAs would soon be developed based on these ideas. Each of them is briefly described next.

Fonseca and Fleming (1993) proposed the **Multi-Objective Genetic Algorithm** (MOGA), which soon became a very popular MOEA because of its effectiveness and ease of use. In MOGA, the rank of an individual corresponds to the number of chromosomes in the current population by which it is dominated. All nondominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region to which they belong. An interesting aspect of MOGA is that the ranking of the entire population is done in one pass, instead of having to reclassify the same individuals several times (as suggested by Goldberg (1989)).

Srinivas and Deb (1994) proposed the **Nondominated Sorting Genetic Algorithm** (NSGA) which is based on several layers of classifications of the individuals as suggested by Goldberg (1989). Before selection is performed, the population is ranked on the basis of nondomination: all nondominated individuals are classified into one

category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, fitness sharing is applied to these classified individuals using their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified. Stochastic remainder proportional selection is adopted for this technique. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population.

Horn *et al.* (1994) proposed the **Niched-Pareto Genetic Algorithm** (NPGA), which uses a tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is the following: two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). There are only two possible outcomes: (1) one of them is dominated (by the individuals randomly chosen from the population) and the other is not; in this case, the nondominated individual wins; (2) the second possible outcome is that the two competitors are either dominated or nondominated (i.e., there is a tie); in that case, the result of the tournament is decided through fitness sharing (Goldberg and Richardson, 1987). Since NPGA does not rank the entire population, but only a sample of it, it is more efficient (algorithmically) than MOGA and NSGA. The few comparative studies among these three MOEAs (MOGA, NPGA, and NSGA) performed during the mid and late 1990s, indicated that MOGA was the most effective and efficient approach, followed by NPGA and NSGA (in a distant third place) (Van Veldhuizen, 1999). MOGA was also the most popular MOEA of its time, mainly within the automatic control community.

3.4 Modern MOEAs

During the mid 1990s, several researchers considered a notion of elitism in their MOEAs (Husbands, 1994). Elitism in a single-objective EA consists of retaining the best individual from the current generation, and passing it intact (i.e., without being affected by crossover or mutation) to the following generation. In MOO, elitism is not straightforward, since

all the Pareto optimal solutions are equally good and, in theory, all of them should be retained.

Elitism was not emphasized (or even considered) in the early MOEAs described in the previous section. It was only in the late 1990s that elitism, in the context of MOO, was taken seriously. This was due to two main factors: the first was the proof of convergence of an MOEA developed by Rudolph (1998) which requires elitism. The second was the publication of the **Strength Pareto Evolutionary Algorithm** (SPEA) (Zitzler and Thiele, 1999) in the *IEEE Transactions on Evolutionary Computation*, which became a landmark in the field. SPEA was conceived as a way of integrating different MOEAs. It incorporates elitism through the use of an archive containing nondominated solutions found previously (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set. For each individual in this external set, a *strength* value is computed. This strength is similar to the ranking value of MOGA (Fonseca and Fleming, 1993), since it is proportional to the number of solutions that a certain individual dominates. In SPEA, the fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. The fitness assignment process of SPEA considers both the closeness to the true Pareto front as well as the distribution of solutions at the same time. Thus, instead of using niches based on distance, Pareto dominance is used to ensure that the solutions are properly distributed along the Pareto front. Although this approach does not require a niche radius, its effectiveness relies on the size of the external nondominated set. In fact, since the external nondominated set participates in the selection process of SPEA, if its size grows too large it might reduce the selection pressure, thus slowing down the search. Because of this, the authors decided to adopt a clustering technique that prunes the contents of the external nondominated set so that its size remains below a certain threshold.

After the publication of the SPEA paper, most researchers in the field started to incorporate external populations in their MOEAs as their elitist mechanism. In 2001, a revised version of SPEA (called **SPEA2**) was introduced. SPEA2 has three main differences with respect to its predecessor (Zitzler *et al.*, 2001): (1) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual, the

number of individuals that dominate it and the number of individuals by which it is dominated; (2) it uses a nearest neighbor density estimation technique which guides the search more efficiently, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions.

The **Pareto Archived Evolution Strategy** (PAES) is another major MOEA that was introduced at about the same time as SPEA (Knowles and Corne, 2000). PAES consists of a (1 + 1) evolution strategy (i.e., a single parent that generates a single offspring) in combination with a historical archive that records the nondominated solutions previously found. This archive is used as a reference set against which each mutated individual is compared. Such a historical archive is the elitist mechanism adopted in PAES. However, an interesting aspect of this algorithm is the procedure used to maintain diversity which consists of a crowding procedure that divides the objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its "coordinates" or "geographical location"). A map of such a grid is maintained, indicating the number of solutions that reside at each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space).

The **Nondominated Sorting Genetic Algorithm II** (NSGA-II) was introduced as an upgrade of NSGA (Srinivas and Deb, 1994), although it is easier to identify their differences than their similarities (Deb *et al.*, 2002). In NSGA-II, for each solution one has to determine how many solutions dominate it and the set of solutions which it dominates. NSGA-II estimates the density of solutions surrounding a particular solution in the population by computing the average distance of two points on either side of this solution along each of the objectives of the problem. This value is the so-called *crowding distance*. During selection, NSGA-II uses a crowded-comparison operator which takes into consideration both the nondomination rank of an individual in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). NSGA-II does not implement an elitist mechanism based on an external archive. Instead, the elitist mechanism of NSGA-II consists of combining the best

parents with the best offsprings obtained. Due to its clever mechanisms, NSGA-II is much more efficient (computationally speaking) than its predecessor, and its performance is so good that it has gained a lot of popularity in the last few years, becoming a benchmark against which other MOEAs are often compared. Many other MOEAs exist (see for example (Coello Coello *et al.*, 2002)), but they will not be discussed due to obvious space limitations. In any case, the MOEAs previously discussed are among the most popular in the current literature.

3.5 MOEAs in Chemical Engineering

A wide variety of techniques have been used to solve MOOPs in chemical engineering, including mathematical programming techniques (e.g., goal programming and the $\varepsilon-$constraint method) and MOEAs. This chapter only focuses on MOEAs, but readers interested in the first type of methods should refer to Bhaskar *et al.* (2000) for a review. It is worth noting, however, that since the late 1990s, MOEAs seem to be the preferred choice of practitioners to tackle MOO chemical engineering applications.

After reviewing the relevant literature, we found two types of papers: (1) those focusing on novel MOEAs or MOEA components, and (2) those focusing on novel applications using an existing MOEA. Section 3.6 briefly describes the most significant MOEAs that originated in the chemical engineering literature. For each algorithm, we mention some of their known applications and their advantages and disadvantages. Section 3.7, on the other hand, presents a selection of some representative MOO applications in chemical engineering that make use of well-known MOEAs. This selection is not meant to be exhaustive but attempts to delineate current research trends in the area.

3.6 MOEAs Originated in Chemical Engineering

As indicated before, this section is devoted to review works whose main goal is to propose a new MOEA (or an important component of it). Among these novel contributions we can find, for instance, an evolutionary operator, a constraint-handling technique and a proposal to

extend a single objective technique in order to deal with multiple objectives. It is important to emphasize that all the MOEAs discussed in this section originated in the chemical engineering community and have been mainly used to solve chemical engineering problems, although most of them can be applied in other domains.

3.6.1 *Neighborhood and Archived Genetic Algorithm*

Mu *et al.* (2003) proposed the Neighborhood and Archived Genetic Algorithm (NAGA), whose main goals are to provide (*i*) a new method to check for nondominance and (*ii*) a new technique to keep diversity in the Pareto front produced by the algorithm. In order to fulfill these goals, NAGA carries out neighborhood comparisons. The procedure to check for nondominance in the current population is divided into two stages. First, each new solution is locally compared to its neighbors. If the solution is locally dominated, then it is discarded since it will be globally dominated as well. On the other hand, if it is locally nondominated, then the solution is retained for the second stage. At this stage, only the locally nondominated solutions are compared with the current approximation set stored in a historical archive, using Pareto dominance again. After checking for nondominance, the new nondominated solutions are compared again using a crowding neighborhood process aimed to keep diversity. The implementation described by Mu *et al.* (2003) only considers one neighbor for each solution, **x**, namely the point resulting from a small perturbation in only one variable of **x**.

Regarding the crowding neighborhood process, if a new nondominated solution, **x**, is in the neighborhood of some solution, $\mathbf{x}^P$, in the archive, i.e., if $x_i \in [x_i^{(j)} - \varepsilon, x_i^{(j)} + \varepsilon]$, for each variable, *i*, and each archive solution, *j*, then the solution is discarded; otherwise, it is added to the archive. The parameter, ε, is defined by $\varepsilon = d \times (x_i^U - x_i^L)$, where *d* is a user-defined parameter, and x_i^U and x_i^L are the upper and lower bounds of the ith variable, respectively. Although, on average, the time required to identify the nondominated solutions is reduced in comparison with the standard Pareto ranking approach, the neighborhood comparisons introduce an extra evaluation of the objective functions for each individual. That is to say, the number of evaluations per generation is doubled with respect to the standard Pareto ranking approach. This is an

important drawback of this approach, particularly in chemical engineering applications where the time required to evaluate the objective functions is usually high. This is, without doubt, an important issue that must be taken into account before deciding to adopt NAGA in an application.

Applications

NAGA was used by Mu *et al.* (2004) to optimize the operation of a paraxylene oxidation process to give terephthalic acid. They consider two objectives: minimization of the concentration of the undesirable 4-carboxy-benzaldehyde (4-CBA) in the product stream, and maximization of the feed flow rate of the paraxylene. They consider four optimization problems using a different number of decision variables (1, 2, 4 and 6 variables). The problem has two constraints. The plot of the Pareto front obtained presented a convex and continuous curve.

Recently, Hou *et al.* (2007) used NAGA to maximize the aromatic yield and minimize the yield of heavy aromatics in a continuous industrial naphtha catalytic reforming process (that aims to obtain aromatic products).

3.6.2 *Criterion Selection MOEAs*

Dedieu *et al.* (2003) proposed an algorithm that can be considered as a criterion selection technique (Coello Coello *et al.*, 2002). That is to say, an algorithm where the solutions are selected based on separate objective performance. The main idea is to optimize separately each objective using a single objective genetic algorithm (SOGA). At the end of the single optimizations, the populations are merged to obtain the nondominated individuals. The authors proposed a variant where all populations generated through all generations of the SOGAs are merged. It is interesting to note that the proposed optimization algorithm was coupled with a discrete event simulator (DES), which was used to evaluate the different objectives and the technical feasibility of the proposed solutions. A detailed description of the DES used can be found in Bernal-Haro *et al.* (2002). Since in this case, the objective functions are not defined explicitly, this makes this kind of application an excellent

candidate to be solved by an evolutionary algorithm which, in contrast to gradient-based techniques, only needs objective function evaluations. As pointed out by Dietz *et al.* (2006), a drawback of this approach is that it is not able to produce a good distribution of solutions along the Pareto front, since it focuses on finding only a few solutions around the optima of each objective considered separately.

Dietz *et al.* (2006) proposed an approach similar to VEGA (Schaffer, 1985) in order to overcome the disadvantages of the previous proposal. In this new algorithm, k subpopulations of the entire population are ranked and selected according to a different objective (assuming k objective functions). After shuffling the subpopulations together, the crossover and mutation operators are applied in the usual way. This procedure is repeated until the stopping criterion is reached. At the end of the search, a procedure to check dominance is applied to obtain the Pareto set approximation. This algorithm produced a Pareto front having a better distribution of solutions than did its predecessor.

Applications

The first algorithm (i.e., the one proposed in Dedieu *et al.* (2003)) was applied to optimize the design of a multi-objective batch plant for manufacturing four products by using three types of equipment (available in different standard sizes). The problem considers two objectives: minimization of the investment cost of the plant and minimization of the number of different sizes of equipment for each unit operation. The second algorithm (i.e., the one proposed in Dietz *et al.* (2006)) was used to optimize the design of a multi-product batch plant for the production of proteins (human insulin, vaccine for Hepatitis B, chymosine and cryophilic protease). This combinatorial problem uses three objectives, the investment cost and two objectives concerning the environmental impact (total biomass quantity released and volume of polyethylene glycol used), and 44 decision variables: 16 continuous variables (operating conditions) and 28 integer variables (batch plant configurations). The cost objective involves investment costs for both equipment and storage vessels, whereas the evaluation of environmental impact combines three methodologies, namely, life cycle assessment, pollution balance principle and pollution vector methodology. The study

considers one problem using the three objectives mentioned above and two bi-objective problems (alternating the environmental objectives).

3.6.3 *The Jumping Gene Operator*

Kasat and Gupta (2003) proposed two new binary mutation operators whose main goal is to accelerate the convergence of the search in terms of the number of generations. These operators, called jumping genes (JG), are the following: (i) *replacement operator*, where a randomly selected *l*-length substring of the chromosome is replaced by a new string with length, *l*, generated at random; and (ii) *reversion operator*, where a randomly selected substring is reversed. The JG operator is applied to a fraction, P_{jump}, of the current population after the mutation phase. In order to evaluate the performance of the JG operators, the authors used NSGA-II (Deb *et al.*, 2002). According to their results, the reversion operator yields similar results than those obtained by NSGA-II without the proposed operator. However, the results of NSGA-II with the replacement operator (NSGA-II-JG) outperform those obtained by the standard NSGA-II. In the three test problems considered and based on visual inspections, NSGA-II-JG showed better convergence and distribution than NSGA-II.

Guria *et al.* (2005) proposed an adaptation of the JG operators aimed to solve *network* problems, e.g., the design of froth flotation circuits (discussed below). In network or circuit optimization problems, usually the optimal configuration includes values of the decision variables that lie exactly at their lower or upper bounds. The modified jumping gene operator (mJG) takes this peculiarity into account and does not select a substring at random, but a substring associated with one of the variables (*i.e.*, a gene). The gene selected is then replaced by a new gene that contains all zeros or all ones in accordance with a certain probability.

It is noteworthy that the JG operator has also been successfully incorporated into a Multi-objective Simulated Annealing technique (Sankararao and Gupta, 2007; see Chapter 4 in this book). The performance assessment of this algorithm was done on three well-known test (benchmark) problems commonly used in the evolutionary MOO field. This algorithm was then employed for the MOO of an industrial fluidized-bed catalytic cracking unit.

Applications

Kasat and Gupta (2003) used the JG operator for the MOO of an industrial fluidized-bed catalytic cracking unit. The objectives considered are the maximization of the yield of gasoline and the minimization of the coke formed on the catalyst during the cracking of heavy compounds. The decision variables included the feed preheat temperature, the air preheat temperature, the catalyst flow rate, and the air flow rate. In order to evaluate the performance of the algorithm, the optimization problem was also solved using sequential quadratic programming (SQP) with the ε-constraint method. According to the results obtained using six different values of ε, SQP failed to converge to the correct solution.

The modified JG (mJG) operator was used by Guria *et al.* (2005) to optimize the design of froth flotation circuits for mineral processing. In particular, they optimized a circuit with two flotation cells and two species. This problem involves the maximization of the recovery (ratio of the flow rates of the solid in the concentrate stream to that in the feed stream) and maximization of the grade (the fraction of the valuable mineral in the concentrate stream). The problem comprises 16 decision variables, namely, 14 cell-linkage parameters and 2 mean residence times. The problem contains three constraints related to the streams and one constraint related to the total volume of the cells. Recently, Guria *et al.* (2006) applied the mJG operator to optimize circuits with four cells and also considered problems with three and four objectives. A three-objective problem (maximization of the overall recovery of the concentrate, maximization of the number of non-linking streams and minimization of the total cell volume) is then solved. All the problems constrain the grade of the product to lie at a fixed value. Finally, a complex and computationally-intensive four-objective optimization problem is solved.

3.6.4 *Multi-Objective Differential Evolution*

Differential evolution (DE) is a branch of evolutionary algorithms developed by Storn and Price (1997) for optimization problems over continuous domains. DE is characterized by representing the variables by real numbers and by its three-parents crossover. At the selection stage,

three parents are chosen and they generate a single offspring by adding the weighted difference vector between two parents to a third parent. The offspring is compared with a parent to determine who passes to the following generation. DE has been very successful in the solution of a variety of continuous single-objective optimization problems in which it has shown great robustness and a very fast convergence. Recently, there have been several proposals to extend DE to MOO (Robič and Filipič, 2005). This section is devoted to present one of these extensions which has been used mainly to solve chemical engineering applications. The Multi-Objective Differential Evolution algorithm (MODE) was proposed by Babu and Jehan (2003). Its general framework is very similar to that of the standard DE. The main differences are: (*i*) the F parameter is generated from a random generator between 0 and 1; (*ii*) only the nondominated solutions are retained for recombination; (*iii*) the generated offspring is placed into the population if it dominates the first selected parent; and (*iv*) the constraints are handled using a penalty function approach.

Applications

MODE was used by Babu *et al.* (2005) to optimize the operation of an adiabatic styrene reactor. This work concerns a comparative study between the performance of MODE and the results of NSGA reported in a previous paper (Yee *et al.*, 2003). This application is described in Section 3.7.3. For comparative purposes, this study adopts the same formulation used by Yee *et al.* (2003). That is to say, the objectives are productivity, selectivity and yield of styrene; the variables are ethyl benzene feed temperature, pressure, steam-over-reactant ratio and initial ethyl benzene flow rate. Two constraints are also considered. On the one hand, the results obtained by MODE agreed with those obtained by NSGA, in particular the behavior of the variables in the Pareto optimal set. On the other hand, based on visual inspections, it was revealed that, in some cases, the Pareto fronts obtained by MODE were better than those obtained by NSGA, while in other cases the Pareto fronts seemed nearly identical (no performance indicators were adopted in this case).

3.7 Some Applications Using Well-Known MOEAs

The aim of this section is to present a selection of MOO chemical engineering applications that were solved using a well-known MOEA (e.g., MOGA (Fonseca and Fleming, 1993) or NSGA-II (Deb *et al.* 2002)) with some small adaptations suitable to the given application. We also found that some authors developed their own approaches based on mechanisms of existing MOEAs (for example, the nondominated sorting mechanism of the NSGA-II (Deb *et al.* 2002)).

The applications treated in this section are divided into five groups:

TYPE I: Related to polymerization processes.
TYPE II: Involving catalytic reactors.
TYPE III: Related to catalytic processes.
TYPE IV: Biological and bioinformatics problems.
TYPE V: General applications.

As we will see later, more than one study was found to address the same application in some cases.

3.7.1 *TYPE I: Optimization of an Industrial Nylon 6 Semibatch Reactor*

Mitra *et al*. (1998) employed NSGA (Srinivas and Deb, 1994) to optimize the operation of an industrial nylon 6 semibatch reactor. The two objectives considered in this study were the minimization of the total reaction time and the concentration of the undesirable cyclic dimer in the polymer produced. The problem involves two equality constraints: one to ensure a desired degree of polymerization in the product and the other, to ensure a desired value of the monomer conversion. The former was handled using a penalty function approach whereas the latter was used as a stopping criterion for the integration of the model equations. The decision variables were the vapor release rate history from the semibatch reactor and the jacket fluid temperature. It is important to note that the former variable is a function of time. Therefore, to encode it properly as a sequence of variables, the continuous rate history was discretized into several equally-spaced time points, with the first of these selected randomly between the two (original) bounds, and the rest selected randomly over smaller bounds around the previous generated value (so as

to ensure feasibility and continuity of the decision variable). The study showed that the solutions using NSGA were superior to the solutions obtained using Pontryagin's minimum principle.

3.7.2 *TYPE I: Optimization of an Industrial Ethylene Reactor*

Tarafder *et al.* (2005a) applied NSGA-II (Deb *et al.*, 2002) to study an industrial ethylene reactor following an MOO approach. The authors selected a free-radical mechanism to model the reactor. Three objectives were considered in this study, namely ethane conversion, ethylene selectivity and the flow rate of ethylene. Four MOO problems were formulated using these objectives. The first bi-objective optimization problem included ethane conversion and ethylene selectivity. These objectives had a conflicting behavior. The flow rate, which is related to the conversion and the selectivity, was included in two additional bi-objective problems: flow rate-conversion and flow rate-selectivity. Finally, a three-objective problem was formulated including all three objectives. The problem involved nine decision variables (seven continuous and two discrete). In order to verify the quality of the Pareto front obtained, an ε-constraint method was applied to generate some solutions at the center and at the extremes of the Pareto front. The results showed that these solutions lie on the Pareto front obtained by NSGA-II. It was observed that the Pareto optimal solutions were better than the industrial operating point in several cases.

3.7.3 *TYPE II: Optimization of an Industrial Styrene Reactor*

Yee *et al.* (2003) make use of the original NSGA (Srinivas and Deb, 1994) to optimize both adiabatic and steam-injected styrene reactors. A pseudo-homogeneous model was used to describe the reactor. This study maximizes three objectives: the amount of styrene produced, the selectivity of styrene and the yield of styrene. Two- and three-objective optimization problems are studied using combinations from these objectives. The decision variables for the adiabatic configuration are the feed temperature of ethyl benzene, inlet pressure, molar ratio of steam to ethyl benzene and the feed flow rate of ethyl benzene. The problem considers three constraints related to temperatures which are handled

using penalty functions. According to the plot of the Pareto front obtained, in two of the bi-objective cases the solutions obtained by NSGA are better than the industrial operating point. For the rest of the cases, the industrial operating point seems to lie on the Pareto front. Concerning the adiabatic configuration, one of the most interesting findings of this study is that in all the bi-objective problems only one variable changes while the others remain nearly constant. Additionally, the results confirm that steam injection is better than adiabatic operation.

In a later paper, Tarafder *et al.* (2005b) carried out the optimization of the entire styrene manufacturing process, which besides the styrene reactor, includes the heat-exchangers and the separations units. Instead of adopting the original NSGA, in this case the more recent NSGA-II is used (Deb *et al.*, 2002). In this study, three objectives are considered: the maximization of styrene production, maximization of the selectivity of styrene and the minimization of the total heat duty. The last objective reduces the emission of gases such as CO_x, SO_x and NO_x to the environment. Apart from the four variables included in the previous work, four more variables are added. Although three different reactor designs are studied (single bed, double bed and steam injection), only the double bed reactor was considered for the three-objective optimization. According to the authors, the constraint domination criterion used in NSGA-II gives better results than obtained by the penalty function approach.

3.7.4 *TYPE II: Optimization of an Industrial Hydrocracking Unit*

Bhutani *et al.* (2006) carried out the optimization of an industrial hydrocracking unit using NSGA-II (Deb *et al.*, 2002). Hydrocracking is a catalytic cracking process for the conversion of feedstock into more valuable lower boiling products. The optimization of a hydrocracking unit involves many objectives and variables. In this study, the authors considered three optimization problems depending, mainly, on the objectives chosen. The first case comprises the maximization of kerosene and the minimization of the flow rate of the make-up hydrogen. The decision variables are the flow rate of the feed, mass flow rate of the recycle gas and its temperature, recycle oil temperature, recycle oil mass fraction and the flow rates of the hydrogen input stream used for

quenching the catalyst beds. The maximum allowable inlet and exit temperatures for the hydrocracking are taken as constraints. The second case considers the maximization of heavy diesel and minimization of makeup hydrogen. The third case involves the maximization of high-value end products and minimization of low-value end products. The hydrocracking unit was modeled using a discrete lumped model approach, where the individual components in the reaction mixture are divided into discrete pseudo-compounds (lumps). Interestingly, the tuning of the model parameters was carried out employing genetic algorithm. The Pareto optimal set obtained by NSGA-II shows the conflict between the objectives in the three cases studied. Also, the results show that the current industrial operating point is inferior to the Pareto solutions.

3.7.5 *TYPE III: Optimization of Semi-Batch Reactive Crystallization Process*

Reactive crystallization is a production step for a wide range of chemical and pharmaceutical industries to produce solid particles with desirable characteristics, such as large crystal size, narrow crystal-size distribution, high-yield and so on. The feed flow rate of the reactants is a key control variable to improve the quality of the product crystals. Sarkar *et al.* (2007) carried out the optimization of a semi-batch reactive crystallization process using NSGA-II (Deb *et al.*, 2002). Since the quality of the product crystals is usually defined by the weight mean size of the crystal size distribution and the coefficient of variation, the authors selected these parameters as the two objectives of the problem. The amount of reactants added at certain intervals was used as the decision variable that defines the feed addition history. In order to define the feed history, the total time was divided into P equal-length intervals and each of these intervals has associated with it an amount of reactant added. Thus, the number of decision variables that define the feed history is P. The optimization problem presents three inequality constraints which are managed with the constraint domination approach in NSGA-II. The problem studied involves the precipitation of barium sulfate from an aqueous feed stream of barium chloride and sodium sulfate. The total batch time (180 s) was divided into ten intervals. In order to verify the closeness of the solutions obtained to the true Pareto front, some

solutions are generated executing repeatedly a weighted sum approach. Based on a visual comparison, the authors conclude that the solutions obtained using the weighted sum approach lie near the Pareto front obtained by NSGA-II.

3.7.6 *TYPE III: Optimization of Simulated Moving Bed Process*

Yu *et al.* (2003a) carried out the MOO of reactive simulated moving beds (RSMBs) for the synthesis of methyl acetate (MeOAc) both at the design and operation stages. They used NSGA (Srinivas and Deb, 1994). The model adopted was described by Yu *et al.* (2003b). The study considers three optimization problems. The first problem concerns the maximization of the purity and the yield of methyl acetate and involves two constraints. The decision variables are the switching time and the eluent flow rate. The results showed that the switching time plays a key role in determining the Pareto front. The second problem is to minimize the requirement of the absorbent and the flow rate of the eluent. The variables are the length of each column and the column configuration (i.e., the number of columns in each section). The last problem involves the maximization of the purity and the yield of methyl acetate and the minimization of the consumption of the eluent.

In a previous study, Zhang *et al.* (2002) used an MOO approach to compare the performance of the SMB process with that of a recently developed variant of SMB called the VARICOL process (Ludemann-Hombourger *et al.*, 2000). For comparison purposes, the authors use the same model for the SMB and the VARICOL processes developed by Ludemann-Hombourger *et al.* (2000). Besides the single-optimization problem used for verification purposes, the study considered two MOOPs. The first problem involves the maximization of the purity of both the raffinate and the extract streams (using the feed flow rate and the eluent consumption as parameters). The variables are then, the fluid flow rate in section one of the SMB system, the switching time and the column configuration. This case considers two constraints which are handled using penalty functions. The second case involves the maximization of the throughput and minimization of the eluent consumption. The decision variables are the same as in the last case: the feed flow rate and the eluent consumption. The authors concluded that the performance of a

VARICOL process is superior to that of an SMB process in terms of treating more feed using less eluent, or producing better product quality for a fixed productivity and solvent consumption. Recently, Yu *et al.* (2005) carried out a study that also compares the SMB and VARICOL processes, but for the hydrolysis of methyl acetate. In this study, the superiority of the VARICOL over the SMB process is observed.

3.7.7 *TYPE IV: Biological and Bioinformatics Problems*

Bioinformatics is the interdisciplinary field that encompasses the analysis of large volumes of biological data to generate useful information and knowledge. This knowledge can be used for applications such as drug discovery, disease diagnosis and prognosis, and determination of the relationship between species.

Some bioinformatics problems can be formulated as MOOPs, for instance the sequence alignment of DNA or protein sequences, protein structure prediction and design, and inference of gene regulatory networks, just to mention a few. The interested reader is referred to the review of Handl *et al.* (2007) that covers in detail more MOO applications in bioinformatics. The next few paragraphs describe some of the current applications in bioinformatics of interest to the chemical engineering community.

Sequence and structure alignment. Malard *et al.* (2004) formulate the *de novo* peptide identification as a constrained MOOP. The objectives considered in the study were the maximization of the similarity between portions of two peptides, and the maximization of the likelihood ratio between the null hypothesis and the alternative hypothesis.

Calonder *et al.* (2006) address the problem of identifying gene modules on the basis of different types of biological data such as gene expression and protein-protein interaction data. Module identification refers to the identification of groups of genes similar with respect to its function or regulation mechanism.

Protein and structure prediction. Chen *et al.* (2005) proposed a method to solve the structure alignment problem for homologous proteins. This problem can be formulated as a MOOP where the objectives are maximize the number of aligned atoms and minimize their distance.

Shin *et al.* (2005) use the controlled NSGA-II (Deb and Goel, 2001) to generate a set of quality DNA sequences. In this study, the quality of a sequence was achieved by minimizing four objectives: the similarity between two sequences in the set, the possible hybridization between sequences in a set, the continuous occurrence of the same base and the possible occurrence of the complementary substring in a sequence.

Gene regulatory networks. Spieth *et al.* (2005) address the problem of finding gene regulatory networks using an EA combined with a local search method. The global optimizer is a genetic algorithm whereas an evolutionary strategy plays the role of the local optimizer.

Recently, Keedwell and Narayanan (2005) combined a genetic algorithm with a neural network to elucidate gene regulatory networks. The genetic algorithm has the goal of evolving a population of genes, while the neural network is used to evaluate how well the expression of the set of genes affects the expression values of other genes.

3.7.8 *TYPE V: Optimization of a Waste Incineration Plant*

Anderson *et al.* (2005) applied MOGA (Fonseca and Fleming, 1993) to optimize the operation of a waste incineration plant. In order to guarantee profitability and taking into account environmental concerns, the objectives of this problem comprise, respectively, the maximization of waste feed rate and minimization of unburnt carbon in the combustion ash. The decision variables considered are the waste feed rate and the residence time of the waste on the burning bed. A constraint was used on the temperature of the chamber. The variant of MOGA used in this study allows the user to define goal values and priorities for the objectives in order to articulate preferences (Fonseca and Fleming, 1998). This modification of MOGA also incorporates a methodology to handle constraints-related information. MOGA performed well in this application, since it converged (as expected) to values of high residence times over a range of values of the waste feed rate.

3.7.9 *TYPE V: Chemical Process Systems Modelling*

Of the three evolutionary techniques, genetic algorithms are by far the most commonly applied in chemical engineering. However, genetic

programming (GP) embodies a powerful technique that has a lot of potential within chemical engineering (e.g., for modelling chemical process systems). In this direction, Hinchliffe *et al.* (1998) proposed a multi-objective genetic programming (MOGP) algorithm to model steady-state chemical process. This technique is based on MOGA (Fonseca and Fleming, 1993) which uses fitness sharing to keep diversity, and the concept of preferability based on a given goal vector. This work includes two case studies: an inferential estimator for bottom product composition in a vacuum distillation column and a model for the degree of starch gelatinization in an industrial cooking extruder. The four objectives considered for both case studies include: (i) root mean square error on the training data set; (ii) residual variance, which aggregates a credit to models that produce accurate approximations apart from a constant offset; (iii) correlation between individuals and the process output; and (iv) model string length, which helps to avoid complex models leading to overfitting. The study compares the performance of the MOGP with that of a single objective genetic programming (SOGP) algorithm proposed in a previous work (Hinchliffe *et al.*, 1996). The comparison was based on the RMS error on the validation data set and on the lengths of the models with lowest RMS error. The comparison involves the distribution of model prediction errors resulting from multiple runs. In the case of the distillation column no significant difference between the distributions of SOGP and MOGP was observed neither in RMS error nor in string length. This is due, according to the authors, to the fact that modelling of the column data is not a particularly difficult problem from a GP point of view. With regard to the cooking extruder, MOGP obtained the best minimum RMS error and the best mean RMS value. However the distribution analysis did not reveal a significant difference between the distributions.

In a more recent study, Hinchliffe and Willis (2003) model dynamic systems using genetic programming. The new approach is evaluated using two case studies, a test system with a time delay and an industrial cooking extruder. The objectives minimized are the root mean square error and the correlation and autocorrelations between residuals. The residuals of a model represent the difference between the predicted and actual values of the process output. In this work, two MOGPs are compared, one based on Pareto ranking but without preferences, and

another one, also based on Pareto ranking but with goal and priority information. From the results obtained in both case studies, the authors conclude that MOGP with preference information was able to evolve a greater number of acceptable solutions than the algorithm that used conventional Pareto ranking.

3.8 Critical Remarks

Most of the studies reviewed in Section 3.6 rely on visual inspections to compare the generated Pareto fronts from different algorithms in order to show which algorithm performs better. However, graphical plots have some drawbacks for comparative purposes. One of the most serious drawbacks is that given the stochastic nature of MOEAs, a unique graphical plot is not enough to state that one algorithm outperforms another since in each run a different Pareto front may be generated. Furthermore, even if we can state that one algorithm is better than another using only visual inspections, it is better to be able to determine, in a quantitative way, how much better it is. MOEA researchers have developed a variety of performance measures for this sake (see (Coello Coello *et al.*, 2002; Zitzler *et al.*, 2003) for further information) and a more extended use of them is expected to occur in future chemical engineering applications of MOEAs.

It is also worth indicating that the chemical engineering applications reviewed in this chapter tend to select a MOEA from a very reduced set (MOGA (Fonseca and Fleming, 1993), NSGA (Srinivas and Deb, 1994) and NSGA-II (Deb *et al.*, 2002)). However there are many other MOEAs that may be worth exploring: for example SPEA2 (Zitzler *et al.*, 2001), PAES (Knowles and Corne, 2000) and ε-MOEA (Deb *et al.*, 2005), which have all been successfully applied in other domains.

Finally, it is important to emphasize that comparative evaluations of MOEAs for any application should be done based on results obtained by the algorithms using the same process model, rather than based on previously reported results (particularly when they are originated by different researchers). This is because of the potential (and unknown) differences in the process models used by different researchers and their effect on optimization results.

3.9 Additional Resources

Launched in 1998, the EMOO repository[1] (Coello Coello, 2006) is one of the main resources for those interested in pursuing research in evolutionary MOO. The EMOO repository contains:

- Public-domain software.
- Test functions (either academic or real-world problems).
- URLs of events of interest for the EMOO community.
- Contact information of those who want to be added to the database of EMOO researchers (name, affiliation, postal address, email, web page and photo, if available).

As of August 2007, the EMOO repository contains:

- Over 2900 bibliographic references, which include 175 Ph.D. theses, 24 Masters theses, more than 790 journal papers and more than 1560 conference papers.
- Contact information of 66 EMOO researchers.
- Public domain implementations of several MOEAs.
- Links to PISA (Bleuler *et al.*, 2003) and ParadisEO-MOEO (Liefooghe *et al.*, 2007), which are modern platforms that facilitate the use and development of MOEAs.

3.10 Future Research

There are several potential areas of future research regarding the use of MOEAs in chemical engineering applications. Some of them are the following:

Use of relaxed forms of dominance: Some researchers have proposed the use of relaxed forms of Pareto dominance as a way of regulating convergence of a MOEA. Laumanns *et al.* (2002) proposed the so-called ε-dominance. This mechanism acts as an archiving strategy to ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found. Several modern MOEAs have adopted the concept of ε-dominance (see for example

[1]The EMOO repository is located at: http://delta.cs.cinvestav.mx/~ccoello/EMOO.

(Deb *et al.*, 2005)), because of its several advantages. Its use within chemical engineering, however, remains to be explored.

Incorporation of user's preferences: Although many of the current MOEA-related work assumes that the user is interested in generating the entire Pareto front of a problem, in practice, normally only a small portion (or even a few solutions) is required. The incorporation of user's preferences is a problem that has been long studied by operation researchers (Figueira *et al.*, 2005). However, relatively little work has been done in this regard by MOEA researchers (Coello Coello, 2000; Branke and Deb, 2005). Nevertheless, this is a topic that certainly deserves attention from practitioners in chemical engineering.

3.11 Conclusions

This chapter has provided a brief introduction to MOEAs and their use in chemical engineering. Both algorithms and applications have been described and analyzed. From the review of the literature that was undertaken to write this chapter, it became evident that chemical engineering practitioners are already familiar with MOEAs. Thus, no attempt was made to raise their interest any further.

Thus, this chapter has attempted to provide a critical review of the current work done with MOEAs in chemical engineering, from a MOEA researchers' perspective. The intention, however, was not to minimize or disregard the important work already done. Instead, the aim was to bring practitioners closer to the MOEA community so that both can interact and mutually benefit. If some of the ideas presented in this chapter are incorporated by chemical engineering practitioners in the years to come, we will then know that the goals of this chapter have been fulfilled.

Acknowledgements

The second author acknowledges support from CONACyT project no. 45683-Y.

References

Anderson, S., Kadirkamanathan, V., Chipperfield, A., Sharifi, V. and Swithenbank, J. (2005). Multi-objective optimization of operational variables in a waste incineration plant, *Computers & Chemical Engineering* **29**(5), pp. 1121-1130.

Babu, B., Chakole, P. and Mubeen, J. (2005). Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor, *Chemical Engineering Science* **60**(17), pp. 4822-4837.

Babu, B. and Jehan, M. M. L. (2003). Differential evolution for multi-objective optimization, in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, Vol.4 (IEEE Press, Canberra, Australia), pp. 2696-2703.

Bernal-Haro, L., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S. (2002). Multiobjective batch plant design: a two-stage methodology. 1. development of a design-oriented discrete-event simulation model, *Industrial and Engineering Chemistry Research* **41**(23), pp. 5727-742.

Bhaskar, V., Gupta, S. K. and Ray, A. K. (2000). Applications of multiobjective optimization in chemical engineering, *Reviews in Chemical Engineering* **16**(1), pp. 1-54.

Bhutani, N., Ray, A. K. and Rangaiah, G. P. (2006). Modeling, simulation and multi-objective optimization of an industrial hydrocracking unit, *Industrial and Engineering Chemistry Research* **45**, pp. 1354-1372.

Bleuler, S., Laumanns, M., Thiele, L. and Zitzler, E. (2003). PISA—A platform and programming language independent interface for search algorithms, in C. M. Fonseca *et al.* (eds.), *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003* (Springer. LNCS Vol. 2632, Faro, Portugal), pp. 494-508.

Branke, J. and Deb, K. (2005). Integrating user preferences into evolutionary multi-objective optimization, in Y. Jin (ed.), *Knowledge Incorporation in Evolutionary Computation* (Springer, Berlin Heidelberg), pp. 461-477, ISBN 3-540-22902-7.

Calonder, M., Bleuler, S. and Zitzler, E. (2006). Module identification from heterogeneous biological data using multiobjective evolutionary algorithms, in *Parallel Problem Solving from Nature (PPSN IX)*. Springer LNCS Vol. 4193, pp. 573-582.

Chen, L., Wu, L., Wang, R., Wang, Y., Zhang, S. and Zhang, X. (2005). Comparison of protein structures by multi-objective optimization, *Genome Informatics* **16**(2), pp. 114-124.

Coello Coello, C. A. (2000). Handling preferences in evolutionary multiobjective optimization: a survey, in *2000 IEEE Congress on Evolutionary Computation*, Vol. 1 (IEEE Service Center, Piscataway, New Jersey), pp. 30-37.

Coello Coello, C. A. (2006). The EMOO repository: a resource for doing research in evolutionary multiobjective optimization, *IEEE Computational Intelligence Magazine* **1**, 1, pp. 37-45.

Coello Coello, C. A. and Lamont, G. B. (eds.) (2004). *Applications of Multi-Objective Evolutionary Algorithms* (World Scientific, Singapore), ISBN 981-256-106-4.

Coello Coello, C. A., Van Veldhuizen, D. A. and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems* (Kluwer Academic Publishers, New York), ISBN 0-3064-6762-3.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms* (John Wiley & Sons, Chichester, UK), ISBN 0-471-87339-X.

Deb, K. and Goel, T. (2001). Controlled elitist non-dominated sorting genetic algorithms for better convergence, in Eckart Zitzler *et al*. (Eds), *First International Conference on Evolutionary Multi-Criterion Optimization*. Springer LNCS Vol. 1993 (Zürich, Switzerland), pp. 67-81.

Deb, K., Mohan, M. and Mishra, S. (2005). Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions, *Evolutionary Computation* **13**(4), pp. 501-525.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* **6**(2), pp. 182-197.

Dedieu, S., Pibouleau, L., Azzaro-Pantel, C. and Domenech, S. (2003). Design and retrofit of multiobjective batch plants via a multicriteria genetic algorithm, *Computers & Chemical Engineering* **27**(12), pp. 1723-1740.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S. (2006). Multiobjective optimization for multiproduct batch plant design under economic and environmental considerations, *Computers & Chemical Engineering* **30**(4), pp. 599-613.

Figueira, J., Greco, S. and Erghott, M. (eds.) (2005). *Multiple Criteria Decision Analysis. State of the Art Surveys* (Springer, USA).

Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in S. Forrest (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms,* (Morgan Kauffman Publishers, San Mateo, California), pp. 416-423.

Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* **28**(1), pp. 26-37.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley Publishing Company, Reading, Massachusetts).

Goldberg, D. E. and Richardson, J. (1987). Genetic algorithm with sharing for multimodal function optimization, in J. J. Grefenstette (ed.), *Proceedings of the Second International Conference on Genetic Algorithms* (Lawrence Erlbaum, Hillsdale, New Jersey), pp. 41-49.

Guria, C., Varma, M., Mehrotra, S. P. and Gupta, S. K. (2006). Simultaneous optimization of the performance of flotation circuits and their simplification using the jumping gene adaptations of genetic algorithm-II: More complex problems, *International Journal of Mineral Processing* **79**(3), pp. 149-166.

Guria, C., Verma, M., Mehrotra, S. and Gupta, S. (2005). Multi-objective optimal synthesis and design of froth flotation circuits for mineral processing, using the jumping gene adaptation of genetic algorithm, *Industrial & Engineering Chemistry Research*, **44**(8), pp. 2621-2633.

Handl, J., Kell, D. and Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology, *IEEE-ACM Transactions on Computational Biology and Bioinformatics*, **4**(2), April-June, pp. 279-292.

Hinchliffe, M., Hiden, H., McKay, B., Willis, M., Tham, M. and Barton, G. (1996). Modelling chemical process systems using a multi-gene genetic programming algorithm, in J. R. Koza (ed.), *Late Breaking Papers at the Genetic Programming 1996 Conference* (Stanford Bookstore, Stanford University, CA, USA), pp. 56-65.

Hinchliffe, M., Willis, M. and Tham, M. (1998). Chemical process systems modelling using multi-objective genetic programming, in J.R. Koza *et al.* (eds.), *Proceedings of the Third Annual Conference on Genetic Programming*, (Morgan Kaufmann Publishers, San Mateo, California), pp. 134-139.

Hinchliffe, M. P. and Willis, M. J. (2003). Dynamic systems modeling using genetic programming, *Computers & Chemical Engineering*, **27**(12), pp. 1841-1854.

Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization, in *Proceedings of the First IEEE Conference on Evolutionary Computation*, Vol. 1 (IEEE Service Center, Piscataway, New Jersey), pp. 82-87.

Hou, W.F., Su, H.Y., Mu, S.J. and Chu, J. (2007). Multiobjective optimization of the industrial naphtha catalytic reforming process, *Chinese Journal of Chemical Engineering* **15**(1), pp. 75-80.

Husbands, P. (1994). Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimisation, in T. C. Fogarty (ed.), *Evolutionary Computing. AIS Workshop. Selected Papers*, Springer, LNCS Vol. 865 (Leeds, UK), pp. 150-165.

Ito, K., Akagi, S. and Nishikawa, M. (1983). A Multiobjective optimization approach to a design problem of heat insulation for thermal distribution piping network systems, *Journal of Mechanisms, Transmissions and Automation in Design (Transactions of the ASME)* **105**, pp. 206-213.

Kasat, R. B. and Gupta, S. K. (2003). Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator, *Computers & Chemical Engineering* **27**(12), pp. 1785-1800.

Keedwell, E. and Narayanan, A. (2005). Discovering gene networks with a neural-genetic hybrid, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2**(3), pp. 231-242.

Knowles, J. D. and Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy, *Evolutionary Computation*, **8**(2), pp. 149-172.

Laumanns, M., Thiele, L., Deb, K. and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multi-objective optimization, *Evolutionary Computation*, **10**(3), pp. 263-282.

Liefooghe, A., Basseur, M., Jourdan, L. and Talbi, E.-G. (2007). ParadisEO-MOEO: A framework for evolutionary multi-objective optimization, in S. Obayashi *et al.* (eds.), *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007* (Springer. LNCS Vol. 4403, Matshushima, Japan), pp. 386-400.

Ludemann-Hombourger, O., Nicoud, R. M. and Bailly, M. (2000). The VARICOL process: a new multicolumn continuous chromatographic Process, *Separation Science and Technology*, **35**(12), pp. 1829-1862.

Malard, J., Heredia-Langner, A., Baxter, D., Jarman, K. and Cannon, W. (2004). Constrained de novo peptide identification via multi-objective optimization, *Online Proceedings of the Third IEEE International Workshop on High Performance Computational Biology (HiCOMB 2004), Santa Fe, New Mexico, USA*.

Miettinen, K. M. (1999). *Nonlinear Multiobjective Optimization* (Kluwer Academic Publishers, Boston, Massachusetts).

Mitra, K., Deb, K. and Gupta, S. K. (1998). Multiobjective dynamic optimization of an industrial nylon 6 semibatch reactor using genetic algorithm, *Journal of Applied Polymer Science*, **69**(1), pp. 69-87.

Mu, S.J., Su, H.Y., Wang, Y.X., and Chu, J. (2003). An efficient evolutionary multi-objective optimization algorithm, in *Proceedings of the 2003 Congress on Evolutionary Computation*, Vol. 2 (IEEE Press, Canberra, Australia), pp. 914-920.

Mu, S.J., Su, H.Y., Jia, T., Gu, Y. and Chu, J. (2004). Scalable multi-objective optimization of industrial purified terephthalic acid (PTA) oxidation process, *Computers & Chemical Engineering* **28**(11), pp. 2219-2231.

Pareto, V. (1896). *Cours D'Economie Politique*, Vol. I and II (F. Rouge, Lausanne).

Robič, T. and Filipič, B. (2005). DEMO: differential evolution for multiobjective optimization, in C. A. Coello Coello *et al.* (eds.), *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005* (Springer. LNCS Vol. 3410, Guanajuato, México), pp. 520-533.

Rosenberg, R. S. (1967). *Simulation of genetic populations with biochemical properties*, Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, USA.

Rudolph, G. (1998). On a multi-objective evolutionary algorithm and its convergence to the Pareto set, in *Proceedings of the 5th IEEE Conference on Evolutionary Computation* (IEEE Press, Piscataway, New Jersey), pp. 511-516.

Sankararao, B. and Gupta, S. K. (2007). Multi-objective optimization of an industrial fluized-bed catalytic cracking unit (FCCU) using two jumping gene adaptations of simulated annealing, *Computers & Chemical Engineering*, **31**, 11, pp. 1496-1515.

Sarkar, D., Rohani, S. and Jutan, A. (2007). Multiobjective optimization of semibatch reactive crystallization processes, *AIChE Journal,* **53**(7), pp. 1164-1177.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms, in *Proceedings of the First International Conference on Genetic Algorithms* (Lawrence Erlbaum, Hillsdale, New Jersey), pp. 93-100.

Shin, S., Lee, I., Kim, D. and Zhang, B. (2005). Multiobjective evolutionary optimization of DNA sequences for reliable DNA Computing, *IEEE Transactions on Evolutionary Computation* **9**(2), pp. 143-158.

Spieth, C., Streichert, F., Supper, J., Speer, N. and Zell, A. (2005). Feedback memetic algorithms for modeling gene regulatory networks, *Proceedings of Computational Intelligence in Bioinformatics and Computational Biology, 2005*, pp. 1-7.

Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation* **2**(3), pp. 221-248.

Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, **11**(4), pp. 341-359.

Tarafder, A., Lee, B., Ray, A. K. and Rangaiah, G. P. (2005a). Multi-objective optimization of an industrial ethylene reactor using a Non-dominated sorting genetic algorithm, *Industrial and Engineering Chemistry Research*, **44**, pp. 124-141.

Tarafder, A., Rangaiah, G. P. and Ray, A. K. (2005b). Multiobjective optimization of an industrial styrene monomer manufacturing process, *Chemical Engineering Science*, **60**(2), pp. 347-363.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Ph.D. thesis, Department of Electrical and Computer Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Yee, A. K., Ray, A. K. and Rangaiah, G. P. (2003). Multiobjective optimization of an industrial styrene reactor, *Computers & Chemical Engineering*, **27**(1), pp. 111-130.

Yu, W., Hidajat, K. and Ray, A. K. (2003a). Application of multiobjective optimization in the design and operation of reactive SMB and its experimental verification, *Industrial & Engineering Chemistry Research*, **42**(26), pp. 6823-6831.

Yu, W., Hidajat, K. and Ray, A. K. (2003b). Modeling, simulation, and experimental study of a simulated moving bed reactor for the synthesis of methyl acetate ester, *Industrial & Engineering Chemistry Research*, **42**(26), pp. 6743-6754.

Yu, W., Hidajat, K. and Ray, A. K. (2005). Optimization of reactive simulated moving bed and varicol systems for hydrolysis of methyl acetate, *Chemical Engineering Journal*, **112**, pp. 57-72.

Zhang, Z., Hidajat, K., Ray, A. K. and Morbidelli, M. (2002). multiobjective optimization of SMB and varicol process for chiral separation, *AIChE Journal,* **48**(12), pp. 2800-2816.

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm, in K.Giannakoglou *et al.* (eds.), *EUROGEN 2001*. (Athens, Greece), pp. 95-100.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation* **3**(4), pp. 257-271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review, *IEEE Transactions on Evolutionary Computation* **7**(2), pp. 117-132.

Chapter 4

Multi-Objective Genetic Algorithm and Simulated Annealing with the Jumping Gene Adaptations

Manojkumar Ramteke and Santosh K. Gupta*
Department of Chemical Engineering,
Indian Institute of Technology, Kanpur 208016, India

*skgupta@iitk.ac.in

Abstract

Two popular evolutionary techniques used for solving multi-objective optimization problems, namely, genetic algorithm and simulated annealing, are discussed. These techniques are inherently more robust than conventional optimization techniques. Incorporating a macro-macro mutation operator, namely, the jumping gene operator, inspired from biology, reduces the computational time required for convergence, significantly. It also helps to obtain the global optimal Pareto set where several Paretos exist. In this article, *detailed* descriptions of genetic algorithm and simulated annealing with the various jumping gene adaptations are presented and then three benchmark problems are solved using them.

Keywords: Genetic Algorithm, Simulated Annealing, Jumping Gene.

4.1 Introduction

As discussed in Chapter 1, most real-world engineering problems require the simultaneous optimization of *several* objectives (multi-objective optimization, MOO) that cannot be compared easily with one another, i.e., are non-commensurate. These cannot be combined into a single, meaningful scalar objective function. A simple, two-objective example, involving two decision ($n = 2$) variables, $\boldsymbol{x}$ ($\equiv [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$), is described by

$$\begin{aligned} &\text{Opt. } I_1(\boldsymbol{x}) \\ &\text{Opt. } I_2(\boldsymbol{x}) \\ &\text{Subject to bounds on } \boldsymbol{x}\text{: } \; x_{i,L} \le x_i \le x_i^U \; ; \; i = 1, 2 \end{aligned} \tag{4.1}$$

An example is the maximization of the desired product from a reactor, while simultaneously minimizing the production of an undesirable side product [a problem involving the minimization of I_i can be transformed into a problem involving the maximization of F_i, using: Min $I_i \rightarrow$ Max $\{F_i \equiv 1/(1 + I_i)\}$]. Often, the solutions are a set of several equally good (non-dominated) optimal solutions, called a Pareto front. These have been described in Chapter 1. The concept of non-dominance is central to the understanding of algorithms for MOO. Fig. 4.1 shows a typical Pareto set, corresponding to the *minimization* of both the objective functions, I_1 and I_2. It is clear that point A is better (superior) in terms of I_1, but worse (inferior) in terms of I_2, when compared to point B. Points A and B are referred to as non-dominated points since neither is superior to (dominates over) the other. Fig. 4.1, thus, represents a Pareto front. Point, C (which is not part of the optimal solution), for example, is inferior to point B with respect to both I_1 and I_2 (both the fitness functions for point C are higher than for point B). Point B, thus, dominates over point C.

Very popular and robust techniques like genetic algorithm (GA) and simulated annealing (SA) are used to solve such problems. The multi-objective forms of these techniques, e.g., NSGA-II (Deb *et al*., 2002) and MOSA (Suppapitnarm *et al*., 2000), are quite commonly used these days. These algorithms often require large amounts of computational (CPU) time. Any adaptation to speed up the solution procedure is, thus,

desirable. An attempt has been made along this direction by Simoes and Costa (1999a, 1999b), Kasat and Gupta (2003), Man *et al*. (2004), Chan *et al*. (2005a, 2005b), Guria *et al*. (2005a), Bhat *et al*. (2006), Bhat (2007), Ripon *et al*. (2007), and Agarwal and Gupta (2007a, 2007b), to improve NSGA-II using the concept of jumping genes [JG; or transposons, predicted by McKlintock (1987)] in biology. The basics of these techniques are first explained using their single-objective forms, e.g., simple GA (SGA) (Holland, 1975; Goldberg, 1989; Deb, 1995 and 2001; Coello Coello *et al*., 2002), and simple SA (SSA) (Metropolis *et al*., 1953; Kirkpatrick *et al*., 1983).

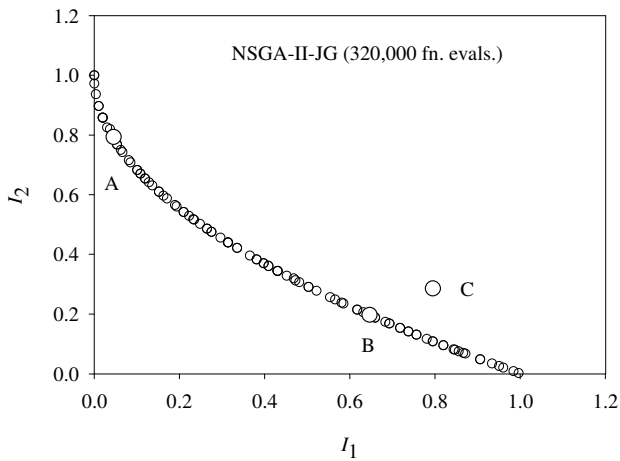


Fig. 4.1 The Pareto set obtained for the ZDT4 problem (Deb, 2001) using NSGA-II-JG. An additional point, C, is also indicated

4.2 Genetic Algorithm

4.2.1 *Simple GA (SGA) for Single-Objective Problems*

The single-objective optimization problem involving two decision variables that we have chosen for illustration is given by

$$\begin{aligned} &\text{Opt } I(\boldsymbol{x}) \\ &\text{Subject to bounds on } \boldsymbol{x}\text{: } \; x_{i,L} \le x_i \le x_i^U \; ; \; i = 1, 2 \end{aligned} \tag{4.2}$$

The feasible region satisfying the bounds is shown schematically in Fig. 4.2. We generate, using several random numbers, N_p solutions, each having n decision variables. *Each* decision variable in any solution is represented in terms of l_{string} binary numbers. The set of $n_{chr} \equiv l_{string} \times n$ binaries representing a solution is referred to as a 'chromosome' or 'string', while the binaries representing any decision variable are referred to as substrings.

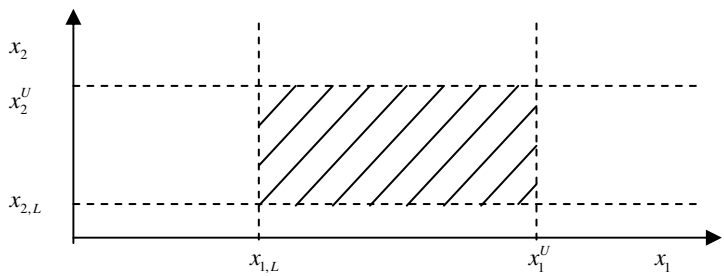


Fig. 4.2 Feasible region for a problem with two decision variables (Eq. 4.2)

An example of two chromosomes (strings) generated using a random number generating code (if the random number, R, is between 0 and 0.5, use the binary, 0, while if $0.5 \le R \le 1.0$, use the binary, 1) and with $l_{string} = 4$, $n = 2$, is:

	$S_3\ S_2\ S_1\ S_0$	$S_3\ S_2\ S_1\ S_0$
1st chromosome:	1 0 1 0	0 1 1 1
2nd chromosome:	1 1 0 1	0 1 0 1
	substring 1	substring 2

(4.3)

Here, S_0, S_1, S_2, and S_3 denote the binaries in any substring at the 0th, 1st, 2nd and 3rd positions, respectively. The domain, $[x_{i,L}, x_i^U]$, for each decision variable is now divided into $(2^{l_{string}} - 1)$ [= 15 in the present example] equal intervals and all the sixteen possible binary numbers assigned sequentially. Fig. 4.3 shows that the lower limit, $x_{i,L}$, for a decision variable is assigned to the 'all 0' substring, (0 0 0 0), while the upper limit, x_i^U, to the 'all 1' substring, (1 1 1 1). The other 14 combinations of the substring are sequentially assigned values in-between the bounds of x_i, as shown in Fig. 4.3.

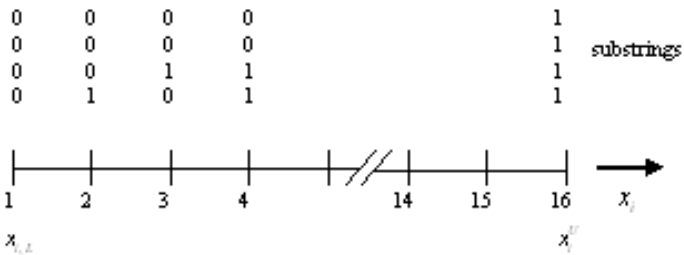


Fig. 4.3 Mapping of binary substrings

The binary substrings are mapped into *real* numbers using the following mapping rule:

$$x_i = x_{i,L} + \frac{x_i^U - x_{i,L}}{2^{l_{string}} - 1} \times \left(\sum_{i=0}^{l_{string}-1} 2^i S_i \right) \tag{4.4}$$

Clearly, the larger the value of l_{string}, the larger are the number of intervals in $[x_{i,L}, x_i^U]$, and the higher is the accuracy of search. In addition, such a mapping ensures that the constraints (bounds) in Eq. 4.2 are satisfied. The mapped real values of each of the (two, here) decision variables are used in a model to evaluate the value of the fitness function, $I(\boldsymbol{x}_j)$, for the jth chromosome.

At this stage, we have a set of N_p feasible solutions (parent chromosomes), each associated with a fitness value. We now need to generate improved chromosomes (daughters) using an appropriate methodology. This is done by mimicking natural genetics. The first step is referred to as reproduction. We make N_p copies of the parent chromosomes at a new location, called the 'gene pool', using proportionate representation based on how 'good' is a chromosome, i.e., the better the jth chromosome (in terms of I), the higher is its chance of getting copied. The probability, P_j, of selecting the jth string for copying is taken as:

$$P_j = \frac{I(\boldsymbol{x}_j)}{\sum_{i=1}^{N_p} I(\boldsymbol{x}_i)} \; ; j = 1, 2, \ldots, N_p \tag{4.5}$$

The actual methodology used for this is the roulette wheel selection. We divide the range of random number, R, into N_p zones: $0 \le R \le P_1$; $P_1 \le R \le P_1 + P_2$; . . . ; $\sum_{j=1}^{N_p-1} P_j \le R \le \sum_{j=1}^{N_p} P_j = 1$, and assign chromosomes, 1, 2, . . . , N_p, to these zones, respectively. For example, if we have five randomly generated chromosomes with the characteristics shown in Table 4.1, we can partition the range, [0, 1], of R and associate the appropriate chromosome to each zone. A random number, R', with $0 \le R' \le 1$, is now generated and the corresponding chromosome is copied (without deletion from the parent pool) into the 'gene pool'. For example, if R' is obtained as 0.45, string 3 is copied. This procedure is repeated N_p times. Clearly, chromosomes having higher fitness values will be selected more frequently in the gene pool. Due to the randomness associated with this selection procedure, there are chances that some 'poor' chromosomes also get copied (survive). This helps maintain diversity of the gene pool (two 'idiots' can produce a genius, etc.).

Table 4.1 Five chromosomes with their fitness values

Chromosome Number	$I(\boldsymbol{x}_j)$	P_j (Eq. 4.5)	Range
1	25	0.25	$0 \le R \le 0.25$
2	5	0.05	$0.25 \le R \le 0.3$
3	40	0.40	$0.3 \le R \le 0.7$
4	10	0.10	$0.7 \le R \le 0.8$
5	20	0.20	$0.8 \le R \le 1$

The crossover operation is now carried out on the chromosomes of the gene pool. We first select two strings in the gene pool, randomly. The chromosomes in the gene pool are assigned a number from 1 to N_p. The first random number, R ($0 \le R \le 1$), is generated. This is multiplied by N_p and rounded off into an integer. The chromosome in the gene pool

corresponding to this integer is selected. A second chromosome is then selected similarly. We check if we need to carry out crossover on this pair, using a crossover probability, P_{cross}. A random number in the range [0, 1] is generated for the selected pair. Crossover is performed if this number happens to lie between 0 to P_{cross}. If the random number lies in [P_{cross}, 1], we copy the pair without carrying out crossover. This procedure is repeated to give N_p chromosomes, with 100(1 - P_{cross}) % of these being the same as the parents. This helps in preserving some of the *elite* members of the parent population in the next generation.

Crossover involves selection of a location (crossover site) in the string, randomly, and swapping the two strings at this site, as shown below:

1001 1¦100		1001 1¦101	
1011 0¦101	⇒	1011 0¦100	(4.6)
parent chromosomes		daughter chromosomes	

In the above pair, there are seven possible internal crossover sites. Generating a random number, R ($0 \leq R \leq 1$), and comparing it with zones [$0 \leq R \leq 1/7$; . . . ; $(6/7 \leq R \leq 1)$] in an equi-partitioned roulette wheel, helps decide the crossover site in this pair.

If we somehow have a population in which all chromosomes happen to have a 1 in, say, the first location (as in the two strings in Eq. 4.6), it will be impossible to get a 0 at this position using crossover. If the optimal solution has 0 at the first position, then we will not be able to obtain it. Similar is the problem at all locations. This drawback is overcome through the mutation operation that follows crossover. In this, all the individual binaries in *all* N_p chromosomes (including the ones copied unchanged from the parent generation), are checked and changed from 1 to 0 (or *vice-versa*) with a small mutation probability, P_{mut}, i.e., if the random number generated corresponding to any binary lies between 0 to P_{mut}, mutation is performed. Too large a value of P_{mut} leads to oscillations of the solutions. The creation of N_p daughter chromosomes using these three operations: selection, crossover and mutation, completes one generation. The process is repeated with the daughter chromosomes becoming the parents in the next generation.

The crossover and mutation operations may create inferior strings but we expect them to be eliminated over successive generations by the reproduction operator (survival of the fittest). Since SGA works probabilistically with several solutions simultaneously, we can get multiple optimal solutions, if present. For the same reason, SGA does not (normally) get stuck in the vicinity of local optimal solutions, and so is quite robust.

Constraints (other than bounds) of the kind, $g_i(x) \le 0$; $i = 1, 2, \ldots, p$, can be taken care of by subtracting these (for a maximization problem) in a weighted form, from the objective function, and maximizing the modified fitness function. These terms act as penalties (Deb, 1995) when any constraint is violated, since they reduce the value of the modified fitness function, thus favoring the elimination of that chromosome over the next few generations. The following example (Deb, 1995) illustrates the procedure:

$$\text{Max } I(x_1, x_2) = \frac{1}{1 + [(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2]}$$

Subject to

constraint: $(x_1 - 5)^2 + x_2^2 - 26 \ge 0$

bounds: $0 \le x_1 \le 5\ ;\ 0 \le x_2 \le 5$ (4.7)

The modified problem to take care of the constraint by the penalty function approach is written as:

$$\text{Max } I(x_1, x_2) = \frac{1}{1 + [(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2]} - w_1[(x_1 - 5)^2 + x_2^2 - 26]^2$$

Subject to: $0 \le x_1 \le 5\ ;\ 0 \le x_2 \le 5$ (4.8)

In Eq. 4.8, w_1 is taken to be a large positive number (depending on the value of the original objective function) in case the constraint is violated; else it is assigned a value of zero. Equality constraints can be handled in a similar manner. The results for this problem for the 40th generation are shown in Fig. 4.4. The computational parameters used are: $l_{string} = 10$,

$P_{cross} = 0.85$, $P_{mut} = 0.01$, $N_p = 100$, and $w_1 = 10^3$. Fig. 4.4 shows that most of the solutions crowd around the optimal point, $(0.829, 2.933)^T$. There are still a few points that violate the constraint.

SGA has several advantages over the traditional techniques. This technique is better than calculus-based methods (both direct and indirect methods) that may get stuck at local optima, and that may miss the global optimum. This technique does not need derivatives, either.

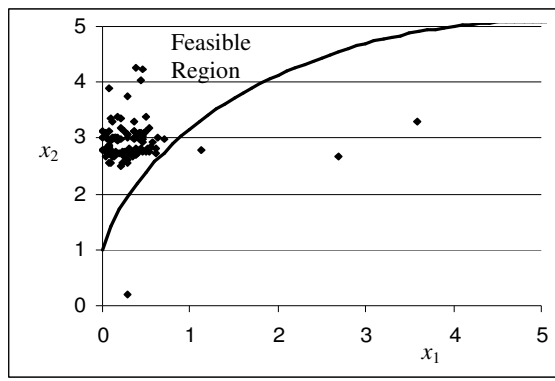


Fig. 4.4 Population at the 40th generation for the constrained optimization problem in Eq. 4.8

4.2.2 *Multi-Objective Elitist Non-Dominated Sorting GA (NSGA-II) and its JG Adaptations*

Several extensions of SGA have been developed (Deb, 2001) to solve problems involving MOO. One of the more popular algorithms is the elitist NSGA-II (Deb *et al.*, 2002). This algorithm has been used extensively to solve a variety of MOO problems in chemical engineering (see Chapter 2), and is now being described using Eq. 4.1 as an example. We generate, randomly, N_p parent chromosomes (as in SGA), in box P (see Fig. 4.5).The binary substrings are mapped and the model is used to evaluate both $I_1(\boldsymbol{x})$ and $I_2(\boldsymbol{x})$. We create a new box, P′, having N_p locations. The first chromosome, 1 (referred to as C_1), is *transferred* (deleted from P) from P to P′.Then the next chromosome, C_i, is taken

temporarily to box P′ sequentially ($i = 2, 3, \ldots, N_p$). C_i is compared with *all* the other chromosomes present in P′, one by one.

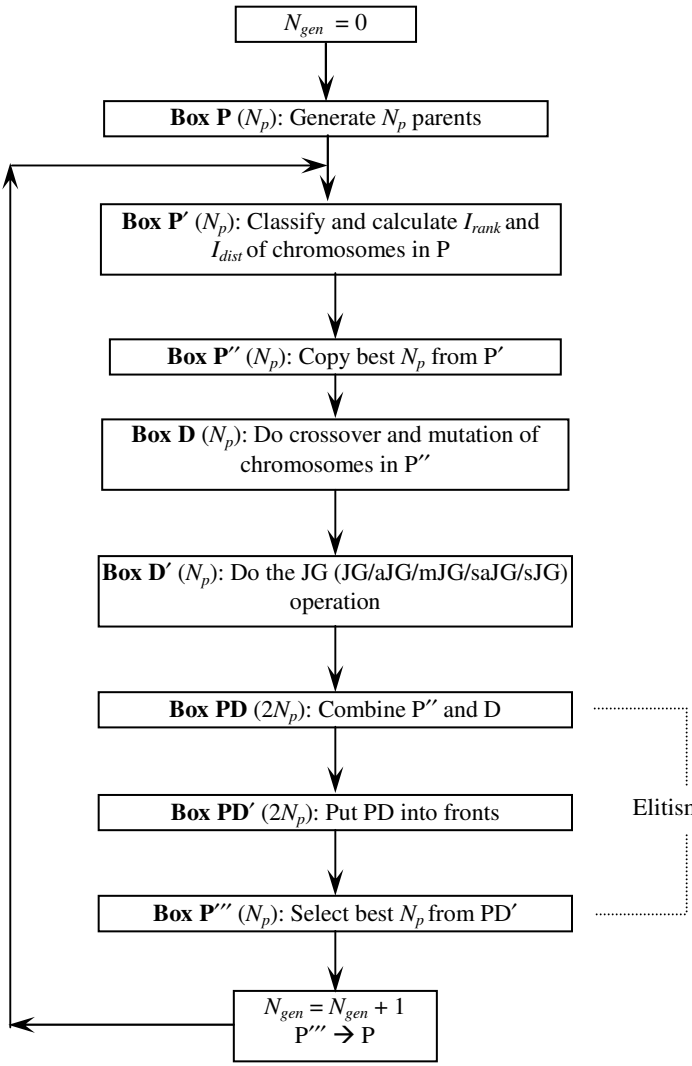


Fig. 4.5 Flowchart for NSGA-II-JG/aJG/mJG/saJG/sJG

If C_i dominates the member of P′ with which it is being compared (i.e., both I_1 and I_2 of C_i are better than those of this member), the inferior

point is removed from P′ and put back into P at its old position (and deleted from P′). If C_i is dominated over by this member, C_i is *returned* to box P and the comparison of C_i stops (and we study the next member, $C_{i+1,}$ from P). If C_i and this chromosome in P′ are non-dominated (i.e., C_i is better than the chromosome in P′ with respect to one fitness function, but worse with respect to the other fitness function), both C_i and the other member are kept in P′. The comparison of C_i with the remaining members of P′ is continued, and either C_i is finally kept in P′, or returned to P. This is continued till all the N_p members in P have been explored. At this stage we have only ($\leq N_p$) non-dominated chromosomes in P′. We say that these comprise the first front, and assign all of these chromosomes a rank of 1 (i.e., $I_{rank,i} = 1$). We now close this sub-box in P′. Further fronts (with $I_{rank,i} = 2, 3, \ldots$, etc.) are generated in P′, using members left in P, till P′ is full, i.e., all N_p members in P are classified into fronts. It is obvious that all the chromosomes in front 1 are the best, followed by those in fronts 2, 3, . . . , etc.

In MOO, we wish to have a good spread of the non-dominated chromosomes (Pareto set), as in Fig. 4.1. In order to achieve this, we try to de-emphasize (kill slowly) solutions that are closely spaced. This is done by assigning a crowding distance, $I_{dist,i}$, to each chromosome, C_i, in P′. We select a front, and re-arrange its chromosomes in order of increasing values of *any one* of the two (for a two-objective problem) fitness functions, say, I_1. Then we find the largest rectangle enclosing chromosome, C_i, in the front, which just touches its nearest neighbors. If there are more than two objectives, we re-arrange the chromosomes in terms of increasing values of *any one* of the fitness functions, and look at the largest hyper-cuboid enclosing chromosome, C_i, which just touches its nearest neighbors. The crowding distance for C_i is then calculated as:

$$I_{dist,i} = \frac{1}{2} \text{ [sum of all the sides of this hyper-cuboid]}$$

$$= \frac{1}{2} \sum_{j=1}^{no.ofobjectives} \left(I_{j,i+1} - I_{j,i-1} \right) \tag{4.9}$$

In Eq. 4.9, $I_{j,i}$ is the value of the jth fitness function of the ith (re-arranged) chromosome. The lower the value of $I_{dist,i}$, the more crowded is the chromosome, C_i. Boundary chromosomes (those having the highest

and lowest values of the selected fitness functions during re-arrangement) are assigned (arbitrarily) high values of $I_{dist,i}$, so as to prevent their being killed. Clearly, if we look at any two chromosomes, i and j, in P′, C_i is better than C_j if $I_{rank,i} < I_{rank,j}$. If, however, $I_{rank,i} = I_{rank,j}$, then C_i is better than C_j if $I_{dist,i} > I_{dist,j}$.

Now that we know how to compare chromosomes in P′, we start to copy them in a gene pool (box P″, having N_p locations). We use the tournament selection procedure. We take any two members from P′ *randomly*, and make a copy of the better of these two chromosomes into P″. We then put both the chromosomes back into P′. This is repeated till P″ has N_p members. Crossover and mutation are now carried out on the chromosomes in P″, as in SGA, finally giving N_p daughters in box D. These are then copied in box D′ for further processing.

4.2.2.1 *Jumping Genes/Transposons (Stryer, 2000)*

As mentioned at the beginning, the speed of convergence of NSGA-II can be improved by the use of the concept of jumping genes [JG; or transposons, predicted by McKlintock (1987)] in biology. In biology, JG is a short DNA that can jump in and out of chromosomes. Initially, scientists considered DNA as stable and invariable, and so the idea of JG met with considerable cynicism. But in the late 1960s, scientists succeeded in isolating JGs from the bacterium, *E. coli*, and named these as transposons. In the 1970s, the role of transposons in transferring bacterial resistance to anti-bodies became understood, and led to increased interest in their studies. At the same time, it was found that transposons also generated genetic variations (diversity) in natural populations. It was observed that these extra-chromosomal transposons were not essential for normal life, but could confer on it properties such as drug resistance and toxigenicity, and, under appropriate conditions, offer advantages in terms of survival. In fact, up to almost 20 % of the genome of an organism could be comprised of transposons. The concept of JG has been exploited to give several JG adaptations, both for the real-coded (Ripon *et al.*, 2007) as well as the binary-coded NSGA-II. Some popular versions of the latter are described below.

4.2.2.2 *(Variable-Length) Binary-Coded NSGA-II-JG (Kasat and Gupta, 2003)*

Kasat and Gupta (2003) found that the binary-coded NSGA-II can be improved significantly by replacing (variable-length) segments of binaries (genes) by randomly-generated jumping genes (see Fig. 4.6). A chromosome in box D′ is checked to see if the JG adaptation is to be carried out on it, using a probability, P_{JG}. If so, two locations, p and q (both integers), are identified randomly on it, and the binary sub-string in-between these points is replaced with a *newly* (randomly) generated string (rs in Fig. 4.6) of the same length. Only a single transposon is assumed to be present in any selected chromosome and the length, n_{chr}, is kept unchanged. This is done to keep the algorithm, NSGA-II-JG, simple. The replacement procedure involves a macro-macro-mutation, and is expected to provide higher genetic diversity.

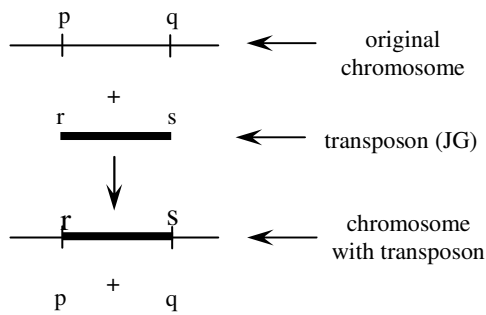


Fig. 4.6 The replacement by a JG in a chromosome

4.2.2.3 *(Fixed-Length) NSGA-II-aJG*

More recently, another adaptation of jumping genes, NSGA-II-aJG, has been developed by Bhat *et al*. (2006) and Bhat (2007). In this fixed-length (f_b binaries) JG adaptation, a probability, P_{JG}, is used to see if a chromosome is to be modified. If yes, a single site only is identified (randomly) in it, from location 1 to n_{chr} - f_b. The other end of the JG is selected f_b binaries beyond it. The sub-string of binaries in-between these

sites is replaced with a *newly* (randomly) generated binary string having the same length.

4.2.2.4 *NSGA-II-mJG ('modified' JG)*

In several problems, the *global*-optimal values of some of the decision variables may lie *exactly* at their bounds, as for example, in froth flotation circuits used for mineral beneficiation. In this problem, the fraction of the flow of an exit stream from a cell going to another cell may lie exactly at its bounds (0 or 1). Since the binary-coded NSGA-II is based on three probabilistic operators (viz., reproduction, crossover and mutation), it is difficult to generate solutions having such values (involving a sequence of several zeros or ones in the sub-string describing the decision variable). A similar problem is also encountered with the binary-coded NSGA-II-JG, where a sub-string of a chromosome, identified randomly, is replaced by a randomly generated new binary JG. This new sub-string may not coincide with a *single* decision variable (it may, in fact, extend over a few decision variables). Thus, it is not easy to generate chromosomes with decision variables exactly at their bounds using these algorithms. This particular problem was addressed by Guria *et al*. (2005b), who developed the modified jumping gene (mJG) operator. A fraction, P_{mJG}, of chromosomes is selected randomly from the daughter population in box D′ (Fig. 4.5). The decision variable (rather than the position of a binary number) to be replaced in these is then identified, using an integral random number lying between 1 and n. The entire set of binaries associated with this variable is replaced either by a sequence of *all* zeros or *all* ones, using the probability, $P_{11\ldots1}$. Replacement of more than one decision variable in a chromosome has also been investigated (Guria *et al*., 2005b) but has not been found to be of much use.

4.2.2.5 *NSGA-II-saJG ('specific adapted' JG)*

Two further adaptations of JG, namely, saJG and sJG, have been developed recently by Agarwal and Gupta (2007a). These do away with a *user-defined* fixed length. The saJG adaptation is used only if each

decision variable is coded using the same number, l_{string}, of binary numbers. The probability, P_{saJG}, of carrying out the saJG operation on a chromosome is used to find out if this operation is to be carried out. If yes, a starting location of the saJG is selected randomly. This position need not correspond to the beginning of a decision variable. A fixed number, $f_b = l_{string}$, of binaries starting from this point is then replaced by new, randomly generated binaries (see Fig. 4.7a). Thus, in a single saJG operation, one, or at most, two *adjacent* decision variables are changed.

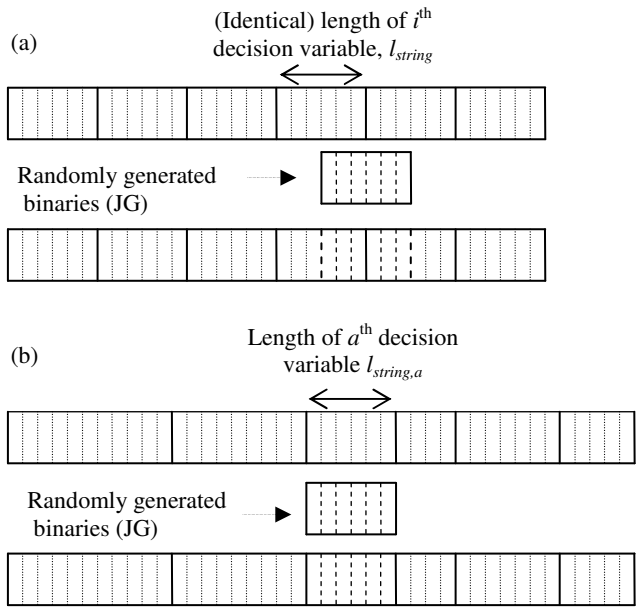


Fig. 4.7 The (a) saJG and (b) sJG operations

4.2.2.6 *NSGA-II-sJG ('specific' JG)*

The user specified probability, P_{sJG}, of carrying out the sJG operation on a chromosome is used to find out if this operation is to be carried out on a chromosome. If this operation is to be performed, a random number is generated and converted into an integral value, a, lying between 1 and n. All the $l_{string,a}$ ($l_{string,a}$ may or may not be equal for different decision variables) binaries of the ath decision variable are replaced by new,

randomly-generated, binaries (see Fig. 4.7b). Thus, a macro-macro mutation of *only* the ath decision variable is carried out.

After the appropriate JG operation is completed (in box D′, Fig. 4.5), the N_p better parents (present in box P″) and all the N_p daughters (after crossover, mutation and the JG operations; present in box D′) are copied into a new box, PD, having $2N_p$ locations. These $2N_p$ chromosomes are re-classified into fronts (in box PD′), using the concept of domination. The *best* N_p chromosomes are taken from box PD′ and put into box P‴ (of size N_p), front-by-front. In case only a few members are needed from the final front to fill up this box, the concept of crowding distance is used. It is clear that this procedure, called elitism, collects the best members from the parents as well as the daughters.

This completes one generation. The members in P‴ are the parents in the next generation.

4.3 Simulated Annealing (SA)

4.3.1 *Simple Simulated Annealing (SSA) for Single-Objective Problems*

SSA (Metropolis *et al*., 1953; Kirkpatrick *et al*., 1983) is inspired by the solidification of molten metals through annealing (slow cooling). The rate of cooling is extremely important in the annealing process. At higher temperatures, the atoms/molecules in a system have large energies and so a few atoms/molecules can come together and form nuclei. At lower temperatures, the atoms or molecules are less energetic (the Boltzmann distribution), and only small excursions around the nuclei take place. This helps in the growth of nuclei. These physical phenomena are mimicked in SSA.

A *single* feasible point is selected initially and the corresponding objective function, I, is evaluated. A new point is then generated *randomly*. The objective function is evaluated at the new point again. The change in the objective function, ΔI ($\equiv I_{new} - I_{old}$), is computed. For a minimization problem, if ΔI is negative (i.e., I decreases), the new point is better than the previous one, and is accepted with a probability of unity. However, if ΔI is positive (the new point is worse than the

previous one), the new point may be accepted but with a probability of exp(-Δ*I*/*T*). This enables some inferior points to be accepted, leading to diversity in the search. Here, *T* is a computational parameter, referred to as the *temperature*. The probability of acceptance of an inferior point, exp(-Δ*I*/*T*), is similar to the Boltzmann factor in statistical mechanics. Initially, the value of *T* is kept large. This enables more points to be accepted (see Fig. 4.8) and leads to extensive exploration of the search space (similar to the nucleation step in solidification). As the number of iterations increases, *T* is reduced in a programmed manner. This leads to *local* exploration around the earlier solutions (growth of nuclei). Fig. 4.8 shows how the probability of acceptance of the inferior solutions is lower at lower values of *T*. Of course, better solutions are always accepted with a probability of unity, i.e., the value of exp(-Δ*I*/*T*) is truncated to 1.0. It is to be noted that in SSA we work with only one solution at any time, in sharp contrast to what is done in SGA.

4.3.2 *Multi-Objective Simulated Annealing (MOSA)*

The procedure used in SSA has been extended to multi-objective problems by Suppapitnarm *et al*. (2000). These workers used the neighborhood perturbation method of Yao *et al*. (1999) to create a new point around an old point. This algorithm is known as multi-objective simulated annealing (MOSA). Since a Pareto set of solutions is to be

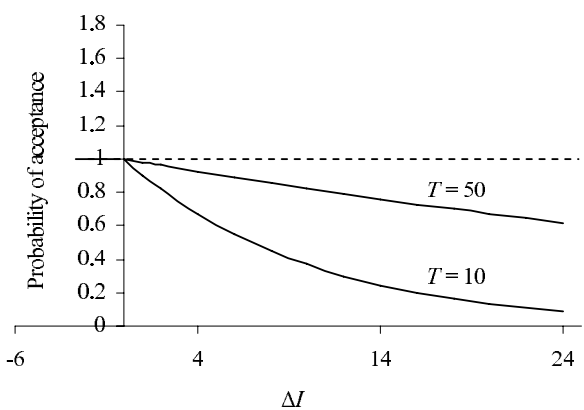


Fig. 4.8 Probability of acceptance of any solution in SSA

obtained for such problems, the solutions obtained using SA, one by one, need to be archived in files, and processed appropriately using the concept of non-dominance and crowding. This is similar to the procedure followed in NSGA-II. The complete details of MOSA, along with its two JG adaptations, are summarized in the Appendix.

4.3.2.1 *MOSA-JG*

The variable and fixed-length JG adaptations of MOSA have been developed by Sankararao and Gupta (2006). A fraction, P_{JG}, of points/solutions (selected randomly) in the *archive* of feasible (not necessarily non-inferior) points (in File 1) is modified by the jumping gene operator (of either kind). The solution selected for the JG operation comprises of N_d decision variables, $\boldsymbol{x}$, which are real numbers (and not a sequence of binaries, as in GA). Two random numbers, *digitized* appropriately, are generated (in MOSA-JG) to identify the sequence of *decision variables* to be replaced. Random numbers, R_i, between 0 and 1 are then used to replace the decision variables so identified, using $x_i^{(new)} = x_{i,L} + R_i\ (x_i^U - x_{i,L})$. The other steps are identical to those in MOSA. Details are provided in the Appendix.

4.3.2.2 *MOSA-aJG*

In the fixed-length JG adaptation, MOSA-aJG, a random number is used to identify a decision variable, p (an integer lying between 0 and N_d - f_b). This is one end of the JG. The other end of the JG lies at decision variable, $p + f_b$, where f_b is the (integral) number of decision variables in the JG. The other steps are identical to those in MOSA. The details of MOSA-aJG are provided in the Appendix.

4.4 Application of the Jumping Gene Adaptations of NSGA-II and MOSA to Three Benchmark Problems

The performance of the different JG adaptations of NSGA-II is studied using three benchmark problems, ZDT4, ZDT2 and ZDT3 (Deb, 2001). Similarly, the performance of MOSA and its JG adaptations (only JG and

aJG), and of NSGA-II and its JG adaptations, is studied using the ZDT4 problem. The benchmark problems are first described.

Problem 1 (ZDT4)

$$\text{Min } I_1 = x_1 \tag{4.10a}$$

$$\text{Min } I_2 = g(\boldsymbol{x})[1 - [I_1/g(\boldsymbol{x})]^{1/2}] \tag{4.10b}$$

where g(**x**) [the Rastrigin function] is given by

$$g(\boldsymbol{x}) \equiv 1+10(n-1)+\sum_{i=2}^{n}\left\{x_i^2 - 10\cos(4\pi x_i)\right\} \tag{4.10c}$$

$$\text{subject to : } 0 \le x_1 \le 1 \tag{4.10d}$$

$$-5 \le x_j \le 5; j = 2, 3, \ldots, n \tag{4.10e}$$

with n = 10. This problem has 99 Pareto fronts, of which only one is the *global* optimal. The latter corresponds to the first decision variable, x_1, lying between 0 and 1 (and so, $0 \le I_1 \le 1$). All the other decision variables, x_j; $j = 2, 3, \ldots, 10$, corresponding to the *global* Pareto set have values equal to 0 (and so, $0 \le I_2 \le 1$). The binary-coded NSGA-II as well as PAES (Knowles and Corne, 2000) have been found (Deb, 2001) to converge to *local* Paretos, rather than to the *global* optimal set. The *real coded* NSGA-II has been found to converge to the global Pareto-set, though.

Problem 2 (ZDT2)

$$\text{Min } I_1 = x_1 \tag{4.11a}$$

$$\text{Min } I_2 = g(\boldsymbol{x})[1 - [I_1/g(\boldsymbol{x})]^2] \tag{4.11b}$$

where g(**x**) is given by

$$g(\boldsymbol{x}) \equiv 1+\frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{4.11c}$$

$$\text{subject to: } 0 \le x_j \le 1;\ j = 1, 2, \ldots, n \tag{4.11d}$$

with $n = 30$. The Pareto-optimal front corresponds to the first decision variable, x_1, lying between 0 and 1. All the other decision variables, x_j; $j = 2, 3, \ldots, 30$, corresponding to the *global* Pareto set have values equal to 0. The complexity of the problem lies in the fact that the Pareto front is non-convex.

Problem 3 (ZDT3)

$$\text{Min } I_1 = x_1 \tag{4.12a}$$

$$\text{Min } I_2 = g(\boldsymbol{x})[1 - \{I_1/g(\boldsymbol{x})\}^{1/2} - \{I_1/g(\boldsymbol{x})\}\sin(10\pi I_1)] \tag{4.12b}$$

where g(**x**) is given by

$$g(\boldsymbol{x}) \equiv 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{4.12c}$$

$$\text{subject to: } 0 \le x_i \le 1;\ i = 1, 2, \ldots, n \tag{4.12d}$$

with $n = 30$. The Pareto-optimal front corresponds to $x_j = 0$; where $j = 2, 3, \ldots, 30$. This problem is a good test for any MOO algorithm since the Pareto front is discontinuous.

4.5 Results and Discussion (Metrics for the Comparison of Results)

The three benchmark problems are solved using four techniques, namely, NSGA-II-JG, NSGA-II-aJG, NSGA-II-saJG and NSGA-II-sJG. The results for the ZDT2 and ZDT3 problems are given in the CD (all extra material and the codes are given in the *Folder*: Chapter 4), while those for the ZDT4 problem only are presented here. The best values of the computational parameters are found by trial for all the problems, and for each of the algorithms used. These are given in Table 4.2 for the ZDT4 problem and in Table CD2 of the CD for the other two problems. Fig. 4.9 gives the results using NSGA-II-sJG and NSGA-II-JG for the ZDT4 problem, while those for the ZDT2 and ZDT3 problems (using NSGA-II-sJG only) are given in the CD.

It is observed from Fig. 4.9 that NSGA-II-JG has not converged till 1,000 generations (Fig. 4.1 shows that NSGA-II-JG does converge after about 1,600 generations or 320,000 function evaluations) for the ZDT4 problem. The Pareto optimal solutions obtained for the four techniques at the end of 1,000 generations (involving almost equal computational effort) cannot be compared *visually* (except in cases where converged results have not been obtained) and require some metrics for comparison. Three metrics (Deb, 2001) are commonly used for detailed comparison of the results. These are: the set-coverage metric (a matrix), $\boldsymbol{C}$, the spacing, S, and the maximum spread, MS. The elements, $C_{p,q}$, of the set-coverage matrix represent the fraction of solutions in q that are weakly

dominated by the solutions in p. If the solution is strictly better than other solutions in at least one objective, then it is referred to as a weakly dominated solution with respect to the other.

Table 4.2 GA parameters for Problem 1 (Agarwal and Gupta, 2007a)[+]

Parameter	Value	Parameter	Value
N_p	100	P_{JG}^{*}	0.50
$N_{gen,max}$	1000	P_{aJG}^{**}	0.50
N_{seed}	0.88876	P_{saJG}^{***}	0.75
l_{chr}	300	P_{sJG}^{****}	0.60
P_{cross}	0.9	f_b^{**}	25
P_{mut}	0.01	f_b^{***}	30

[+] Values for the ZDT2 and ZDT3 problems are given in Table CD2 in the CD

[*] For NSGA-II-JG; [**] for NSGA-II-aJG; [***] for NSGA-II-saJG; [****] for NSGA-II-sJG

The spacing is a measure of the relative distance between *consecutive* (nearest neighbor) solutions in the non-dominated set. The maximum spread is the length of the diagonal of the hyper-box formed by the extreme function values in the non-dominated set. For two-objective problems, this metric refers to the Euclidean distance between the two extreme solutions in the ***I***-space. It is given by

$$S = \sqrt{\frac{1}{Q}\sum_{i=1}^{Q}(d_i - \bar{d})^2} \tag{4.13a}$$

where

$$d_i = \min_{k \in Q;\, k \neq i} \sum_{l=1}^{m} \left| I_l^i - I_l^k \right| \tag{4.13b}$$

and

$$\bar{d} = \sum_{i=1}^{Q} \frac{d_i}{Q} \tag{4.13c}$$

In Eq. 4.13, m is the number of objective functions and Q is the number of non-dominated solutions. Clearly, d_i is the 'spacing' (sum of *each* of the distances in the $\boldsymbol{I}$-space) between the ith point and its *nearest* neighbor, $\overline{d}$ is its mean value, and S is the standard deviation of the different d_i. An algorithm that gives non-dominated solutions having a smaller value of S but larger values of the maximum spread is, obviously, superior.

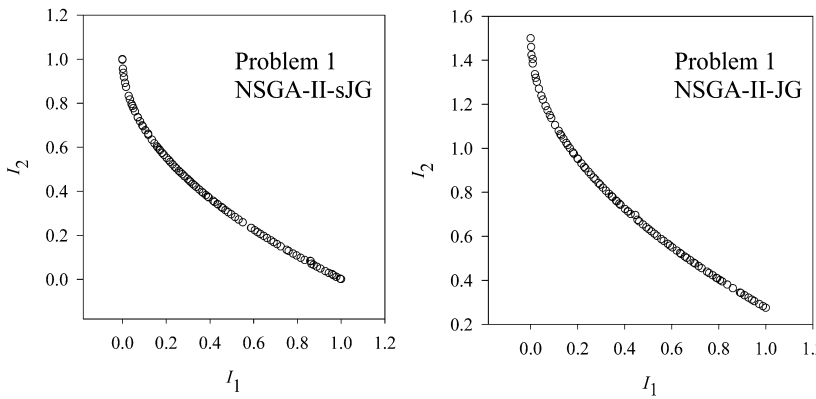


Fig. 4.9 Optimal solutions for the ZDT4 problem after 1,000 generations (200,000 function evaluations) using NSGA-II-sJG and NSGA-II-JG (Agarwal and Gupta, 2007a)

The three metrics for the ZDT4 problem are given in Table 4.3 (the top 4×4 entries represent $\boldsymbol{C}$). It is observed for the ZDT4 problem that NSGA-II-sJG is the best of all the algorithms in terms of spacing, since it gives an optimal front with the lowest spacing. Although, NSGA-II-JG gives a high value of the maximum spread (desirable), but, as it performs badly in terms of the set-coverage and spacing, it is considered inferior. Similarly, it is observed from Table CD3 in the CD that NSGA-II-sJG and NSGA-II-saJG are better than NSGA-II-aJG and NSGA-II-JG for Problem 2 in terms of the set coverage metric, while in terms of the spacing, NSGA-II-aJG is better. NSGA-II-saJG is the best in terms of the maximum spread. The set-coverage metric for Problem 3 (Table CD3) indicates that NSGA-II-aJG is the best. However, in terms of the spacing,

NSGA-II-sJG is superior, while NSGA-II-saJG is the best when the maximum spread is considered.

Yet another method to compare algorithms for MOO problems is the box plots (Chambers *et al*., 1983). These are shown for the ZDT4 problem in Fig. 4.10 and in Fig. CD10 in the CD for the other two cases. These plots show the distribution (in terms of quartiles and outliers) of the points, graphically. For example, the box plot of I_1 for any technique indicates the entire range of I_1 distributed over four quartiles, with 0-25 % of the solutions having the lowest values of I_1 indicated by the lower vertical line with a whisker, the next 25-50 % of the solutions by the lower box, 50-75 % of the solutions by the upper part of the box, and the remaining 75-100 % of the solutions having the highest values of I_1, by the upper vertical line with a whisker. Points beyond the 5 % and 95 % range (outliers) are shown by separate circles. The mean values of $\boldsymbol{I}$ are shown by dotted lines on these plots. A good algorithm should give box plots in which all the regions are equally long, and the mean line coincides with the upper line of the lower box. It is observed that for Problem 1, NSGA-II-sJG gives the best box plot, for Problem 2 (see Fig. CD10), NSGA-II-aJG gives the best box plot, while for Problem 3, NSGA-II-sJG and NSGA-II-saJG give comparable results.

Table 4.3 Metrics for Problem 1 (Agarwal and Gupta, 2007a) after 1,000 generations

	NSGA-II-JG	NSGA-II-aJG	NSGA-II-saJG	NSGA-II-sJG
Set coverage metric				
NSGA-II-JG	-	0.00E+00	0.00E+00	0.00E+00
NSGA-II-aJG	9.90E-01	-	6.00E-02	2.00E-02
NSGA-II-saJG	9.90E-01	4.00E-02	-	4.00E-02
NSGA-II-sJG	9.90E-01	3.00E-02	8.00E-02	-
Spacing	9.27E-03	7.74E-03	6.17E-03	8.21E-03
Maximum spread	1.5809	1.4138	1.4141	1.4138

A study of all the results indicates that NSGA-II-JG is inferior to the other algorithms, at least for the three benchmark problems studied. NSGA-II-sJG and NSGA-II-saJG appear to be satisfactory and comparable. The latter two algorithms do not have the disadvantage of user-defined fixed length of the JG, as required in NSGA-II-aJG.

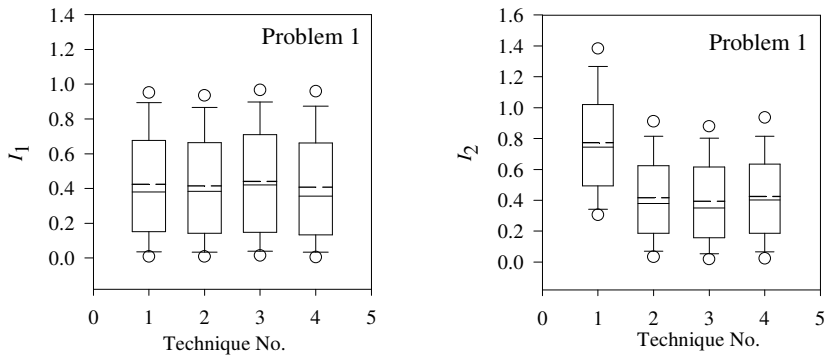


Fig. 4.10 Box plots of I_1 and I_2 for the ZDT4 problem after 1,000 generations; Technique 1: NSGA-II-JG; 2: NSGA-II-aJG; 3: NSGA-II-saJG; 4: NSGA-II-sJG (Agarwal and Gupta, 2007a)

In order to compare the adaptations of the NSGA-II and MOSA families, the ZDT4 problem (Problem 1) is solved using the six techniques, NSGA-II, NSGA-II-JG, NSGA-II-aJG as well as using MOSA, MOSA-JG and MOSA-aJG. Best values of the several computational parameters have been obtained by trials for each of the algorithms. These are given in Table 4.4.

It must be emphasized that new solutions around the less crowded ones keep getting generated in the MOSA family till the computations are stopped. Hence, *S* will keep decreasing (improving) continuously. Better solutions (with lower *S*) are also generated when the NSGA family is used but with higher population sizes (the same solutions get repeated after some generations in this). Hence, these two families of algorithms can be compared *only* when both involve the same number of simulations. This should be done only after a reasonable Pareto set, ascertained by visual inspection (so that the *qualitative* nature of the set

does not change much after this point), is obtained. These are referred to as *near* optimal solutions.

Table 4.4 Parameters used in the adaptations of the GA and SA families for the ZDT4 problem (Sankararao, 2007)

Genetic Algorithm family (NSGA-II, NSGA-II-JG and NSGA-II-aJG)		Simulated Annealing family (MOSA, MOSA-JG and MOSA-aJG)	
Parameter	Problem 1	Parameter	Problem 1
N_p	100	B	1.0
P_{cross}	0.9	N_{seed}	0.88876
P_{mut}	$1/l_{chr}$	C	1.0
P_{jump} [+]	0.5	P_{jump} [*]	0.5
$N_{gen,max}$	400	$\boldsymbol{S}^{(0)}$	0.5
l_{chr}	300	$\boldsymbol{T}$ [**]	150
N_{seed}	0.88876	$N_{T,1}$	5000
f_b [++]	25	$N_{T,2}$	900
Number of function evaluations (FN)	80,000	r_B	1.0
		r_1	1.0
		Φ_1	1.0
		A_{MIN}	6
		f_b [***]	5
		Number of function evaluations (FN)	80,000

\+ Not required for NSGA-II
++ Not required for NSGA-II and NSGA-II-JG
* Not required for MOSA
** Same for all objective functions
*** Not required for MOSA and MOSA-JG

Fig. 4.11 shows the sets of near optimal non-dominated solutions obtained for the ZDT4 problem after 80,000 function evaluations (40,000 for each of the two objective functions). It is observed from this diagram that NSGA-II and NSGA-II-JG have not reached the near optimal stage even after 80,000 function evaluations. In fact, NSGA-II-JG is found to take about 280,000 function evaluations to reach the near optimal stage (converged results for this algorithm are shown in Fig. 4.12).

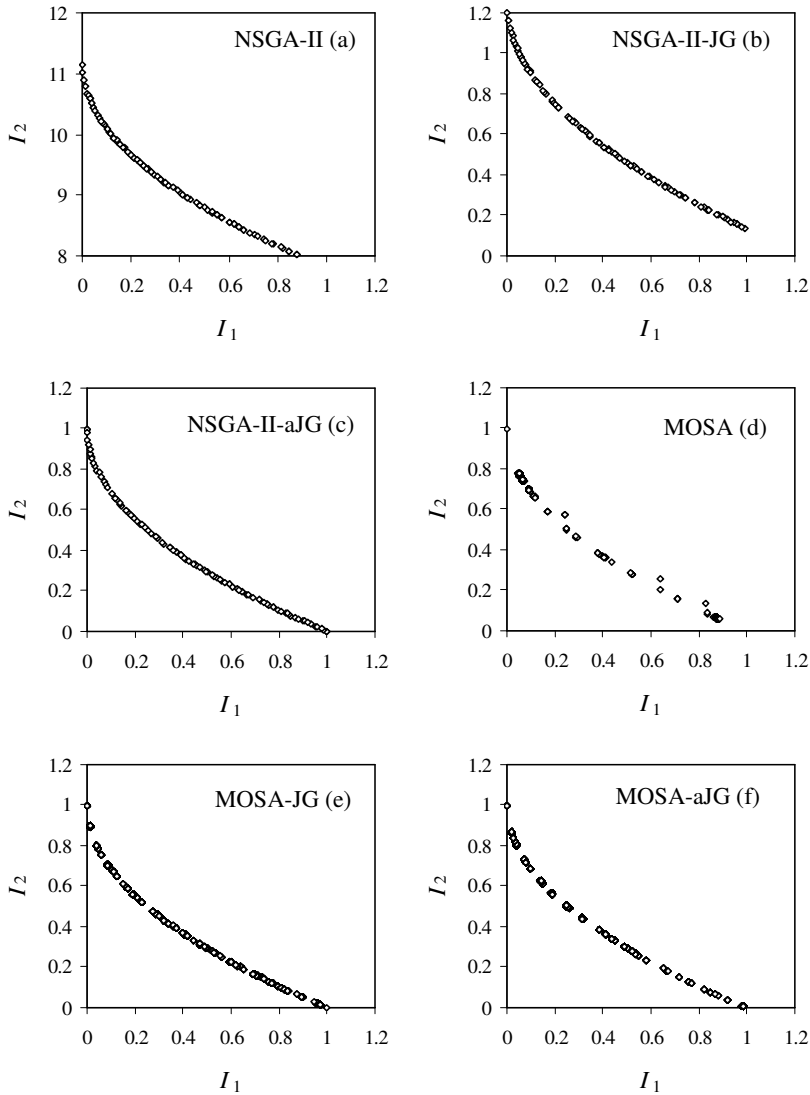


Fig. 4.11 Comparison of the optimal solutions of the ZDT4 problem obtained by different algorithms after 80,000 function evaluations (adapted from Sankararao, 2007)

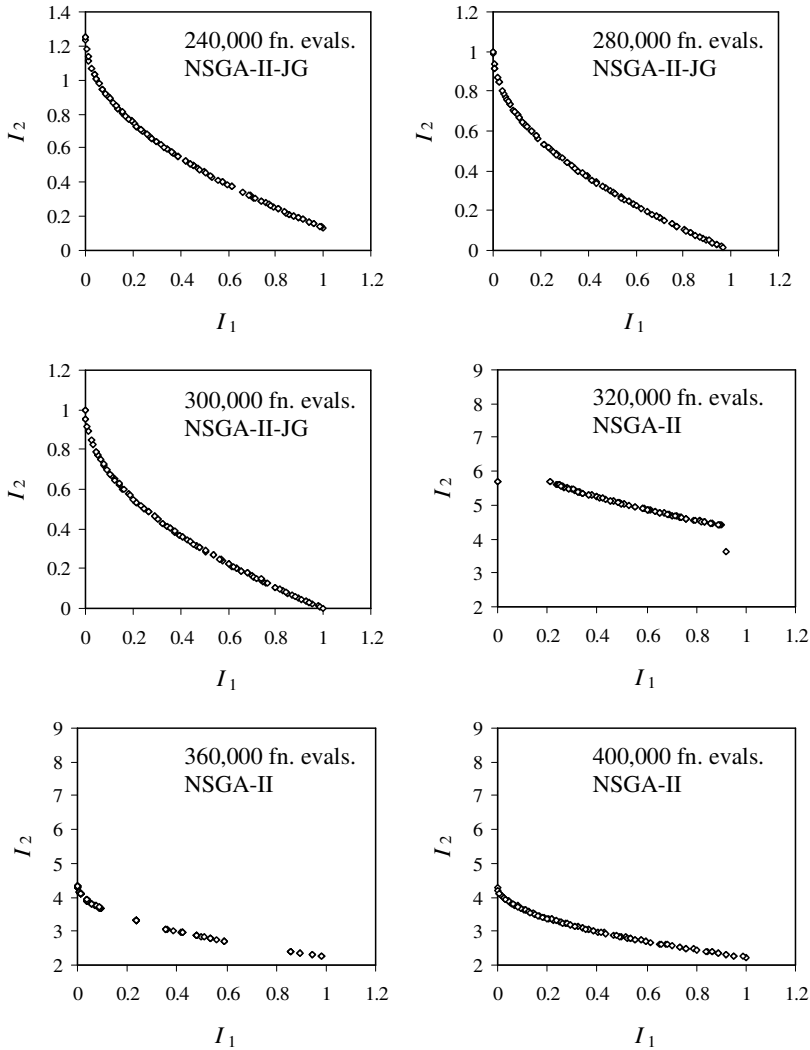


Fig. 4.12 Plots of non-dominated solutions obtained with NSGA-II-JG after 240,000 – 300,000 function evaluations (fn. evals.) and for NSGA-II for 320,000 – 400,000 fn. evals. for the ZDT4 problem. Note that I_1 and I_2 extend over [0, 1] (global Pareto set) for NSGA-II-JG only after about 300,000 function evaluations, and do not show this characteristic for NSGA-II

Table 4.5 Maximum spread and spacing metrics for Problem 1 (number of function evaluations = 80,000) (adapted from Sankararao, 2007)

Algorithm	Metric	
	Maximum spread	Spacing
NSGA-II	--	--
NSGA-II-JG	--	--
NSGA-II-aJG	1.41418	0.00770
MOSA	1.29058	0.02000
MOSA-JG	1.41421	0.01419
MOSA-aJG	1.39525	0.01533

It is also observed (Fig. 4.12) that NSGA-II does not converge to the *global* Pareto optimal set (note that the optimal values of I_2 in Fig. 4.11a for NSGA-II are not in the range of [0, 1], characteristic of the global Pareto set) even after 400,000 function evaluations. The remaining algorithms, NSGA-II-JG, NSGA- II-aJG and the three members of the MOSA family converge to the correct solution (more results are given in Figs. CD12 and CD13).

The values of two metrics, namely, the maximum spread and the spacing after 80,000 function evaluations, are given in Table 4.5. The metrics for NSGA-II and NSGA-II-JG are not given since the results after 80,000 function evaluations have not converged. It is observed from Table 4.5 that MOSA-JG has the largest value of the MS and the least value of the spacing from among the MOSA family. NSGA-II-aJG gives a lower value of the MS but the spacing is lower than that obtained with MOSA-JG. Hence, either of these two algorithms can be selected for this problem.

The Fortran 90 codes for NSGA-II-aJG (adapted from the original FORTRAN code of NSGA-II developed by Deb, http://www.iitk.ac.in/kangal/codes.shtml), and of MOSA-aJG (as developed by Sankararao and Gupta, 2006, Sankararao, 2007) are given in the CD. These can be run on MS PowerStation™ Version 4.0. The input file in the CD also

needs to be used with the MOSA-aJG code. These codes can be modified for the other JG adaptations easily.

It may be mentioned here that though the binary-coded NSGA-II fails to converge to the global optimal solutions for this test problem, the *real-coded* NSGA-II does, indeed, converge to the correct Pareto solution (in 100,000 function evaluations; Deb, 2001). Speeding up of the real-coded NSGA-II using the JG adaptation has also been observed by Ripon *et al.* (2007). Hence, the JG operator is a useful adaptation for NSGA-II for the solution of complex MOO problems.

The use of the jumping gene operator (any of the adaptations) increases the randomness/diversity and, thus, usually gives better results. Among the various JG adaptations discussed above, the sJG adaptation is found to be better for the benchmark problems studied here. But the choice of a particular adaptation is problem-specific, e.g., in froth flotation circuits the mJG adaptation (Guria *et al.*, 2005b) is found to be better.

The effect of varying the parameter, P_{JG}, has been studied by Kasat and Gupta (2003). They have shown that for the ZDT4 problem, the global Pareto set was obtained as early as in the 600th generation, using $P_{JG} = 0.7$. A further increase in P_{JG} results in larger amounts of fluctuations and, thus, requires a higher number of generations for convergence. This clearly indicates that there exists an optimal value of P_{JG}, which, unfortunately, is problem-specific. One expects that the best value of P_{JG} would be lower if the degree of elitism was smaller. A value of P_{JG} of 0.5 is suggested to be a good starting guess for this parameter.

4.6 Some Recent Chemical Engineering Applications Using the JG Adaptations of NSGA-II and MOSA

The jumping gene adaptations of GA and SA have been used extensively for solving problems of industrial importance in chemical engineering (see Chapter 2). Very recently, Ramteke and Gupta (2007) have carried out the MOO of an industrial semi-batch nylon-6 reactor using the rate, $V_T(t)$, of release of vapors, and the temperature, $T_J(t)$, of the jacket fluid, as decision variables. They obtained the two optimal histories using

NSGA-II-aJG and MOSA-aJG. The use of the histories of *two* decision variables leads to better solutions as compared to results obtained with a *single* history only (Mitra *et al*., 1998). Interestingly, it was found that NSGA-II-aJG was superior to MOSA-aJG for this problem. Agarwal and Gupta (2007a) used NSGA-II-sJG/saJG for optimizing a single heat exchanger. They extended their work to optimize heat exchanger networks (Agarwal and Gupta, 2007b). They observed improved and more general solutions than provided by the Pinch technique (Linnhoff and Hindmarsh, 1983). MOSA-JG and MOSA-aJG (in MOSA, aJG and sJG adaptations are the same) have been used by Sankararao and Gupta (2007b) for optimizing a pressure swing adsorption unit.

4.7 Conclusions

The working of two evolutionary algorithms, NSGA-II and MOSA, are explained *step-by-step*. The various JG adaptations of binary-coded NSGA-II as well as MOSA are used to solve three benchmark problems. The use of the JG adaptation leads to faster convergence as compared to the original algorithms. These adaptations are being used regularly for solving complex chemical engineering problems.

Acknowledgement

Partial financial support from the Department of Science and Technology, Government of India, New Delhi [through grant SR/S3/CE/46/2005-SERC-Engg, dated November 29, 2005] is gratefully acknowledged.

APPENDIX

Multi-Objective Simulated Annealing with the JG Adaptations (MOSA-JG/aJG; see Flowchart in Fig. 4.A 1)

Step 0) INITIALIZATION

Initialize $S_i^{(0)}$, $T_{i,START}$

Set counters: $l = 0$, $k = 0$

l is the number of iterations (feasible points; not in any File), and k is the counter for the number of accepted points (in File 1)

Step 1) GENERATION OF INITIAL POINT

Generate the *first* point within the bounds of $\boldsymbol{x}$ using several random numbers, R_i:

$x_i^{(0)} = x_{i,L} + R_i\,(x_i^U - x_{i,L})$; $0 \le R_i \le 1$; $i = 1, 2, \ldots, N_d$

Check constraints; if not feasible, go to Step 1; counters unchanged

If feasible; $l \rightarrow l + 1$ (store in Files 1 and 2)

Normalize the N_d decision variables, $\boldsymbol{x}$, using their upper and lower bounds to give

$\boldsymbol{U}$ ($\equiv[U_1, U_2, \ldots, U_{Nd}]^T$); with $-1 \le \boldsymbol{U} \le 1$:

$U_i^{(k)} = (x_i^{(k)} - x_{i,L})/[(x_i^U - x_{i,L})/2] - 1$; $i = 1, 2, \ldots, N_d$

Step 2) PERTURBATION (GENERATE A NEW POINT)

Generate next point, $\boldsymbol{x}^{(k+1)}$, using

$U_i^{(k+1)} = U_i^{(k)} + R_i\,S_i^{(k)}$; $-1 \le R_i \le 1$; $i = 1, 2, \ldots, N_d$

Convert $\boldsymbol{U}$ to $\boldsymbol{x}$ using

$x_i^{(k+1)} = \{[(U_i^{(k+1)} + 1)/2] \times [x_i^U - x_{i,L}]\} + x_{i,L}$

Check feasibility (bounds; constraints):

If not feasible, go to Step 2; l unchanged

If feasible, $l = l + 1$ (do not store yet); do jumping gene operation (see Step A below) on $\boldsymbol{x}^{(k+1)}$ (to generate a feasible point)

Calculate $I_i^{(k+1)}(\boldsymbol{x}^{(k+1)})$; $i = 1, 2, \ldots, m$

Check if the new point with JG can be archived in File 2 (non-dominated points) (see Step B below)

If yes, put in File 2 and also in File 1 (of accepted points); $k = k + 1$

If not, do not copy this point in File 2, but check for its acceptance (see Step C below) in File 1; k not updated
Check if Step 2 should be repeated or we need to go to Steps 3 and/or 4 (see details below)

Step A: JUMPING GENE OPERATION
Choose one of the two operations: either random-length JG adaptation (JG) or fixed-length JG adaptation (aJG)
a) Random-length JG adaptation
Generate two random numbers
Convert to integers, p_1 and p_2, between 0 and N_d
Replace the decision variables so identified
$x_i^{(new)} = x_{i,L} + R_i\,(x_i^U - x_{i,L})$; $i = p_1$ to p_2
Check if constraints are satisfied. If not, regenerate new JG (p_1 and p_2 unchanged)
b) Fixed-length (f_b) JG adaptation (aJG)
Generate one random number
Convert to an integer, p, between 0 and $N_d - f_b$
The other end of the JG is at $p + f_b$
Replace all the decision variables in-between p and $p + f_b$ using
$x_i^{(new)} = x_{i,L} + R_i\,(x_i^U - x_{i,L})$; $i = p$ to $p + f_b$
Check if constraints are satisfied. If not, regenerate new JG (p unchanged)

Step B: ARCHIVING OF NON-DOMINATED POINTS
Compare $\boldsymbol{x}^{(k+1)}$ with *every* non-dominated member present in File 2, one by one
If $\boldsymbol{x}^{(k+1)}$ is a non-dominated point, copy it in File 2 (as well as in File 1)
If $\boldsymbol{x}^{(k+1)}$ is dominated over by a member already present in File 2, do not include it in File 2 but check if you can copy it in File 1 (Step C below)
If an earlier member is dominated over by this new point, delete the former from File 2 (it is already in File 1)

Step C: PROBABILITY OF ACCEPTANCE

Calculate the probability, $P[\boldsymbol{x}^{(k+1)}]$, of acceptance (into File 1), of the new feasible point

$$P[\boldsymbol{x}^{(k+1)}] = \prod_{i=1}^{m} \exp\left[\frac{-(I_i^{(k+1)} - I_i^{(k)})}{T_i}\right]$$

If $P[\boldsymbol{x}^{(k+1)}] > 1$, $\boldsymbol{x}^{(k+1)}$ is superior to $\boldsymbol{x}^{(k)}$, accept $\boldsymbol{x}^{(k+1)}$ (i. e., use $P = 1$); $k \rightarrow k+1$

If $0 < P[\boldsymbol{x}^{(k+1)}] < 1$, $\boldsymbol{x}^{(k+1)}$ is worse than $\boldsymbol{x}^{(k)}$, accept $\boldsymbol{x}^{(k+1)}$ with a probability of $P[\boldsymbol{x}^{(k+1)}]$ (i.e., generate an R, $0 \le R \le 1$; if $R \le P[\boldsymbol{x}^{(k+1)}]$, accept the point)

If $\boldsymbol{x}^{(k+1)}$ is accepted, copy it in File 1, $k \rightarrow k+1$ and

Update $S_i^{(k)}$ using:

$S_i^{(k+1)} = 0.9\ S_i^{(k)} + 0.21\ [U_i^{(k+1)} - U_i^{(k)}]$; [the second term on the right gives a small effect of the previous step size (in $\boldsymbol{U}$)]

s.t.: $1 \times 10^{-4} \le S_i^{(k+1)} \le 0.5$

Else, return to $\boldsymbol{x}^{(k)}$; no update of k or $S_i^{(k)}$

Step 3) ANNEALING SCHEDULE

The *first* call to the annealing schedule is to be done after $N_{T,1}$ iterations (counter l). T_i is updated using the N accepted points in File 1

$$\sigma_i^2 = \frac{\sum_{j=1}^{N} \left(I_{i,j} - I_{i,AV}\right)^2}{N}; \quad I_{i,AV} = \frac{\sum_{j=1}^{N} I_{i,j}}{N}$$

$T_{i,NEW} = B \times \sigma_i$, where B is an initial annealing parameter [= 1]

Go to Step 4 below

The next call to the annealing schedule is to be done after $N_{T,2}$ *further* iterations, or, after N_A (= 0.4 $N_{T,2}$) further acceptances (in File 1). The *new*, N, accepted points are used to evaluate:

$T_{i,NEW} = \alpha_i\ T_{i,OLD}$

where α_i = MAX [0.5, exp(- 0.7 $T_{i,OLD}/\sigma_{i,NEW}$)]

$$\sigma_{i,NEW}^2 = \frac{\sum_{j=1}^{N} \left(I_{i,j} - I_{i,AV}\right)^2}{N}; \quad I_{i,AV} = \frac{\sum_{j=1}^{N} I_{i,j}}{N}$$

Go to Step 2 (giving a perturbation to the last point) or to Step 4, as required

Step 4) RETURN TO BASE

Return to an uncrowded base point *periodically* (so as to generate a more continuous Pareto set, by locally exploring around uncrowded points in the non-dominated set)

First return to base (selected an uncrowded point; see later) is done after $N_{B,1}$ [$= N_{T,1}$] iterations. Thereafter, the return to base is after $N_{B,j}$ iterations, $j = 2, 3, \ldots$:

$N_{B,2} = 2N_{T,2}$; $N_{B,j} = r_B N_{B,j-1}$; $j = 3, 4, \ldots$

$0 \leq r_B \leq 1$ [$r_B = 0.9$]

s.t.: $N_{B,j} \geq 10$

Generate File 3 (of uncrowded members in File 2)

Normalize I_i (using max and min values) for *all* A_S non-dominated archived points in File 2, so that $0 \leq I_i \leq 1$

Find the crowding distances ($I_{dist,i}$) of *all* A_S points in File 2 (as in NSGA-II)

Copy A_j of the most uncrowded points (including the boundary points) into File 3, where

$A_j = \Phi_j A_S$; $\Phi_1 = 1$; $\Phi_{j+1} = r\Phi_j$; $r = 0.9$; $A_j \geq A_{MIN} = 6$

Use a random number (digitized appropriately) to select the uncrowded base point to be returned to, from File 3

Go to Step 2 (giving a perturbation to this selected uncrowded point)

Comments

Since File 2 contains all the non-dominated solutions till the current iteration, there is no concept of elitism in MOSA. Also, Step A in Step 2 is omitted in MOSA. For SSA, Steps A, B (in Step 2) and 4 are omitted.

* * *

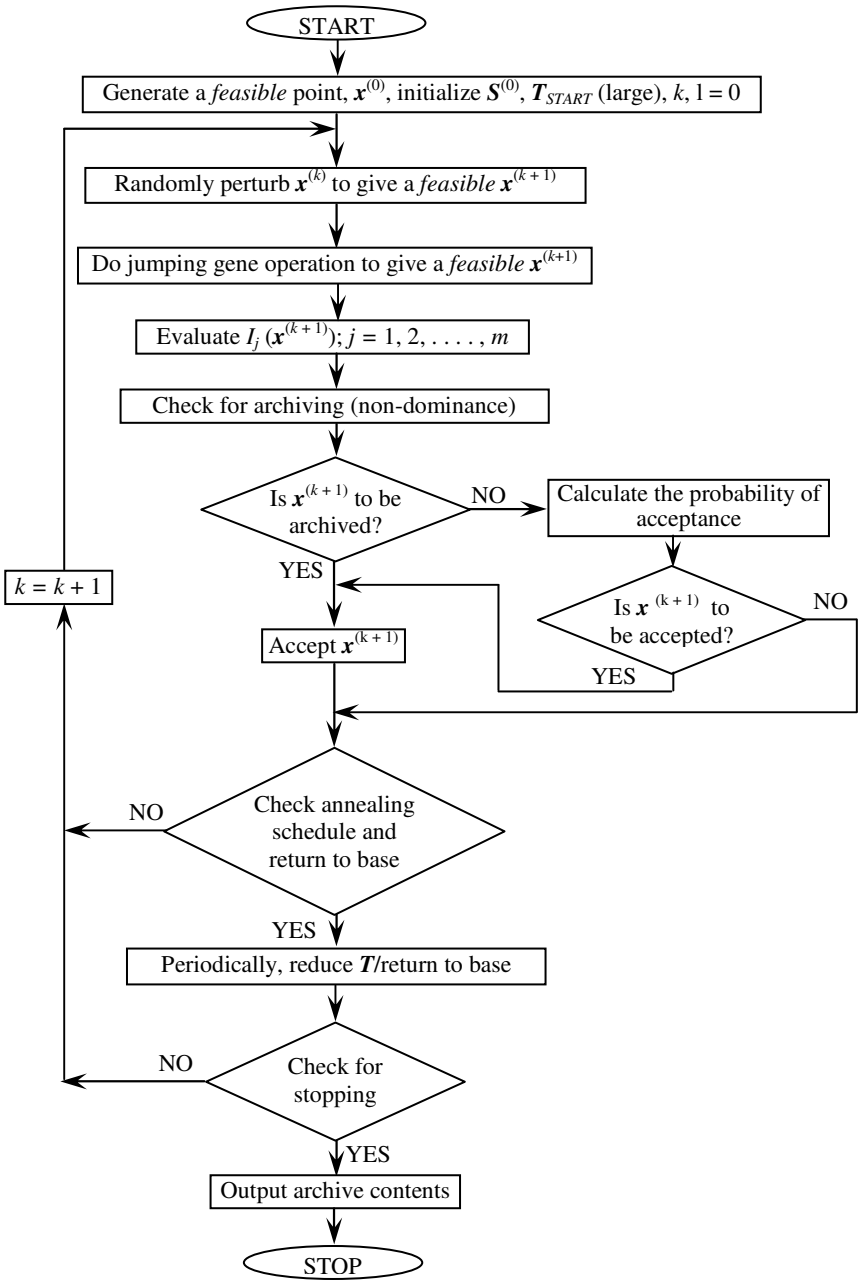


Fig. 4.A 1 Flowchart for MOSA-JG/aJG

Nomenclature

A_{MIN}	lower limit on the number of archived solutions taken in the return-to-base operation
B	parameter for temperature in the annealing schedule
C	parameter used to call the annealing schedule
f_b	fixed length of the JG
I_i	i^{th} objective function
l_{chr}	length of chromosome
$l_{string,i}$	number of binaries used to represent the ith decision variable
m	number of objective functions
N	number of accepted solutions after initial return-to-base
$N_{B,i}$	number of iterations to be performed for call to return-to-base
N_d	number of decision variables in SA
N_{gen}	number of generations
$N_{gen,max}$	maximum number of generations
N_p	population size
n	number of decision variables in GA
N_{seed}	random seed
$N_{T,i}$	number of iterations to be performed before reducing the temperature
P_{aJG}	probability of carrying out the aJG operation
P_{cross}	probability of carrying out the crossover operation
$P_{11\ldots1}$	probability for changing all binaries of a selected decision variable to zero
P_{JG}	probability of carrying out the JG operation
P_{mJG}	probability of carrying out the mJG operation
P_{mut}	probability of carrying out the mutation operation
P_{sJG}	probability of carrying out the sJG operation
P_{saJG}	probability of carrying out the saJG operation
r_B	return-to-base parameter
r_I	parameter to update Φ_i
R	random number
$\boldsymbol{S}^{(k)}$	N_d-dimensional vector of step sizes in the kth acceptance
$\boldsymbol{T}$	m-dimensional vector of computational temperatures in MOSA
$\boldsymbol{U}$	N_d-dimensional vector of normalized decision variables, U_i
$\boldsymbol{x}$	vector of decision variables, x_i

Greek symbols

α_i Temperature reducing parameter

σ_i Standard deviation of accepted solutions

Φ_1 Fraction of archived solutions taken for the return-to-base operation in the first call

Δf Difference in new and old values of the objectivefunction

References

Agarwal, A. and Gupta, S. K. (2007a). Jumping gene adaptations (saJG and sJG) of the elitist non-dominated sorting genetic algorithm (NSGA-II) and their use in multi-objective optimal design of shell and tube heat exchangers, *Chem. Eng. Res. Des.*, (submitted).

Agarwal, A. and Gupta, S. K. (2007b). Multi-objective optimal design of heat exchanger networks using new adaptations of the elitist non-dominated sorting genetic algorithm, *Ind. Eng. Chem. Res.*, (in press).

Bhat, S. A. (2007). On-line optimizing control of bulk free radical polymerization of methyl methacrylate in a batch reactor using virtual instrumentation, Ph.D. Thesis, Indian Institute of Technology, Kanpur, 146 pages.

Bhat, S. A., Gupta, S., Saraf, D. N. and Gupta, S. K. (2006). On-line optimizing control of bulk free radical polymerization reactors under temporary loss of temperature regulation: experimental study on a 1-L batch reactor, *Ind. Eng. Chem. Res.*, 45, pp. 7530-7539.

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983). *Graphical Methods for Data Analysis*, Wadsworth, Belmont, CA.

Coello Coello, C. A., Veldhuizen, D. A. V. and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic/Plenum Publishers, New York.

Chan, T. M., Man, K. F., Tang, K. S. and Kwong, S. (2005a). A jumping gene algorithm for multiobjective resource management in wideband CDMA systems, *Comput. J.*, 48, pp. 749-768.

Chan, T. M., Man, K. F., Tang, K. S. and Kwong, S. (2005b). Optimization of wireless local area network in IC factory using a jumping-gene paradigm, *3rd IEEE International Conference on Industrial Informatics (INDIN)*, pp. 773-778.

Deb, K. (2001). *Multi-objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. A. (2002). Fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, 6, pp. 181-197.

Deb, K. (1995). *Optimization for Engineering Design: Algorithms and Examples*, Prentice Hall of India, New Delhi, India.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, Reading, MA, USA.

Guria, C., Bhattacharya, P. K. and Gupta, S. K. (2005a). Multi-objective optimization of reverse osmosis desalination units using different adaptations of the non-dominated sorting genetic algorithm (NSGA), *Comput. Chem. Eng.*, 29, pp. 1977-1995.

Guria, C., Verma, M., Mehrotra, S. P. and Gupta, S. K. (2005b). Multi-objective optimal synthesis of froth-floatation circuits for mineral processing using the jumping gene adaptation of genetic algorithm, *Ind. Eng. Chem. Res.*, 44, pp. 2621-2633.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA.

Kasat, R. B. and Gupta, S. K. (2003). Multiobjective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm with the jumping gene operator, *Comput. Chem. Eng.*, 27, pp. 1785-1800.

Kirpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). Optimization with simulated annealing, *Science*, 220, pp. 671-680.

Knowles, J. D. and Corne, D. W. (2000). Approximating the non-dominated front using the Pareto archived evolution strategy, *Evol. Comput.*, 8, pp. 149-172.

Linnhoff, B. and Hindmarsh, E. (1983). The pinch design method for heat exchanger network, *Chem. Eng. Sci*., 38, pp. 745-763.

Man, K. F., Chan, T. M., Tang, K. S. and Kwong, S. (2004). Jumping genes in evolutionary computing, *The 30th Annual Conference of IEEE Industrial Electronics Society (IECON'04)*, Busan, Korea.

McClintock, B. (1987). The discovery and characterization of transposable elements. In *The Collected Papers of Barbara McClintock*, Garland, New York, USA.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953). Equation of state calculations by fast computing machines, *J. Chem. Phys.*, 21, pp. 1087-1092.

Mitra, K., Deb, K. and Gupta, S. K. (1998). Multiobjective dynamic optimization of an industrial nylon 6 semibatch reactor using genetic algorithm, *J. Appl. Polym. Sci*., 69, pp. 69-87.

Ramteke, M. and Gupta, S. K. (2007). Multi-objective optimization of an industrial nylon-6 semi batch reactor using the a-jumping gene adaptations of genetic algorithm and simulated annealing, *Poly. Eng. Sci.,* in press.

Ripon, K. S. N., Kwong, S. and Man, K. F. (2007). Real-coding jumping gene genetic algorithm (RJGGA) for multi-objective optimization, *Inf. Sci.*, 177, pp. 632-654.

Sankararao, B. (2007). Multi-objective optimization of chemical engineering units using adaptations of simulated annealing, Ph.D. Thesis, Indian Institute of Technology, Kanpur, 224 pages.

Sankararao, B. and Gupta, S. K. (2006). Multi-objective optimization of the dynamic operation of an industrial steam reformer using the jumping gene adaptations of simulated annealing, *Asia-Pac. J. Chem. Eng.*, 1, pp. 21-31.

Sankararao, B. and Gupta, S. K. (2007a). Multi-objective optimization of an industrial fluidized catalytic cracking unit (FCCU) using two jumping gene adaptations of simulated annealing, *Comput. Chem. Eng.*, 31, pp. 1496-1515.

Sankararao, B. and Gupta, S. K. (2007b). Multi-objective optimization of pressure swing adsorbers (PSAs) for air separation, *Ind. Eng. Chem. Res.*, 46, pp. 3751-3765.

Simoes A. B. and Costa, E. (1999a). Transposition *vs*. crossover: an empirical study, *Proc. of GECCO-99*, Morgan Kaufmann, Orlando, Florida, pp. 612-619.

Simoes A. B. and Costa, E. (1999b). Transposition: a biologically inspired mechanism to use with genetic algorithm, *Proc. of the 4th ICANNGA99*, Springer Verlag, Portorez, Slovenia, pp. 178-186.

Suppapitnarm, A., Seffen, K. A., Parks, G. T. and Clarkson, P. J. (2000). A simulated annealing algorithm for multiobjective optimization, *Eng. Opt.*, 33, pp. 59-85.

Stryer, L. (2000). *Biochemistry* (4th ed.), W. H. Freeman, New York, USA.

Yao, X., Liu, Y. and Lin, G. (1999). Evolutionary programming made faster, *IEEE Trans. Evol. Comput.*, 3, pp. 82-102.

Exercises

1. Solve the ZDT4 Problem 1 using NSGA-II-aJG and MOSA-aJG (the two codes in the CD give the solutions).
2. Modify the codes in the CD to solve the ZDT2 Problem 2 and the ZDT3 Problem 3, using NSGA-II-aJG and MOSA-aJG.

This page intentionally left blank

Chapter 5

Surrogate Assisted Evolutionary Algorithm for Multi-Objective Optimization

Tapabrata Ray*, Amitay Isaacs and Warren Smith

School of Aerospace, Civil and Mechanical Engineering,
University of New South Wales, Australian Defence Force Academy,
Canberra, ACT 2600, Australia
*t.ray@adfa.edu.au

Abstract

Evolutionary algorithms (EAs) are population based approaches that start with an initial population of candidate solutions and evolve them over a number of generations to finally arrive at a set of desired solutions. Such population based algorithms are particularly attractive for multi-objective optimization (MOO) problems as they can result in a set of non-dominated solutions in a single run. However, they are known to require evaluations of a large number of candidate solutions during the process that often becomes prohibitive for problems involving computationally expensive analyses. Use of multiple processors and cheaper approximations (surrogates or meta-models) in lieu of the actual analyses are attractive means to contain the computational time within affordable limits. A major problem in using surrogates within an evolutionary algorithm lies with its representation accuracy; the problem is far more acute for multi-objective problems where both the proximity to the Pareto front and the diversity of the solutions along the non-dominated front are required. In this chapter, a surrogate assisted evolutionary algorithm (SAEA) for multi-objective optimization is presented.

A Radial Basis Function (RBF) network is used as a surrogate model. The algorithm performs actual evaluations of objectives and constraints for all the members of the initial population and periodically evaluates all the members of the population after every S generations. An external archive of the unique solutions evaluated using actual analysis is maintained to train the RBF model which is then used in lieu of the actual analysis for the next S generations. In order to ensure the prediction accuracy of the RBF surrogate model, a candidate solution is only approximated if at least one candidate solution in the archive exists in the vicinity (based on a user defined distance threshold) and the accuracy of the surrogate is within an user defined limit. Five multi-objective test problems are presented in this study and a comparison with Nondominated Sorting Genetic Algorithm II (NSGA-II) [Deb *et al.* (2002a)] is included to highlight the benefits offered by the approach. SAEA algorithm consistently reported better non-dominated solutions for all the test cases for the same number of actual evaluations of candidate solutions.

Keywords: multi-objective optimization, surrogate models, radial basis function network, evolutionary algorithm

5.1 Introduction

A multi-objective constrained minimization problem is represented as in Eq. (5.1).

$$\begin{aligned} &\text{Minimize } \ f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}) \\ &\text{Subject to } \ g_i(\mathbf{x}) \geq 0, \ 1 \leq i \leq p \end{aligned} \tag{5.1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the n-dimensional design vector bounded by design space $\mathcal{S} \subset \mathbb{R}^n$. Design space is defined by the lower bound $\underline{x_i}$ and upper bound $\overline{x_i}$ for each variable x_i. There are m objectives to be simultaneously minimized subject to p equality and/or inequality constraints.

Evolutionary algorithms (EAs) have been successfully applied to a range of multi-objective problems. They are particularly suitable for multi-objective problems as they result in a set of non-dominated solutions in a single run. Furthermore, EAs do not rely on functional and slope continuity and thus can be readily applied to optimization problems with mixed variables. However, EAs are essentially population based methods and require evaluation of numerous candidate solutions before converging to the desired set of solutions. Such an approach turns out to be computationally prohibitive for realistic Multidisciplinary Design Optimization problems and

there is a growing interest in the use of surrogates to reduce the number of actual function evaluations.

A comprehensive review on the use of fitness approximation in the context of evolutionary computation has been reported by Jin (2005). The choice of surrogate models reported in the literature range from neural network based models like multilayer perceptrons [Nain and Deb (2002)] and radial basis function networks [Won *et al.* (2003); Farina (2002)], quadratic response surfaces, Kriging [Knowles (2006); Wilson *et al.* (2001); Won and Ray (2005)] and cokriging models [Simpson *et al.* (1998); Won and Ray (2004)]. In terms of identifying data to create the surrogates, there have been proposals ranging from a random sampling to more sophisticated design of experiments based approaches relying on orthogonal arrays [Knowles (2006); Wilson *et al.* (2001)]. Novel elite preservation strategies within a surrogate framework have also been reported for single objective optimization problems [Won and Ray (2005)]. In terms of applications, there are numerous reports on the use of variable fidelity models for airfoil shape optimization, aircraft design, *etc.* Although, there are many reports on the use of surrogates for a single objective optimization and their performance on mathematical benchmarks, there are only a few in the area of multi-objective optimization and even fewer that deal with problems with more than 10 variables.

One of the earlier attempts in this area is by Wilson *et al.* (2001) where a Kriging based surrogate was generated from a Latin Hypercube sampling. This surrogate was used throughout the course of optimization. Algorithms based on such an one-shot approximation approach are likely to face problems when the initial set of solutions generated differ substantially from the final set [Schaffer (1985)]. Periodic retraining of the surrogate models is necessary as the search proceeds to localized areas. Farina (2002) reported the use of a RBF network within an evolutionary algorithm. The RBF network was created using all data points that had been computed using actual function evaluations. It is important to note that such an approach will incur increasing computational cost to train the neural network over generations. Nain and Deb (2002) proposed a multi-fidelity model (coarse to fine grain) for surrogate assisted multi-objective optimization where a multilayer perceptron was periodically retrained and used in alternation with actual computations to solve a B-spline curve fitting problem. A similar approach of alternating between the actual evaluations and the surrogate models predictions have also been reported by Ray and Smith (2006). The study used a RBF model that was trained using the candidate

solutions of the population after every K generations. Nain and Deb (2005) reported the performance of their model on two test functions namely the ZDT4 and TNK test functions. The Pareto Efficient Global Optimization (ParEGO) has been reported recently by Knowles (2006). The algorithm relies on a Kriging based surrogate and the sampling points are generated via design of experiments. However, the method requires knowledge about the limits of the objective function space and cannot guarantee a uniform distribution of the solutions along the non-dominated front. Prior information about the limits of the objective functions may not always be available. Another recent paper by Chafekar *et al.* (2005) reports the use of multiple GAs, each of which uses a reduced model of an objective with regular information exchange among them to obtain a well distributed non-dominated set of solutions. Other reports of the use of surrogates for multi-objective optimization appear in the papers by Voutchkov and Keane (2006) and Santana-Quinter *et al.* (2007).

In this chapter, a surrogate assisted evolutionary algorithm (SAEA) that eliminates some of the problems discussed above is proposed. Its performance on a number of mathematical benchmarks is reported and compared with the results of NSGA-II. The features of the algorithm are discussed in Sec. 5.2 and the results are presented in Sec. 5.3. Summarized in Sec. 5.4 are the findings and some of the ongoing developments.

5.2 Surrogate Assisted Evolutionary Algorithm

The proposed Surrogate Assisted Evolutionary Algorithm is outlined in Algorithm 5.1. The MATLAB code of the algorithm is available in the folder "Chapter 5" on the CD.

The basic evolutionary algorithm is the same as the NSGA-II by Deb *et al.* (2002b). The algorithm starts with a random initial population and uses actual evaluations of the objective and the constraint functions. Surrogate models are then created for all the objectives and the constraints. The algorithm then uses the surrogate models to evaluate the values of the objective and constraint functions for next S generations. The surrogate models are periodically trained after every S generations.

An external archive of actual evaluations is maintained and used to create the surrogate models. The archive only maintains the unique candidate solutions evaluated using actual evaluations over generations.

A surrogate model is created for each of the objectives and the constraints using a fraction ($0 < \alpha < 1$) of the candidate solutions in the

Algorithm 5.1 Surrogate Assisted Evolutionary Algorithm

Require: $N_G > 1$ {Number of Generations}
Require: $M > 0$ {Population size}
Require: $I_{TRAIN} > 0$ {Periodic Surrogate Training Interval}
$\mathcal{A} = \emptyset$ {Archive of Actual Evaluations}
P_1 = Initialize()
Evaluate(P_1)
$\mathcal{A}$ = AddToArchive($\mathcal{A}, P_1$)
$\mathcal{S}$ = BuildSurrogate($\mathcal{A}$)
for $i = 2$ to N_G **do**
  Rank(P_{i-1})
  repeat
    $p1, p2$ = Select(P_{i-1})
    $c1, c2$ = Crossover($p1, p2$)
    Mutate($c1$)
    Mutate($c2$)
    Add $c1, c2$ to C_{i-1}
  until C_{i-1} has M children
  if modulo$(i-1, I_{TRAIN}) == 0$ **then**
    $\mathcal{A}$ = AddToArchive($\mathcal{A}, P_i$)
    $\mathcal{S}$ = BuildSurrogate($\mathcal{A}$)
  end if
  Evaluate(C_{i-1}, $\mathcal{S}$)
  P_i = Reduce($P_{i-1} + C_{i-1}$)
end for
Evaluate(P_{N_G}) {Final population re-evaluated}

archive. These solutions are identified by the k-Means clustering algorithm. Such an approach is in accordance with the suggestions by Haykin (1999) and Jin (2005) that using the entire archive could lead to over-fitting and the introduction of false optima.

5.2.1 *Initialization*

All the solutions in a population are initialized by selecting individual variable values as given in Eq. (5.2).

$$x_i = \underline{x_i} + \mathcal{U}[0,1]\ (\overline{x_i} - \underline{x_i}) \qquad 1 \leq i \leq n \tag{5.2}$$

where x_i denotes the initialized variable, and $\mathcal{U}[0,1]$ is an uniform random number lying between 0 and 1.

5.2.2 *Actual Solution Archive*

All the unique candidate solutions are maintained in an external archive. These solutions are used to train the surrogate models periodically after every S generations. Each entry in the archive comprises the candidate design ($\mathbf{x}$) normalized in the range $[0,1]$, objectives $(f_1,\ldots,f_m)$ and constraints $(g_1,\ldots,g_p)$ evaluated at $\mathbf{x}$. To avoid numerical difficulties in fitting surrogate models, a new candidate solution is added to the archive only if none of the solutions in the archive lie in the close neighborhood of that solution (computed using the Euclidean distance). The neighborhood radius of $1.e-6$ is used for this study.

5.2.3 *Selection*

The procedure for selection of parents is the same as that of NSGA-II. Binary tournament is used to select a parent from two individuals. Binary tournament between two candidate solutions $\mathbf{x}^1$ and $\mathbf{x}^2$ is performed as follows:

(1) If $\mathbf{x}^1$ is feasible and $\mathbf{x}^2$ is infeasible: $\mathbf{x}^1$ is selected and vice versa.
(2) If both $\mathbf{x}^1$ and $\mathbf{x}^2$ are infeasible: the one for which the value of the maximum violated constraint is smaller is selected.
(3) If both $\mathbf{x}^1$ and $\mathbf{x}^2$ are feasible and $\mathbf{x}^1$ dominates $\mathbf{x}^2$: $\mathbf{x}^1$ is selected and vice versa.
(4) If both $\mathbf{x}^1$ and $\mathbf{x}^2$ are feasible and neither dominate the other: one of $\mathbf{x}^1$ and $\mathbf{x}^2$ is selected randomly.

Four solutions are chosen at random from the population. From the first two solutions, parent 1 is selected and from the last two individuals, parent 2 is selected using binary tournament. To ensure that all the solutions in the population take part in the selection process, a shuffled list of IDs (1 to M) is created and individuals are picked up 4 at a time from this list.

5.2.4 *Crossover and Mutation*

Simulated Binary Crossover (SBX) [Deb and Agrawal (1995)] operator for real valued variables is used to create two children from a pair of parents.

The offsprings $\mathbf{y}^1$ and $\mathbf{y}^2$ are created from parents $\mathbf{x}^1$ and $\mathbf{x}^2$ by operating on one variable at a time as shown in Eq. (5.3),

$$\begin{aligned} y_i^1 &= 0.5\left[(1+\beta_{q_i})\, x_i^1 + (1-\beta_{q_i})\, x_i^2\right] \\ y_i^2 &= 0.5\left[(1-\beta_{q_i})\, x_i^1 + (1+\beta_{q_i})\, x_i^2\right] \end{aligned} \tag{5.3}$$

where β_{q_i} is calculated as,

$$\beta_{q_i} = \begin{cases} (2u_i)^{1/\eta_c+1}, & \text{if } u_i \leq 0.5, \\ \left(\frac{1}{2(1-u_i)}\right)^{1/\eta_c+1} & \text{if } u_i > 0.5. \end{cases} \tag{5.4}$$

and where u_i is the uniform random number in the range $[0, 1)$ and η_c is the user defined parameter *Distribution Index for Crossover*.

The Mutation operator used is the polynomial mutation operator defined by Deb and Goyal (1996). Each variable of $\mathbf{y}$ is obtained from a corresponding variable of $\mathbf{x}$ as given in Eq. (5.5).

$$y_i = x_i + (\overline{x_i} - \underline{x_i})\, \bar{\delta}_i \tag{5.5}$$

where $\bar{\delta}_i$ is calculated as,

$$\bar{\delta}_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5, \\ 1 - [2(1-r_i)]^{1/(\eta_m+1)}, & \text{if } r_i \geq 0.5. \end{cases} \tag{5.6}$$

and where r_i is the uniform random number in the range $[0, 1)$ and η_m is the user defined parameter *Distribution Index for Mutation*.

If the value of any of the variables calculated using crossover and mutation operators falls below the lower bound (or above the upper bound), the value of that variable is fixed at the lower bound (or the upper bound).

5.2.5 *Ranking*

Ranking of the candidate solutions involves ranking the feasible and the infeasible solutions separately. The feasible solutions are first sorted using non-dominated sorting to generate fronts of non-dominated solutions. Solutions within each front are ranked based on the decreasing value of the crowding distance. The infeasible solutions are ranked based on increasing order of the maximum violated constraint value.

5.2.6 *Reduction*

The reduction procedure for retaining M solutions for the next generation from a set of $2M$ solutions (parent and offspring population) uses the ranks obtained by the ranking procedure.

(1) If the feasible solutions number more than M,

- the M solutions are selected in the order of non-dominated fronts and decreasing crowding distance in each front.

(2) If the feasible solutions are less than or equal to M,

- all the feasible solutions are selected in the order of non-dominated fronts and decreasing crowding distance in each front, and
- all the remaining solutions are selected from the infeasible solutions in the order of increasing value of the maximum constraint violation.

5.2.7 *Building Surrogates*

Outlined in Algorithm 5.2 are the steps involved in building surrogates for all the objectives and the constraints evaluated using the actual analysis of the solutions in the archive. A fraction (α) of solutions in the archive is used to build the surrogates to prevent over-fitting. These solutions are identified by the k-Means clustering algorithm where design variables $x_1, \ldots, x_n$ are used as the clustering attributes. To restrict the time required to train the RBF model, an upper bound is put on the maximum number of solutions (K_{max}) used to train the surrogate models. A subset ($\mathcal{A}_s$) of the archive is created by selecting the solutions closest to the centroids of the K clusters obtained by the k-Means algorithm. The subset of the archive is then used to build surrogates for each of the objectives and the constraints.

5.2.7.1 *Radial Basis Functions*

Radial Basis functions (RBF) belong to the class of Artificial Neural Networks (ANNs) and are a popular choice for approximating nonlinear functions. RBF ϕ has its output symmetric around an associated centre μ.

$$\phi(\mathbf{x}) = \phi(\|\mathbf{x} - \mu\|) \tag{5.7}$$

where the argument of ϕ is a vector norm. A common RBF is the Gaussian function with the Euclidean norm.

$$\phi(r) = e^{-r^2/\sigma^2} \tag{5.8}$$

where σ is the scale or width parameter.

A set of RBFs can serve as a basis for representing a wide class of functions that are expressible as linear combinations of the chosen RBFs

Algorithm 5.2 Building of Surrogates

Require: $\mathcal{A}$ {Archive of actual evaluations}
Require: K_{max} {Maximum solutions used in creating surrogates}
Require: $0 < \alpha < 1$ {Fraction of archive solutions used}
Require: $m > 0, p \geq 0$ {Number of objectives and constraints}
1: $K = \alpha\ |\mathcal{A}|$
2: **if** $K > K_{max}$ **then**
3: $\quad K = K_{max}$
4: **end if**
5: $\mathcal{A}_s = \text{KMeans}(\mathcal{A}, K)$
6: **for** $i = 1$ to m **do**
7: $\quad \mathcal{S}_{f_i} = \text{Surrogate}(\mathcal{A}_s)$
8: **end for**
9: **for** $i = 1$ to p **do**
10: $\quad \mathcal{S}_{g_i} = \text{Surrogate}(\mathcal{A}_s)$
11: **end for**

as shown in Eq. (5.9).

$$y(\mathbf{x}) = \sum_{i=1}^{m} w_i\ \phi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{5.9}$$

However, Eq. (5.9) is usually very expensive to implement if the number of the data points is large. Thus a generalized RBF network is usually adopted of the following form:

$$y(\mathbf{x}) = \sum_{i=1}^{k} w_i\ \phi(\|\mathbf{x} - \mu_i\|) \tag{5.10}$$

where μ_i are the k centres (identified by the k-Means algorithm). Here, k is typically smaller than m. The coefficients w_i are the unknown parameters that are to be "learned." The training is usually achieved via the least squares solution:

$$\mathbf{w} = \mathbf{A}^{+}\ \mathbf{y} \tag{5.11}$$

where $\mathbf{A}^{+}$ is the pseudo-inverse and $\mathbf{y}$ is the target output vector. Most often $\mathbf{A}$ is a not a square matrix, hence no inverse exists. Calculation of the pseudo-inverse is computationally expensive for large problems and thus the recursive least-squares estimation is often used.

5.2.8 *Evaluation using Surrogates*

The RBF models are used to compute the values of the objectives and the constraints of new candidate solutions in lieu of the actual analysis for S generations. A fraction of solutions from the archive are used in surrogate building and the rest of the solutions in the archive are used to evaluate the quality of the RBF surrogates. Mean squared error (MSE) in the actual and the predicted values of the objectives and the constraints (normalized by the actual values) at the remaining solutions in the archive is used as the measure to validate the surrogate models. If the MSE for a surrogate model is less than the user specified threshold, then the surrogate model is said to be valid. Only if the surrogate models for all the objectives and the constraints are valid, they are used for prediction at the new candidate solutions, otherwise the actual analysis is used instead. For this study a threshold value of 5 is used as the MSE criterion for each surrogate model. Even if the surrogate model is valid, candidate solutions may still be computed using actual analysis if their distances to the closest point in the archive are more than 5% of the solid diagonal of the design space (*i.e.* the new solutions are far-off compared to the explored region of the design space).

5.2.9 *k-Means Clustering Algorithm*

The k-Means clustering algorithm is used to identify k solutions that are used to create the surrogate models. From the possible m solutions in the archive, k centres $\mu_1, \ldots, \mu_k$ are obtained.

The k-Means Algorithm as outlined in Algorithm 5.3 starts with the first k data points as the k centres. Then the membership function (ψ) is calculated for each of the solutions. For each solution, the closest centre is obtained using the Euclidean distance measure, and that solution is assigned membership to the closest centre. A constant weight function is used to assign equal importance to all the solutions. Then the location of each of k centres is recomputed from all the solutions according to their memberships and weights. Assignment of memberships and recalculation of centres are repeated till the membership function is unchanged.

Algorithm 5.3 k-Means Clustering Algorithm

Require: $\{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$ $\{m$ data sets$\}$
Require: $k > 0$
$C = \{\mu_1, \ldots, \mu_k\} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$
repeat
  for $i = 1$ to m **do**
    $\psi(\mu_j|\mathbf{x}^i) = 0, \quad j = 1, \ldots, k$
    $l = \arg\min_j \|x_i - \mu_j\|^2$
    $\psi(\mu_l|\mathbf{x}^i) = 1$
    $w(\mathbf{x}^i) = 1$
  end for
  for $i = 1$ to k **do**
    $A = \sum_{j=1}^{m} \psi(\mu_i|\mathbf{x}^j)\, w(x_j)\, \mathbf{x}^j$
    $B = \sum_{j=1}^{m} \psi(\mu_i|\mathbf{x}^j)\, w(x_j)$
    $\mu_i = A/B$
  end for
until ψ is constant

5.3 Numerical Examples

In the following numerical examples an S value of 5 is used meaning the algorithm uses actual evaluations and creates surrogate models on a 5 generation cycle and for the intermediate generations uses the surrogate models. The probability of crossover is set to 0.9 and the probability of mutation is set to 0.1. The distribution index for crossover is 10 and distribution index for mutation is 20. To build surrogates, 80% of the solutions in the archive are used ($\alpha = 0.8$) with the maximum limit of 500 solutions ($K_{max} = 500$). A user limit of 5 is used as the MSE criterion to validate individual surrogate model.

All the numerical examples are solved using the proposed algorithm SAEA and NSGA-II. The SAEA uses the same random generator as that of NSGA-II. Same random seed is chosen for both NSGA-II and SAEA ensuring that the initial population is same for both the algorithms. As the performance of SAEA is consistently better than that of NSGA-II for the test problems using the same number of function evaluations across different parameter values and random seeds, representative results of a single run are presented.

5.3.1 *Zitzler-Deb-Thiele's (ZDT) Test Problems*

The ZDT test problems are two objective problems framed by Zitzler *et al.* [Deb (2001)]. ZDT test problems are unconstrained problems.

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}), \\ & f_2(\mathbf{x}) = g(\mathbf{x})\, h(f_1(\mathbf{x}), g(\mathbf{x})). \end{aligned}$$

5.3.1.1 *ZDT1*

Problem ZDT1 has a convex Pareto optimal front and is defined as shown in Eq. (5.12).

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ g(\mathbf{x}) &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i, \\ h(f_1, g) &= 1 - \sqrt{f_1/g}. \end{aligned} \tag{5.12}$$

where $n = 10$ and $x_i \in [0, 1]$. The Pareto region corresponds to $0 \leq x_1^* \leq 1$ and $x_i^* = 0$ for $i = 2, 3, \ldots, 10$.

A population size of 100 has been used to solve ZDT1 problem and the population is allowed to evolve over 101 generations. The results of the algorithm with and without surrogate assistance are presented in Fig. 5.1. It is clear from Fig. 5.1 that with surrogate assistance, the algorithm could generate a better set of non-dominated solutions that are very close to the actual Pareto front and are well distributed along the Pareto front. SAEA performed 2,882 actual evaluations and 7,218 approximations.

If NSGA-II is allowed to run for sufficiently large number of function evaluations, it can capture the Pareto front. As seen from Fig. 5.2, NSGA-II with 10,000 function evaluations is able to distribute solutions along the Pareto front. SAEA achieves a saving of more than 70% in the number of function evaluations (2,882 vs. 10,000) as compared to NSGA-II to capture the Pareto front. The search space for ZDT1 is 10-D and requires function evaluations of the order of 10,000 to achieve close proximity to the Pareto front and good spread of non-dominated solutions along the Pareto front.

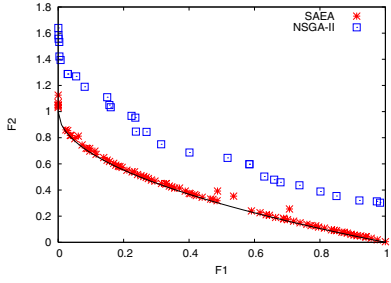


Fig. 5.1 Results for ZDT1 obtained by SAEA and NSGA-II using limited number of function evaluations

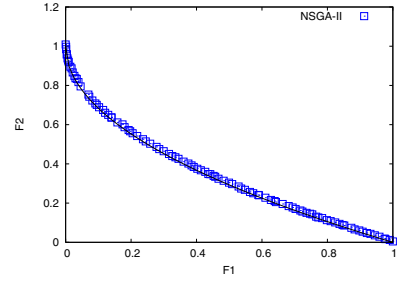


Fig. 5.2 Results of ZDT1 obtained by NSGA-II using 10,000 function evaluations

5.3.1.2 *ZDT2*

Problem ZDT2 has a concave Pareto optimal front. The problem is presented in Eq. (5.13).

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ g(\mathbf{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h(f_1, g) &= 1 - (f_1/g)^2. \end{aligned} \tag{5.13}$$

where $n = 10$ and $x_i \in [0, 1]$. The Pareto region corresponds to $0 \leq x_1^* \leq 1$ and $x_i^* = 0$ for $i = 2, 3, \ldots, 10$.

We have used a population size of 100 and allowed it to evolve over 101 generations. The results of the algorithm with and without surrogate assistance are presented in Fig. 5.3. It can be seen from Fig. 5.3 that for same number of function evaluations, SAEA is able to get a good spread of solutions along the Pareto front. The surrogate model performed 3,056 actual evaluations and 7,044 approximations. NSGA-II results are obtained after 3,100 actual evaluations.

5.3.1.3 *ZDT3*

Problem ZDT3 has a number of disconnected Pareto fronts. The problem is presented in Eq. (5.14).

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ g(\mathbf{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1). \end{aligned} \tag{5.14}$$

where $n = 10$ and $x_i \in [0, 1]$. The Pareto region corresponds to $x_i^* = 0$ for $i = 2, 3, \ldots, 10$.

We have used a population size of 100 and allowed it to evolve over 101 generations. The results of the algorithm with and without surrogate assistance are presented in Fig. 5.4. The Pareto fronts for ZDT3 are distributed along the boundary of the search space as seen in Fig. 5.4. Only the lower portions of the troughs contribute to the Pareto fronts. The surrogate assisted algorithm performed 7,094 actual evaluations and 3,006 approximations while NSGA-II performed 7,100 function evaluations. It is seen from Fig. 5.4 that the non-dominated solution obtained by both SAEA and NSGA-II are close to the actual Pareto front.

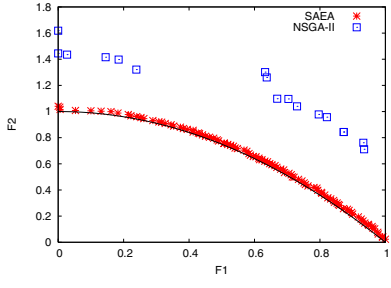


Fig. 5.3 Results of ZDT2 obtained by SAEA and NSGA-II using limited number of function evaluations

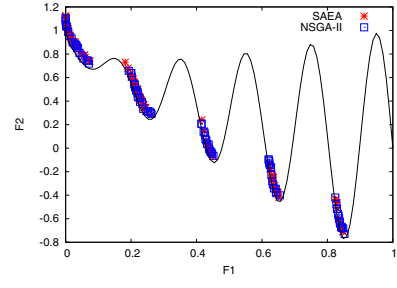


Fig. 5.4 Results of ZDT3 obtained by SAEA and NSGA-II using limited number of function evaluations

5.3.1.4 *Performance Comparison*

The performance of any multi-objective optimization algorithm can be qualitatively deduced from looking at the non-dominated front. To compare two algorithms one needs a quantitative measure. One such measure is Inverted Generational Distance (IGD) introduced by Veldhuizen and Lamont (2000). IGD is defined by

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{5.15}$$

where n is the number of non-dominated solutions obtained and d_i is the Euclidean distance between each of the solutions and the closest solution on the true Pareto front. IGD metric is the measure of the separation between the Pareto front and the non-dominated solutions obtained. If all the non-dominated solutions obtained lie on the true Pareto front $IGD = 0$.

Inverted Generational Distances for problems ZDT1, ZDT2 and ZDT3 are given in Table 5.1 for a better understanding. The true Pareto sets for ZDT test problems are represented by 100 solutions on the Pareto front. It is seen that IGD for the non-dominated solutions obtained by SAEA is smaller than that of NSGA-II. Other metrics can also be used to determine and compare results between different optimization algorithms.

Table 5.1 Inverted Generational Distance for problems ZDT1, ZDT2, and ZDT3

	ZDT1	ZDT2	ZDT3
NSGA-II	0.5027	0.6331	0.2898
SAEA	0.2558	0.2413	0.2734

5.3.2 *Osyczka and Kundu (OSY) Test Problem*

The OSY problem is a six-variable problem with two-objectives and six inequality constraints. The problem is presented in Eq. (5.16).

$$\begin{aligned} f_1(\mathbf{x}) =& -\left[25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2\right. \\ &\left.+(x_4-4)^2+(x_5-1)^2\right], \\ f_2(\mathbf{x}) =& x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_6^2, \\ g_1(\mathbf{x}) =& x_1+x_2-2\geq 0, \\ g_2(\mathbf{x}) =& 6-x_1-x_2\geq 0, \\ g_3(\mathbf{x}) =& 2-x_2+x_1\geq 0, \\ g_4(\mathbf{x}) =& 2-x_1+3x_2\geq 0, \\ g_5(\mathbf{x}) =& 4-(x_3-3)^2-x_4\geq 0, \\ g_6(\mathbf{x}) =& (x_5-3)^2+x_6-4\geq 0, \end{aligned} \tag{5.16}$$

where $0\leq x_1,x_2,x_6\leq 10$, $1\leq x_3,x_5\leq 5$, $0\leq x_4\leq 6$.

For problem OSY, a population of size 100 is evolved over 101 generations. The results of the algorithm with and without surrogate assistance are presented in Fig. 5.5. The surrogate model performed 4,816 actual evaluations and 5,284 approximations. NSGA-II was run for 4,900 actual evaluations. Although both NSGA-II and SAEA obtain solutions near the Pareto front, the non-dominated solutions obtained by SAEA have a better spread as compared that of NSGA-II.

5.3.3 *Tanaka (TNK) Test Problem*

The TNK problem is a two-variable problem with two-objectives and two inequality constraints. The problem is presented in Eq. (5.17).

$$\begin{aligned} f_1(\mathbf{x}) =& x_1, \\ f_2(\mathbf{x}) =& x_2, \\ g_1(\mathbf{x}) =& x_1^2 + x_2^2 - 1 - 0.1\cos(16\arctan\frac{x1}{x2}) \geq 0, \\ g_2(\mathbf{x}) =& 0.5 - (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \geq 0, \end{aligned} \tag{5.17}$$

where $0 \leq x_1, x_2 \leq \pi$.

For problem TNK, a population of 100 is allowed to evolve over 101 generations. The results of the algorithm with and without surrogate assistance are presented in Fig. 5.6. The surrogate model performed 1,905 actual evaluations and 8,195 approximations. NSGA-II was run for 2,000 actual evaluations. As seen in Fig. 5.6 the spread of the solutions on the Pareto optimal front is much better using surrogate assisted algorithm than that of NSGA-II with fewer function evaluations.

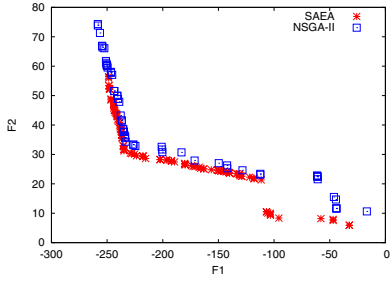


Fig. 5.5 Results of OSY obtained by SAEA and NSGA-II using limited number of function evaluations

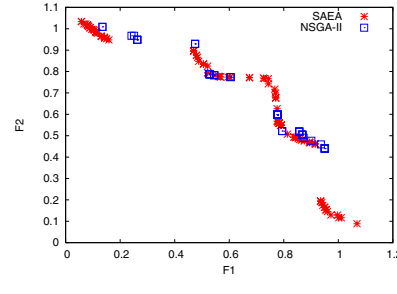


Fig. 5.6 Results of TNK obtained by SAEA and NSGA-II using the same number of function evaluations

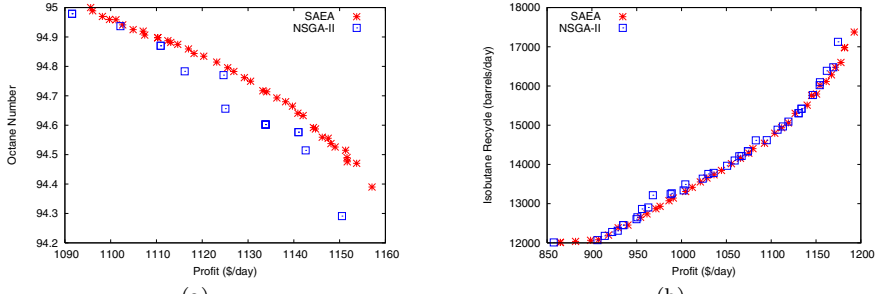
5.3.4 *Alkylation Process Optimization*

Alkylation process has been described in Sec. 1.4. Two optimization problems referred in Chap. 1 as Case A (Maximize Profit and Maximize Octane Number) and Case B (Maximize Profit and Minimize Isobutane Recycle) are solved using NSGA-II and SAEA to illustrate the benefits of SAEA. Variable bounds for optimization problem Case A is the same as those

listed in Table 1.2, while for Case B the variable bound for x_2 was set to lie between 12,000 and 17,500 barrels/day.

For both optimization problems a population of size 40 is evolved over 31 generations. The results of Case A are shown in Fig. 5.7(a). SAEA performed 426 actual evaluations and NSGA-II was run with 440 evaluations. The non-dominated solutions obtained by SAEA have a much better spread as compared to the non-dominated solutions obtained by NSGA-II.

The results of the alkylation optimization process Case B are shown in Fig. 5.7(b). SAEA performed 403 actual evaluations and NSGA-II was run with 440 evaluations. It is seen from the overlapping non-dominated solutions that SAEA performance is on par with the NSGA-II.



(a) (b)

Fig. 5.7 Results obtained by SAEA and NSGA-II for alkylation process optimization. (a) Case A. (b) Case B.

5.4 Conclusion

In this chapter, an evolutionary algorithm for multi-objective optimization that is embedded with a surrogate model to reduce the computational cost has been introduced. The surrogate assisted evolutionary algorithm (SAEA) alternates in cycles *i.e.* SAEA performs actual evaluations once every S generations, trains the surrogate models and employs predictions of the surrogate models in lieu of actual evaluations for the intermediate generations.

In order to maximize the use of information from all actual evaluations, the algorithm maintains an external archive that is used to train the RBF model, periodically after every S generations. In order to maintain prediction accuracy, a candidate solution is only approximated using the

RBF model if at least one solution exists in the archive that is within a distance threshold. This distance threshold plays an important role in the early stages of evolution where more candidate solutions are evaluated using actual computations even during the surrogate phase. Furthermore, a candidate solution is only evaluated using the surrogate model if the MSE of the surrogate on the validation set is below an user defined threshold.

The results of all the test problems support the fact that better non-dominated solutions can be delivered by the SAEA as compared to NSGA-II for the same number of actual function evaluations. Although the algorithm incurs additional computational cost for solution clustering and periodic training of RBF models, such cost is insignificant for problems where the evaluation of a single candidate solution requires expensive analyses like finite element methods or computational fluid dynamics.

In lieu of the RBF model used in this study, other surrogate models such as multilayer perceptron or Kriging could be used. In order to evade the problem of selection of surrogate models *a priori*, the authors are investigating the performance of surrogate ensembles. The surrogate assisted optimization models are being used by the authors for real-life problems, *e.g.* scram-jet design, nose cone design and structural optimization problems requiring computationally expensive analyses.

References

Chafekar, D., Shi, L., Rasheed, K. and Xuan, J. (2005). Multiobjective GA optimization using reduced models, *IEEE Transactions on Systems, Man and Cybernetics, Part C* **35**, 2, pp. 261–265.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms* (John Wiley and Sons Pvt. Ltd.).

Deb, K. and Agrawal, S. (1995). Simulated binary crossover for continuous search space, *Complex Systems* **9**, 2, pp. 115–148.

Deb, K. and Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design, *Computer Science and Informatics* **26**, 4, pp. 30–45.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002a). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* **6**, 2, pp. 182–197.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002b). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* **6**, 2, pp. 182–197.

Farina, M. (2002). A neural network based generalized response surface multiobjective evolutionary algorithm, in *Proceedings of IEEE World Congress on Computational Intelligence (WCCI-2002)* (Hawaii).

Haykin, S. (1999). *Neural Networks: A Comprehensive foundation*, 2nd edn. (Prentice-Hall Internal Inc., New Jersey).

Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation, *Soft Computing - A Fusion of Foundations, Methodologies and Applications* **9**, 1, pp. 3–12.

Knowles, J. (2006). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, *IEEE Transactions on Evolutionary Computation* **10**, 1, pp. 50–66.

Nain, P. K. S. and Deb, K. (2002). A computationally effective multiobjective search optimization technique using course to fine grain modeing, Tech. Rep. 2002005, Kangal, IIT Kanpur, India.

Nain, P. K. S. and Deb, K. (2005). A multiobjective optimization procedure with successive approximate models, Tech. Rep. 2005002, Kangal, IIT Kanpur, India.

Ray, T. and Smith, W. (2006). Surrogate assisted evolutionary algorithm for multiobjective optimization, *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* , pp. 1–8.

Santana-Quinter, L. V., Serrano-Hernandez, V. A., Coello Coello, C. A., Hernandez-Diaz, A. G. and Molina, J. (2007). Use of radial basis functions and rough sets for evolutionary multi-objective optimization, in *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM'2007)* (IEEE Press, Honolulu, Hawaii, USA), pp. 107–114.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms, in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 93–100.

Simpson, T. W., Korte, J. J., Mauery, T. M. and Mistree, F. (1998). Comparison of response surface and kriging models for multidisciplinary design optimization, in *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, AIAA-98-4755 (St. Louis, MO).

Veldhuizen, D. V. and Lamont, G. (2000). On measuring multiobjective evolutionary algorithm performance, in *Proceedings of the Congress on Evolutionary Computation (CEC 2000)*, Vol. 1, pp. 204–211.

Voutchkov, I. and Keane, A. J. (2006). Multiobjective optimization using surrogates, in I. C. Parmee (ed.), *Adaptive Computing in Design and Manufacture 2006. Proceedings of the Seventh International Conference* (The Institute for People-centred Computation, Bristol, UK), pp. 167–175.

Wilson, B., Cappelleri, D., Simpson, T. W. and Frecker, M. (2001). Efficient Pareto frontier exploration using surrogate approximations, *Optimization and Engineering* **2**, 1, pp. 31–50.

Won, K. S. and Ray, T. (2004). Performance of kriging and cokriging based surrogate models within the unified framework for surrogate assisted optimization, in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC'04* (Portland).

Won, K. S. and Ray, T. (2005). A framework for design optimization using surrogates, *Engineering Optimization* **37**, 7, pp. 685–703.

Won, K. S., Ray, T. and Tai, K. (2003). A framework for optimization using approximate functions, in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'03)* (Canberra).

Exercises

(1) For the following optimization problems find the set of non-dominated solutions using NSGA-II and SAEA.

(a) Problem BNH

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}) = 4x_1^2 + 4x_2^2, \\ \text{Minimize} \quad & f_2(\mathbf{x}) = (x_1 - 5)^2 + (x_2 - 5)^2, \\ \text{subject to} \quad & (x_1 - 5)^2 + x_2^2 \leq 25, \\ & (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7, \\ & 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3. \end{aligned}$$

(b) Problem SRN

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2, \\ \text{Minimize} \quad & f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2, \\ \text{subject to} \quad & x_1^2 + x_2^2 \leq 225, \\ & x_1 - 3x_2 + 10 \leq 0, \\ & -20 \leq x_1, x_2 \leq 20. \end{aligned}$$

(c) Problem ZDT4

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}), \\ \text{Minimize} \quad & f_2(\mathbf{x}) = g(\mathbf{x})\, h(f_1(\mathbf{x}), g(\mathbf{x})), \\ \text{where} \quad & f_1(\mathbf{x}) = x_1, \\ & g(\mathbf{x}) = 1 + 10(10 - 1) + \sum_{i=2}^{10} (x_i^2 - 10\cos(4\pi x_i)), \\ & h(f_1, g) = 1 - \sqrt{f_1/g}, \\ & 0 \leq x_1, x_2, \ldots, x_{10} \leq 1. \end{aligned}$$

(d) Problem SCH1

$$\begin{aligned} \text{Minimize} \quad & f_1(x) = x^2, \\ \text{Maximize} \quad & f_2(x) = (x - 2)^2, \\ & -100 \leq x \leq 100. \end{aligned}$$

(2) Observe the performance of NSGA-II on the above problems with population size varying between 100 and 200, number of generations between 100 and 200, probability of crossover between 0.8 and 1.0, probability of mutation between 0.01 and 0.1, distribution index of crossover between 5 and 20, distribution index of mutation between 10 and 50.
(3) Observe the performance of SAEA on the above problems with varying periodicity of training from every 2 to 10 generations and different user specified distance threshold values.

This page intentionally left blank

Chapter 6

Why Use Interactive Multi-Objective Optimization in Chemical Process Design?

Kaisa Miettinen and Jussi Hakanen
Department of Mathematical Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
kaisa.miettinen@jyu.fi, jussi.hakanen@jyu.fi

Abstract
Problems in chemical engineering, like most real-world optimization problems, typically, have several conflicting performance criteria or objectives and they often are computationally demanding, which sets special requirements on the optimization methods used. In this paper, we point out some shortcomings of some widely used basic methods of multi-objective optimization. As an alternative, we suggest using interactive approaches where the role of a decision maker or a designer is emphasized. Interactive multi-objective optimization has been shown to suit well for chemical process design problems because it takes the preferences of the decision maker into account in an iterative manner that enables a focused search for the best Pareto optimal solution, that is, the best compromise between the conflicting objectives. For this reason, only those solutions that are of interest to the decision maker need to be generated making this kind of an approach computationally efficient. Besides, the decision maker does not have to compare many solutions at a time which makes interactive approaches more usable from the cognitive point of view. Furthermore, during the interactive solution process the decision maker can learn about the interrelationships among the objectives. In addition to describing the general philosophy of interactive approaches, we discuss the possibilities of interactive multi-objective optimization in chemical process design and give some examples of interactive methods to illustrate the ideas. Finally, we demonstrate the usefulness of interactive approaches in chemical process design by summarizing some reported studies related to, for example, paper making and sugar industries. Let us emphasize that the approaches described are appropriate for problems with more than two objective functions.

Keywords: Multiple criteria decision making (MCDM), interactive methods, scalarization, chemical engineering, Pareto optimality

6.1. Introduction

Problems involving multiple conflicting criteria or objectives are generally known as multiple criteria decision making (MCDM) problems. In such problems, instead of a well-defined single optimal solution, there are many compromise solutions, so-called Pareto optimal solutions, that are mathematically incomparable. In the MCDM literature, solving a multi-objective optimization problem is usually understood as helping a human decision maker (DM) in considering the multiple objectives simultaneously and in finding a Pareto optimal solution that pleases him/her the most. In other words, the solution process needs some involvement of the DM and the final solution is determined by his/her preferences.

Examples of surveys of methods available for multi-objective optimization are Chankong and Haimes (1983); Hwang and Masud (1979); Marler and Arora (2004); Miettinen (1999); Sawaragi *et al.* (1985); Steuer (1986); Vincke (1992). The methods can be classified in different ways. In Hwang and Masud (1979); Miettinen (1999) they are divided into four classes according to the role of the DM in the solution process. If there is no DM and his/her preference information available, we can use so-called *no-preference methods* which find some neutral compromise solution without any additional preference information. In *a priori methods*, the DM first gives preference information and then the method looks for a Pareto optimal solution satisfying the hopes as well as possible. This is a straightforward approach but the difficulty is that the DM may have too optimistic or pessimistic hopes and then the solution generated may be far from them and, thus, disappointing.

In *a posteriori methods*, a representative set of Pareto optimal solutions is generated and then the DM must select the most preferred one. In this way, the DM gets an overview of the problem but it may be difficult for the DM to analyze a large amount of information. A natural visualization on a plane is possible only for problems involving two objectives. Furthermore, generating the set of Pareto optimal solutions may be computationally expensive. Evolutionary multi-objective optimization (EMO) algorithms belong to this class but it may happen that the solutions generated are not really Pareto optimal but only nondominated in the current population. The fourth class is that of *interactive methods*. Many interactive methods exist but they should become more widely known among people solving real applications. In interactive approaches, a solution pattern is formed and then repeated and the DM can specify and adjust one's preference information between each iteration.

In this chapter, we introduce scalarization based approaches and, in particular, interactive methods as alternatives to EMO approaches. Our aim here is to widen the awareness of the readers of the existence of interactive methods and the advantages and usefulness of using them. Sometimes multi-objective and bi-objective optimization are regarded as synonyms but this kind of thinking is very limiting. Our approaches are capable of handling genuine multi-objective optimization with more than two objectives. Besides discussing drawbacks of some widely used methods (like the weighting method) that sometimes are regarded as the only nonevolutionary multi-objective optimization methods available, we introduce some interactive methods and the NIMBUS method (Miettinen, 1999; Miettinen and Mäkelä, 2006), in particular. In addition, we summarize encouraging experiences of solving some chemical engineering problems with the interactive NIMBUS method. The motivation here is that it is important to get to know that a variety of methods and approaches exists. In this way, people solving different problems are able to use the most appropriate approaches. In this respect, scalarization based and interactive methods complement evolutionary approaches. More information about bringing the MCDM and EMO fields closer is available in Branke *et al.* (2008).

This paper is organized as follows. In Section 6.2, main concepts and the idea of scalarization based methods are introduced. In addition, some basic multi-objective optimization methods and their shortcomings are presented and comparative aspects between scalarization based and evolutionary approaches are discussed. Section 6.3 concentrates on interactive multi-objective optimization and some methods. Advantages of using interactive approaches in chemical process design are discussed in Section 6.4 and some applications related to sugar and papermaking industries are summarized in Section 6.5. Finally, concluding remarks are given in Section 6.6.

6.2. Concepts, Basic Methods and Some Shortcomings

6.2.1. *Concepts*

Let us consider *multi-objective optimization problems* of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S, \end{array} \tag{6.1}$$

where we have k (≥ 2) conflicting *objective functions* $f_i : \mathbf{R}^n \to \mathbf{R}$ that we want to minimize simultaneously. In addition, we have *decision (variable)*

or design vectors $\mathbf{x} = (x_1, \ldots, x_n)^T$ belonging to the nonempty *feasible region* $S \subset \mathbf{R}^n$ defined by equality, inequality and/or box constraints. In multi-objective optimization, we typically are interested in *objective vectors* consisting of *objective (function) values* $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ and the image of the feasible region is called a *feasible objective region* $Z = \mathbf{f}(S)$. (Note that if some function f_i should be maximized, it is equivalent to minimize $-f_i$. Thus, without losing any generality, we consider problems of the form (6.1).)

For multi-objective optimization, theoretical background has been laid, e.g., in Edgeworth (1881); Koopmans (1951); Kuhn and Tucker (1951); Pareto (1896, 1906). Typically, there is no unique optimal solution but a set of mathematically incomparable solutions can be identified. An objective vector can be regarded as optimal if none of its components (i.e., objective values) can be improved without deterioration to at least one of the other objectives. To be more specific, a decision vector $\mathbf{x}' \in S$ and the corresponding objective vector $\mathbf{f}(\mathbf{x}')$ are called *Pareto optimal* if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i = 1, \ldots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ for at least one index j. In the MCDM literature, widely used synonyms of Pareto optimal solutions are nondominated, efficient, noninferior or Edgeworth-Pareto optimal solutions.

As mentioned in the introduction, we here assume that a DM is able to participate in the solution process. (S)he is expected to know the problem domain and be able to specify preference information related to the objectives and/or different solutions. We assume that less is preferred to more in each objective for him/her. (In other words, all the objective functions are to be minimized.) If the problem is correctly formulated, the final solution of a rational DM is always Pareto optimal. Thus, we can restrict our consideration to Pareto optimal solutions. For this reason, it is important that the multi-objective optimization method used is able to find any Pareto optimal solution and produce only Pareto optimal solutions. However, weakly Pareto optimal solutions are sometimes used because they may be easier to generate than Pareto optimal ones. A decision vector $\mathbf{x}' \in S$ (and the corresponding objective vector) is *weakly Pareto optimal* if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}')$ for all $i = 1, \ldots, k$. Note that Pareto optimality implies weak Pareto optimality but not vice versa.

The DM may find information about the ranges of feasible Pareto optimal objective vectors useful. Lower bounds form a so-called *ideal objective vector* $\mathbf{z}^\star \in \mathbf{R}^k$. Its components $z_i^\star$ are obtained by minimizing each ob-

jective function individually subject to the feasible region. Sometimes, for computational reasons, we also need a strictly better *utopian objective vector* $\mathbf{z}^{\star\star}$ defined as $z_i^{\star\star} = z_i^{\star} - \varepsilon$ for $i = 1, \ldots, k$, where ε is some small positive scalar.

The upper bounds of the Pareto optimal set, that is, the components of a *nadir objective vector* $\mathbf{z}^{\mathrm{nad}}$, are in practice difficult to obtain. It can be estimated using a payoff table but the estimate may be unreliable (see, e.g., Miettinen (1999) and references therein).

Finding a final solution to problem (6.1) is called a *solution process*. It usually involves the DM and an analyst. An *analyst* can be a human being or a computer program. The analyst's role is to support the DM and generate information for the DM. Let us emphasize that the DM is not assumed to know multi-objective optimization theory or methods but (s)he is supposed to be an expert in the problem domain, that is, understand the application considered and have insight into the problem. Based on that, (s)he is supposed to be able to specify preference information related to the objectives considered and different solutions. The DM can be, e.g., a *designer*. The task of a multi-objective optimization method is to help the DM in finding the most preferred solution as the final one. The *most preferred solution* is a Pareto optimal solution which is satisfactory for the DM.

Multi-objective optimization problems can be solved by *scalarizing* the problem, in other words, by forming a problem (or several problems) involving a single objective function (and possibly some additional constraints). Because the *scalarized problem* has a real-valued objective function (possibly depending on some parameters originating, e.g., from preference information), it can be solved using appropriate (local, global, mixed-integer etc.) single objective optimizers and, thus, we can utilize the theoretical background and large amount of methods developed for single objective optimization. The real-valued objective function can be called a *scalarizing function*. Such *scalarizing approaches* should be favored that generate Pareto optimal solutions and can find any Pareto optimal solution (as discussed earlier). Depending on whether a local or a global optimizer is used we get either locally or globally Pareto optimal solutions (for nonconvex problems). Because locally Pareto optimal solutions are irrational for DMs, it is important to use appropriate optimizers.

As said in the definition, to move from one Pareto optimal solution to another Pareto optimal solution means trading off. More formally, a *trade-off* is the ratio of change in objective function values involving the

increment of one objective function that occurs when the value of some other objective function decreases. For details, see, e.g., Chankong and Haimes (1983); Miettinen (1999).

The DM can specify preference information in many ways and the task is to find a format that the DM finds most natural and intuitive. One possibility is that the DM specifies *aspiration levels* $\bar{z}_i$ $(i = 1, \ldots, k)$ that are desirable or acceptable objective function values. The vector $\bar{\mathbf{z}} \in \mathbf{R}^k$ consisting of aspiration levels is called a *reference point*.

6.2.2. *Some Basic Methods*

When discussing methods, it is in order to begin with two widely used ones, the weighting method and the ε-constraint method. They can be called basic methods. In many applications one can actually see that they are used without necessarily recognizing them as multi-objective optimization methods or explicitly even saying that the problem considered is a multi-objective optimization one. This means that when formulating and solving the problem, the difference between modeling and optimization phases is not always clear. One can say that these basic methods represent ideas that first come to one's mind when one wants to consider several objective functions simultaneously. However, these methods have some shortcomings that are not necessarily widely known and, for that reason, we want to point them out. In this section, we also briefly discuss some characteristics of EMO approaches when compared to scalarizing approaches. Proofs of theorems related to optimality as well as further details about the methods can be found in Miettinen (1999).

6.2.2.1. *Weighting Method*

The scalarized problem to be solved in the *weighting method* (Gass and Saaty, 1955; Zadeh, 1963) is

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{k} w_i f_i(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{6.2}$$

where the weights are nonnegative, that is, $w_i \geq 0$ for all $i = 1, \ldots, k$ and, typically, $\sum_{i=1}^{k} w_i = 1$. The solution of (6.2) is weakly Pareto optimal and Pareto optimal if $w_i > 0$ for all $i = 1, \ldots, k$ or the solution is unique.

The weighting method can be used (as an *a posteriori* method) so that different weights are set to generate different Pareto optimal solutions and

then the DM must select the most satisfactory one. Alternatively, the DM can be asked to specify the weights reflecting his/her preferences (as an *a priori* method).

It is important to point out that if the problem is nonconvex, the weighting method does not work as it is expected to. However, surprisingly few people applying it seem to realize this. To be more specific, any Pareto optimal solution can be found by altering the weights only if the problem is convex. Thus, it may happen that some Pareto optimal solutions of nonconvex problems remain undiscovered no matter how the weights are varied. This is a serious drawback because it is not always easy to check the convexity in real applications, e.g., involving black box objective functions. If the method is used for generating a representation of the Pareto optimal set, the DM gets a completely misleading impression because only some parts of the Pareto optimal set are covered. Furthermore, as shown by Das and Dennis (1997), an evenly distributed set of weights does not necessarily produce an evenly distributed representation of the Pareto optimal set, even if the problem is convex.

If the method is used as an *a priori* method, the DM is expected to represent his/her preferences in the form of weights. However, in general, the role of the weights may be greatly misleading and it is not at all clear what the concept 'relative importance of an objective' means (Podinovski, 1994; Roy and Mousseau, 1996). Besides, giving weights implies eliciting global preferences which may be hard if not impossible. Furthermore, the DM may get an unsatisfactory solution if some of the objective functions correlate with each other (Steuer, 1986). This is also demonstrated in Tanner (1991) with an example originally formulated by P. Korhonen. The problem (involving three candidates and five objectives) is about choosing a spouse. There, the weights representing the preferences of the DM result with a spouse who is the worst in the objective that was given the biggest weight, that is, the highest importance.

Overall, we can say that it is not necessarily easy for the DM (or the analyst) to control a solution process with weights because weights behave in an indirect way. It makes no sense to end up in a situation where one tries to guess such weights that would produce a desirable solution. Because the DM can not be properly supported in this, (s)he is likely to get frustrated. Instead, it is then better to use real interactive methods (see Section 6.3) where more intuitive preference information can be used.

6.2.2.2. *ε-Constraint Method*

If one of the objective functions is selected to be optimized and the others are converted into constraints, we get the scalarization of the *ε-constraint method* (Haimes *et al.*, 1971; Chankong and Haimes, 1983):

$$\begin{array}{ll} \text{minimize} & f_\ell(\mathbf{x}) \\ \text{subject to} & f_j(\mathbf{x}) \le \varepsilon_j \quad \text{for all} \quad j = 1, \ldots, k, \quad j \ne \ell, \\ & \mathbf{x} \in S, \end{array} \tag{6.3}$$

where $\ell \in \{1, \ldots, k\}$ and ε_j are upper bounds for the objectives f_j, $j \ne \ell$.

The solution of problem (6.3) is always weakly Pareto optimal and Pareto optimal if it is unique. On the other hand, $\mathbf{x}^* \in S$ is Pareto optimal if and only if it solves (6.3) for every $\ell = 1, \ldots, k$, where $\varepsilon_j = f_j(\mathbf{x}^*)$ for $j = 1, \ldots, k$, $j \ne \ell$. Thus, ensuring Pareto optimality means either solving k problems or obtaining a unique solution (which is not necessarily easy to verify in practice). What is positive when compared to the weighting method is that the ε-constraint method can find any Pareto optimal solution even for nonconvex problems.

In practice, it is not always easy to set the upper bounds so that problem (6.3) has feasible solutions. It may also be difficult to select which of the objective functions should be the one to be optimized. These choices may affect the solutions obtained. For using the ε-constraint method as an a posteriori method, systematic ways of perturbing the upper bounds are suggested in Chankong and Haimes (1983). On the other hand, when used in an *a priori* way, the drawback is that if there is a promising solution really close to the bound specified but on the infeasible side, it will never be found because single objective optimizers must obey the constraints specified. However, the DM may want to study solutions corresponding to different bounds. If this is the case, it is again recommended to use interactive methods.

6.2.2.3. *Evolutionary Multi-Objective Optimization*

Since most of this book is devoted to evolutionary methods for multi-objective optimization, we here only wish to discuss some differences between EMO approaches and scalarization based approaches. As mentioned before, EMO approaches are *a posteriori* type of methods and they try to generate an approximation of the Pareto optimal set. In bi-objective optimization problems, it is easy to plot the objective vectors produced on a plane and ask the DM to select the most preferred one. While looking at the

visualization, the DM sees an overview of the trade-offs in the problem and most likely can choose the final solution. However, if the problem has more than two objectives, there is no natural way of visualization for objective vectors but one has to settle for projections or use other additional tools which are not necessarily very intuitive. This means that it is problematic to represent the many different solutions for the DM to compare. Another question is how to support him/her in selecting one of many solutions in a reasoned way. Furthermore, generating a good representation of a Pareto optimal set in a higher-dimensional feasible objective region necessitates high population sizes which implies high computational costs. If function evaluations are costly, the calculation takes a lot of effort. Besides, there may be many areas in the Pareto optimal set that the DM is not interested in. In such cases, we waste computational resources in finding solutions that are not needed at all.

On the other hand, it is not sensible, e.g., to restrict consideration to two objectives only, for the purpose of intuitive visualization. It is better to consider the problem as a whole and use as many objectives as needed instead of artificial simplifications. Furthermore, as mentioned earlier, EMO approaches do not necessarily guarantee that they generate Pareto optimal solutions. Because of the above-mentioned aspects, EMO approaches may not always be the best methods for solving multi-objective optimization problems and that is why we introduce scalarization based and interactive methods, in particular, to be considered as alternative approaches. When using them, the DM can concentrate on interesting solutions only and computational effort is not wasted. Furthermore, the DM can decide how many solutions (s)he wants to compare at a time.

The strength of evolutionary approaches is their wide applicability to, e.g., nondifferentiable and nonconvex problems. We wish to emphasize that this positive feature can be combined with scalarization based approaches by using evolutionary algorithms (i.e., not EMO but single objective optimizers) for solving the scalarized problem.

6.3. Interactive Multi-Objective Optimization

As said in the introduction, in interactive multi-objective optimization methods, a solution pattern is formed and repeated and the DM specifies preference information progressively during the solution process. In other words, the solution process is iterative and the phases of preference elicitation and solution generation alternate. In brief, the main steps of a

general interactive method are the following: (1) initialization (e.g., calculating ideal and nadir values and showing them to the DM), (2) generate a Pareto optimal starting point (some neutral compromise solution or solution given by the DM), (3) ask for preference information from the DM (e.g., aspiration levels or number of new solutions to be generated), (4) generate new Pareto optimal solution(s) according to the preferences and show it/them and possibly some other information about the problem to the DM. If several solutions were generated, ask the DM to select the best solution so far, and (6) stop, if the DM wants to. Otherwise, go to step (3).

The most important stopping criterion is the satisfaction of the DM in some solution. (Some interactive methods use also algorithmic stopping criteria but we do not go into such details here.) In each iteration, some information about the problem or solutions available is given to the DM and then (s)he is supposed to answer some questions or to give some other kind of information. New solutions are generated based on the information specified. In this way, the DM directs the solution process towards such Pareto optimal solutions that (s)he is interested in and only such solutions are generated.

The advantage of interactive methods is that the DM can specify and correct his/her preferences and selections during the solution process. Because of the iterative nature, the DM does not need to have any global preference structure and (s)he can *learn* during the solution process. This is a very important strength of interactive methods. Actually, finding the final solution is not always the only task but it is also noteworthy that the DM gets to know the problem, its possibilities and limitations.

We can say that interactive methods overcome weaknesses of *a priori* and *a posteriori* methods: the DM does not need a global preference structure and only interesting Pareto optimal solutions need to be considered. The latter means both savings in computational cost, which in many computationally complicated real problems is a significant advantage, and avoids setting cognitive overload on the DM, which the comparison of many solutions typically implies.

Many interactive methods exist and none of them is superior to all the others but some methods may suit different DMs and problems better than the others. Methods differ from each other by both the style of interaction and technical realization: e.g., what kind of information is given to the DM, the form of preference information specified by the DM and what kind of a scalarizing function is used or, more generally, which inner process is used to generate Pareto optimal solutions (Miettinen, 1999). It is always

important that the DM finds the method worthwhile and is able to use it properly, in other words, the DM must find preferences easy and intuitive to provide in the style selected. In many cases, we can identify two phases in the solution process: a *learning phase* when the DM wants to learn about the problem and what kind of solutions are feasible and a *decision phase* when the most preferred solution is found in the region identified in the first phase. If so desired, the two phases can be used iteratively, as well.

Descriptions of interactive methods are given, e.g., in Buchanan (1986); Chankong and Haimes (1983); Hwang and Masud (1979); Miettinen (1999); Sawaragi *et al.* (1985); Steuer (1986); Stewart (1992); Vanderpooten and Vincke (1989) and methods with applications to large-scale systems and industry are presented in Haimes *et al.* (1990); Statnikov (1999); Tabucanon (1988). Special attention to describing methods for nonlinear problems is paid in Miettinen (1999). Here we only describe a few interactive methods. We concentrate on methods where the DM specifies preferences in the form of reference points or classification.

6.3.1. *Reference Point Approaches*

In reference point based methods, the DM first specifies a reference point $\bar{\mathbf{z}} \in \mathbf{R}^k$ consisting of desirable aspiration levels for each objective and then this reference point is projected onto the Pareto optimal set. That is, a Pareto optimal solution closest to the reference point is found. The distance can be measured in different ways. Specifying a reference points is an intuitive way for the DM to direct the search of the most preferred solution. It is straightforward to compare the point specified and the solution obtained without artificial concepts. Examples of methods of this type are the reference point method and the 'light beam search'.

The reference point method is based on using a so-called *achievement (scalarizing) function* (Wierzbicki, 1982). The achievement function measures the distance between the reference point and Pareto optimal solutions and produces a new Pareto optimal solution closest to the reference point. The beauty here is that Pareto optimal solutions are generated no matter how the reference point is specified, that is, they can be attainable or not. We have an example of an achievement function in the problem

$$\begin{aligned} &\text{minimize} \quad \max_{i=1,\ldots,k} \left[w_i(f_i(\mathbf{x}) - \bar{z}_i) \right] + \rho \sum_{i=1}^{k} w_i f_i(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{6.4}$$

where w_i $(i = 1, \ldots, k)$ are fixed scaling coefficients, e.g., $w_i = 1/(z_i^{\text{nad}} - z_i^{\star\star})$ and $\rho > 0$ is a relatively small scalar. The solution of problem (6.4) is Pareto optimal and, as said, different Pareto optimal solutions can be generated by setting a different reference point (Miettinen, 1999).

In the reference point method, the DM specifies a reference point and the corresponding solution of (6.4) is shown to him/her. In addition, the DM is shown k other solutions obtained by slightly shifting the reference point in each coordinate direction. Thus, the DM can compare $k+1$ Pareto optimal solutions close to the reference point. Then the DM can set a new reference point (i.e., adjust the reference point according to his/her preferences) and the solution process continues as long as the DM wants to. When the Pareto optimal solutions are generated, the DM learns more about the possibilities and limitations of the problem and, therefore, can use more appropriate reference points.

Because of the intuitive character of reference points, it is advisable to use the achievement scalarizing function even when it is not possible to use an interactive approach. This means that the DM expresses his/her hopes in the form of a reference point and the solution of (6.4) is then shown to him/her. In this way, a reference point based approach can be used as an *a priori* method. It is also possible to use reservation levels representing objective values that should be achieved (besides aspiration levels). For further details, see Wierzbicki *et al.* (2000).

Another reference point based method is the *'light beam search'* (Jaszkiewicz and Slowinski, 1999). It uses a similar achievement function as the reference point method but combined with tools of multiattribute decision analysis (designed for comparing a discrete set of solutions). Besides a reference point, the DM must supply thresholds for objective functions describing indifference and preference in objective values. This information is used to derive outranking relations between solutions. As a result, incomparable or indifferent solutions are not displayed to the DM.

6.3.2. *Classification-Based Methods*

As discussed, moving from one Pareto optimal solution to another implies trading off. In other words, to move to another Pareto optimal solution where some objective function gets a better value, some other objective function must be allowed to get worse. This is the starting point of classification-based methods where the DM studies a Pareto optimal solution and says what kind of changes in the objective function values would

lead to a more preferred solution. Larichev (1992) has shown that for DMs classification is a cognitively valid way of expressing preference information.

Classification is an intuitive way for the DM to direct the solution process because no artificial concepts are used. Objective function values are as such meaningful and understandable for the DM. The DM can express hopes about improved solutions and directly see and compare how well the hopes could be realized.

To be more specific, when classifying objective functions the DM indicates which function values should improve, which ones are acceptable and which are allowed to get worse. In addition, amounts of improvement or impairments are asked from the DM. There exist several classification-based interactive multi-objective optimization methods. They use different numbers of classes and generate new solutions in different ways.

Let us point out that expressing preference information as a reference point (Miettinen and Mäkelä, 2002; Miettinen *et al.*, 2006) is closely related to classification. However, when classification assumes that some objective function must be allowed to get worse, a reference point can be set without considering the current solution. Even though it is not possible to improve all objective function values of a Pareto optimal solution simultaneously, the DM can still express preferences without paying attention to this fact and then see what kind of solutions are feasible. On the other hand, when using classification, the DM is more in control and selects functions to be improved and specifies amounts of impairment for the others.

Next, we briefly introduce the satisficing trade-off method and then describe the NIMBUS method in some more detail. We pay more attention to NIMBUS (and software implementing it) because we shall refer to it later when discussing applications.

6.3.2.1. *Satisficing Trade-Off Method*

The satisficing trade-off method (STOM) (Nakayama and Sawaragi, 1984) is based on the classification of objective functions at the current Pareto optimal solution into the three classes described earlier. A reference point can be formed based on this information.

For functions whose values the DM wants to improve, (s)he also has to specify desirable aspiration levels. If some function has an acceptable value, it is set as the corresponding aspiration level. Under some assumptions, it is possible to calculate how much impairment should be allowed in the other objective functions in order to attain the desired improvements. This is

called automatic trade-off (Nakayama, 1995). In this way, the DM has to specify less information. Once all components of a reference point have been set, one can solve a scalarized problem

$$\begin{aligned} &\text{minimize} \quad \max_{i=1,\ldots,k} \left[\frac{f_i(\mathbf{x}) - z_i^{\star\star}}{\bar{z}_i - z_i^{\star\star}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{\bar{z}_i - z_i^{\star\star}} \\ &\text{subject to } \mathbf{x} \in S, \end{aligned} \tag{6.5}$$

where $\rho > 0$ is a relatively small scalar. Then, the solution of (6.5) (which is guaranteed to be Pareto optimal, see Miettinen (1999)) is shown to the DM and (s)he can stop or classify the objective functions again. The DM can easily learn about the problem by comparing the hopes expressed in the classification and the Pareto optimal solution obtained.

If it is not possible to use automatic trade-off, classifying the objective functions or setting a reference point are almost the same, as discussed earlier. The only difference is that here the reference point is set such that some objective functions must be allowed to get impaired values. Let us finally mention that STOM has been applied to many engineering problems, e.g., in Nakayama (1995); Nakayama and Furukawa (1985); Nakayama and Sawaragi (1984).

6.3.2.2. *The NIMBUS Method*

The NIMBUS method (Miettinen, 1999; Miettinen and Mäkelä, 1995, 1999, 2000, 2006) is an interactive method based on classification of the objective functions into up to five classes. To be more specific, the DM is asked to specify how the current Pareto optimal solution $\mathbf{f}(\mathbf{x}^h)$ should be improved by classifying the objective functions into classes where the functions f_i

- should be improved as much as possible ($i \in I^{imp}$),
- should be improved until a specified aspiration level $\bar{z}_i$ ($i \in I^{asp}$),
- are satisfactory at the moment ($i \in I^{sat}$),
- can impair till a specified bound ε_i ($i \in I^{bound}$) and
- can change freely ($i \in I^{free}$).

A classification is feasible if at least one of the objective functions is allowed to get worse. Then the original multi-objective optimization problem is converted into a scalarized problem using the classification information specified. The solution of the scalarized problem reflects how well the hopes expressed in the classification could be achieved.

There exist several variants of NIMBUS (Miettinen, 1999; Miettinen and Mäkelä, 1995, 1999, 2000, 2006). Here we concentrate on the synchronous version (Miettinen and Mäkelä, 2006), where several scalarizing functions can be used based on a classification once expressed. Because they take the preference information into account in slightly different ways (Miettinen and Mäkelä, 2002), the DM can learn more about different solutions satisfying his/her hopes and choose the one that best obeys his/her preferences. An example of the scalarized problems used is

$$\begin{aligned} \text{minimize} \quad & \max_{i \in I^{imp},\, j \in I^{asp}} \left[\frac{f_i(\mathbf{x}) - z_i^\star}{z_i^{nad} - z_i^{\star\star}}, \frac{f_j(\mathbf{x}) - \bar{z}_j}{z_j^{nad} - z_j^{\star\star}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{z_i^{nad} - z_i^{\star\star}} \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^h) \text{ for all } i \in I^{imp} \cup I^{asp} \cup I^{sat}, \\ & f_i(\mathbf{x}) \leq \varepsilon_i, \text{ for all } i \in I^{bound}, \ \mathbf{x} \in S, \end{aligned} \tag{6.6}$$

where $\rho > 0$ is a relatively small scalar. The three other scalarized problems used in the synchronous NIMBUS method and further details are given in Miettinen and Mäkelä (2006).

Once the DM has classified the objective functions, (s)he can decide how many Pareto optimal solutions (between one and four) based on this information (s)he wants to see and compare. Then, as many scalarized problems are formed and solved and the new solutions are shown to the DM together with the current solution. If the DM has found the most preferred solution, the solution process stops. Otherwise, the DM can select a solution as a starting point of a new classification or ask for a desired number of intermediate (Pareto optimal) solutions between any two solutions generated so far. The DM can also save any interesting solutions to a database and return to them later. All the solutions considered are Pareto optimal. For details of the algorithm, see Miettinen and Mäkelä (2006).

In the initialization phase of the NIMBUS method, the ranges in the Pareto optimal set, that is, the ideal and the nadir objective vectors are computed to give the DM some information about the possibilities of the problem. The starting point of the solution process can be specified by the DM or it can be a neutral compromise solution located approximately in the middle of the Pareto optimal set. To get it, we set $(\mathbf{z}^{\text{nad}} + \mathbf{z}^{\star\star})/2$ as a reference point and solve (6.4).

In NIMBUS, the DM iteratively expresses his/her desires and learns about the feasible solutions available for the problem considered. Unlike some other methods based on classification, the success of the solution process does not depend entirely on how well the DM manages in specifying

the classification and the appropriate parameter values. It is important that the classification is not irreversible. Thus, the DM is free to go back or explore intermediate points. (S)he can easily get to know the problem and its possibilities by specifying, e.g., loose upper bounds and examining intermediate solutions. NIMBUS has been successfully applied, e.g., in the fields of optimal control and optimal design (Hämäläinen *et al.*, 2003; Heikkola *et al.*, 2006; Madetoja *et al.*, 2006; Miettinen *et al.*, 1998).

As far as software is concerned, the interactive nature of the solution process naturally sets its own requirements (Hakanen, 2006). First of all, a good graphical user-interface (GUI) is needed in order to enable the interaction between the DM and the method. In addition, visualizations of the solutions obtained must be available for the DM to compare and evaluate the solutions generated. With interactive methods, more than three objective functions can easily be considered, which sets more requirements on the visualization when compared to, e.g., visualizing the Pareto optimal set for bi-objective problems.

Currently, the NIMBUS method has two implementations: WWW-NIMBUS® and IND-NIMBUS®. The *WWW-NIMBUS®* system (Miettinen and Mäkelä, 2000, 2006) has been operating via the Internet at `http://nimbus.it.jyu.fi` since 1995 and can be used free of charge for teaching and academic purposes. Only a browser is required for using WWW-NIMBUS® and, therefore, the user has always the latest version available. All the computation is performed in the server computer at the University of Jyväskylä.

As far as using WWW-NIMBUS® is concerned, one can create an account of one's own or visit the system as a guest. Once an account has been created, it is possible to save problems and solutions in the system. WWW-NIMBUS® takes the user from one web page to another. The problem to be solved can be input by filling a web form (or as a subroutine). The system first asks for the name and the dimensions of the problem. In the second web page, the user can type in the formulas of each objective and constraint function as well as ranges (and initial values) for variables.

There are different single-objective optimizers available for solving the scalarized problems formed and the user can decide after each classification which optimizer to use or use the default one. The proximal bundle method (Mäkelä and Neittaanmäki, 1992) is a local optimizer and needs initial values for variables as well as (sub)gradients for functions. (The system can generate the latter automatically.) Alternatively, it is possible to use two variants of (global) real-coded genetic algorithms that differ from each other

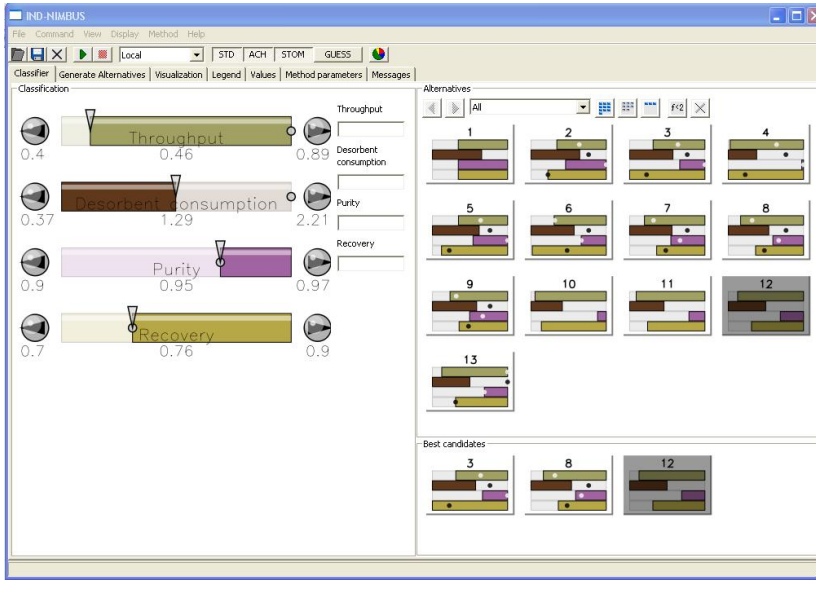


Fig. 6.1. A screenshot of IND-NIMBUS®. (This figure is available in color in the file `ind_nimbus.jpg` in Chapter 6 on the CD.)

in constraint handling (Miettinen *et al.*, 2003b). These optimizers can also handle mixed-integer problems. A hybrid of a global and a local optimizer can also be used. There are also different visualizations available to aid the user in analyzing and comparing different Pareto optimal solutions. The system has a tutorial that guides the user through the different phases of the interactive solution process. In addition, each web page has a separate help available.

IND-NIMBUS® (Miettinen, 2006; Ojalehto *et al.*, 2007) is a commercial implementation of the NIMBUS method developed for solving industrial multi-objective optimization problems (`http://ind-nimbus.it.jyu.fi/`). IND-NIMBUS® is available for Linux and MS-Windows operating systems. A screenshot of IND-NIMBUS® can be seen in Fig. 6.1. The bars on the left represent the current values of objective functions and the DM can classify the functions by clicking with a mouse or by specifying desired function values. The window on the right shows Pareto optimal solutions generated so far and interesting solutions can be saved in the lower right corner as best candidates.

Problems formulated with various simulators or modeling tools can be connected with IND-NIMBUS® and different underlying single-objective optimizers can be used depending on the properties of the problem considered. IND-NIMBUS® has been connected, e.g., to the BALAS® process simulator (`http://virtual.vtt.fi/virtual/balas/`), developed at the VTT Technical Research Center of Finland. In addition to the single-objective optimizers available in WWW-NIMBUS®, e.g., the IPOPT optimizer (Wächter and Biegler, 2006) has been used in IND-NIMBUS®.

In what follows, we call the combination of IND-NIMBUS® and some modeling tool or a simulator by the name *IND-NIMBUS® process design tool*. In Section 6.5 we discuss how it has been applied in some chemical process design problems.

6.3.3. *Some Other Interactive Methods*

A natural way of developing new methods is hybridizing ideas of different existing ones. It is, e.g., fruitful to hybridize ideas of *a posteriori* and interactive methods. In this way, the DM can both get a general overview of the possibilities and limitations of the problem and direct the search to a desired direction in order to find the most preferred solution. An example of such a method is introduced in Miettinen *et al.* (2003a), where NIMBUS (see Subsection 6.3.2.2) is hybridized with the feasible goals method (Lotov *et al.*, 2004). The latter generates visual interactive displays of the feasible objective vectors which helps the DM in understanding what kinds of solutions are available. Then it is easier to make classifications for NIMBUS.

Another hybrid is described in Klamroth and Miettinen (2007), where an adaptive approximation method (Klamroth *et al.*, 2002) approximating the Pareto optimal set is hybridized with reference point ideas. This means that the approximation is made more accurate only in those parts of the Pareto optimal set that the DM is interested in. Finally, let us mention one more hybrid method where reference points and achievement scalarizing functions are hybridized in EMO, see Thiele *et al.* (2007). On a general level, the idea is the same as in the previous hybrid but here the achievement scalarizing function is incorporated in the fitness evaluation and the interactive algorithm is different. Other ideas of handling preferences in EMO are surveyed in Coello (2000).

6.4. Interactive Approaches in Chemical Process Design

Multi-objective optimization has been applied to problems in chemical engineering frequently during the last 25 years (see, e.g., Andersson (2000); Chakraborty and Linninger (2002); Clark and Westerberg (1983); Kawajiri and Biegler (2006a); Ko and Moon (2002); Lim *et al.* (1999); Subramani *et al.* (2003)) as described in Chapter 2. Usually, only two objective functions have been considered and the Pareto optimal set has been approximated by using either the weighting method or the ε-constraint method. In other words, the simplest methods have been used and sometimes the authors have not realized that they have been using a multi-objective optimization method. As mentioned, regardless of their simplicity, these methods have serious drawbacks. Recently, EMO methods have become popular in solving chemical engineering problems, but still only two or three objectives have been considered maybe due to the limitations of EMO approaches discussed earlier (Bhaskar *et al.*, 2000; Rajesh *et al.*, 2001; Roosen *et al.*, 2003; Subramani *et al.*, 2003; Tarafder *et al.*, 2005; Zhang *et al.*, 2002).

Interactive multi-objective optimization methods have considerable advantages over the methods mentioned above. However, they have been used very rarely in chemical engineering. For example, interactive methods can not be found in the survey of Marler and Arora (2004) and they are only briefly mentioned in Andersson (2000) and Bhaskar *et al.* (2000). This might be due to the lack of knowledge of interactive methods or the lack of appropriate interactive multi-objective optimization software. The few examples of interactive multi-objective optimization in chemical engineering include Grauer *et al.* (1984) and Umeda and Kuriyama (1980).

In what follows, we describe and summarize research on multi-objective optimization in chemical engineering reported in Hakanen (2006) and Hakanen *et al.* (2004, 2005, 2006, 2008, 2007). These studies have focused on offering chemical engineering an efficient and practical way of handling all the necessary aspects of the problem, that is, to be able to simultaneously consider several conflicting objective functions that affect the behaviour of the problem considered. Therefore, they have been solved using the interactive NIMBUS method.

6.5. Applications of Interactive Approaches

Interactive multi-objective optimization can successfully be applied in chemical process design problems. For example, encouraging experiences

related to papermaking and sugar industries have been obtained in Hakanen *et al.* (2004, 2005, 2006, 2008, 2007). The reported solutions of these industrial problems are based on utilizing the NIMBUS method and IND-NIMBUS® and involving DMs having experience and knowledge about the problems in question. In this section, we shortly describe problems related to simulated moving bed processes, water allocation in a paper mill and a heat recovery system design. The interactive solution process is described in more detail for the first problem in order to give an idea of the interaction between the method and the DM. Other problems are described on a more general level with further references.

6.5.1. *Simulated Moving Bed Processes*

Simulated moving bed (SMB) processes are related to the separation of chemical products. Efficient purification techniques are crucial in chemical process industries. Liquid chromatographic separation has been widely used for products with an extremely high boiling point, or thermally unstable products such as proteins. In liquid chromatographic processes, a small amount of feed mixture is supplied to an end of a column which is packed with adsorbent particles, and then pushed to the other end with desorbent (water, organic solvent, or mixture of these).

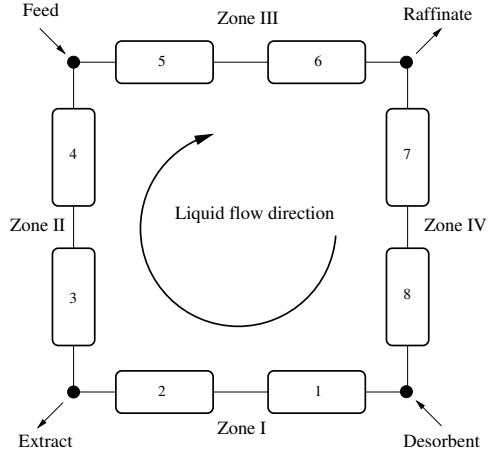


Fig. 6.2. A schematic diagram of an SMB process.

An SMB process is a realization of a continuous and counter-current operation of a liquid chromatographic separation process and it emerged

from the industry in the 1960's (Ruthven and Ching, 1989). An SMB unit consists of multiple columns which are packed with adsorbent particles. The columns are connected to each other making a circulation loop, see Fig. 6.2 (with eight columns). The feed mixture is inserted into the process in the upper left corner, while desorbent input is in the lower right corner. The two products, raffinate and extract, are collected in the upper right corner and the lower left corner, respectively. Feed mixture and desorbent are supplied between columns continuously. At the same time, the two products are withdrawn from the loop also continuously. The two inlet and two outlet streams are switched in the direction of the liquid flow at a regular interval. Because of the four inlet/outlet streams, the SMB loop has four liquid velocity zones as shown in Fig. 6.2.

The SMB model consists of partial differential equations (PDEs) for the concentrations of chemical components, restrictions for the connections between different columns and cyclic steady-state constraints (Kawajiri and Biegler, 2006b). Previously, the SMB processes have been usually optimized with respect to one objective only. Recently, multi-objective optimization has been applied in periodic separation processes (Ko and Moon, 2002), in gas separation and in SMB processes (Subramani *et al.*, 2003). Ko and Moon used a modified sum of weighted objective functions to obtain a representation of the Pareto optimal set. Their approach is valid for two objective functions only. On the other hand, Subramani et al. applied EMO to a problem where they had two or three objective functions.

In order to accelerate the process optimization, Kawajiri and Biegler (2006b) have developed an efficient full discretization approach combined with a large-scale nonlinear programming method for the optimization of SMBs. More recently, they have extended this approach to a superstructure SMB formulation and used the ε-constraint method to solve the bi-objective problem, where throughput and desorbent consumption were optimized (Kawajiri and Biegler, 2006a).

We can say that interactive methods have not been used to optimize SMB processes and, usually, only one or two objective functions have been considered. The advantages of interactive multi-objective optimization in SMB processes has been demonstrated in Hakanen *et al.* (2008, 2007) for the separation of fructose and glucose (the values of the parameters in the SMB model used come from Hashimoto *et al.* (1983); Kawajiri and Biegler (2006b)). In Hakanen *et al.* (2008, 2007), the problem formulation consists of four objective functions: maximize throughput (T, $[m/h]$), minimize consumption of solvent in the desorbent stream (D, $[m/h]$), maximize product

purity (P, [%]), and maximize recovery of the valuable component in the product (R, [%]). In Hakanen *et al.* (2007), a standard formulation of the SMB model is used while a superstructure formulation of SMBs is used in Hakanen *et al.* (2008). The superstructure formulation is a more general way to represent SMBs and it can produce novel SMB operating schemes (Kawajiri and Biegler, 2006a).

Fig. 6.3 shows the differences between standard and superstructure SMB configurations. The standard configuration has only one fixed place for each input and output stream whereas the superstructure SMB allows more diverse configurations because the input and output streams can be placed in some of the alternative positions shown in Fig. 6.3.

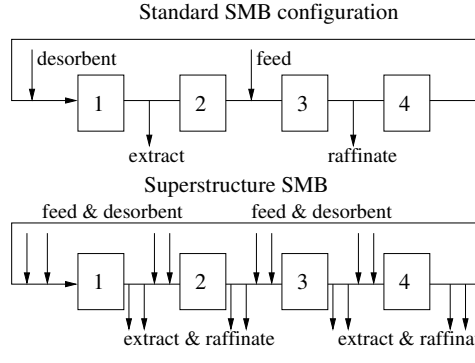


Fig. 6.3. A schematic diagram of the standard and the superstructure SMB processes.

For the PDE model of the SMB process, full discretization was used, that is, both temporal and spatial variables were discretized leading to a huge system of algebraic equations. The standard SMB optimization problem has 33 997 decision variables and 33 992 equality constraints while the superstructure SMB optimization problem has 34 102 decision variables and 34 017 equality constraints. Note that there are many more degrees of freedom in the superstructure formulation (altogether 85) than in the standard SMB formulation (5 degrees of freedom).

The SMB process is a dynamic process operating on periodic cycles which makes it a challenging optimization problem. The IPOPT optimizer (Wächter and Biegler, 2006) was used within the IND-NIMBUS® software (as an underlying optimizer) to produce new Pareto optimal solutions. The IPOPT optimizer was chosen because it has been developed for solving large scale optimization problems.

In what follows, we describe the interactive solution process for the standard SMB process using the IND-NIMBUS® process design tool. For further details, see Hakanen *et al.* (2007). The aim here is to give an understanding of the nature of an interactive solution process. The DM involved was an expert in SMB processes. First, in the initialization phase, the ranges in the Pareto optimal set were computed as $\mathbf{z}^\star = (0.891, 0.369, 97.2, 90.0)^T$ and $\mathbf{z}^{\text{nad}} = (0.400, 2.21, 90.0, 70.0)^T$. A neutral compromise solution $\mathbf{f}(\mathbf{x}^1) = (0.569, 1.58, 92.5, 76.9)^T$ was the starting point for the interactive solution process. Remember that the objective functions represented throughput (T), consumption of desorbent (D), purity (P) and recovery (R) and their values are here presented in objective vectors in this order (T, D, P, R). Note that the second objective function was minimized while the others were maximized.

In $\mathbf{f}(\mathbf{x}^1)$, the DM wanted to improve purity and throughput while desorbent consumption and recovery were allowed to deteriorate till specified levels. Therefore, the DM made the classification I^{imp}={P}, I^{asp}={T}, $\bar{z}_T = 0.715$, I^{bound}={D,R} with $\epsilon_D = 1.78$ and $\epsilon_R = 74.5$. The DM wanted to get four new solutions and they were $\mathbf{f}(\mathbf{x}^2) = (0.569, 1.56, 93.3, 74.5)^T$, $\mathbf{f}(\mathbf{x}^3) = (0.553, 1.43, 94.8, 70.0)^T$, $\mathbf{f}(\mathbf{x}^4) = (0.412, 1.07, 97.0, 70.0)^T$ and $\mathbf{f}(\mathbf{x}^5) = (0.570, 1.52, 93.9, 72.4)^T$. All the new solutions had a better purity than $\mathbf{f}(\mathbf{x}^1)$ but the bounds in the classification for D and R did not allow throughput to improve as much as the DM would have liked. Among the new solutions, he selected $\mathbf{f}(\mathbf{x}^3)$ as the basis of the next classification.

Next, he wanted to explore trade-off between improving recovery and letting desorbent consumption deteriorate (purity and throughput were satisfactory at the moment). The classification I^{asp}={R}, $\bar{z}_R = 0.796$, I^{sat}={P,T}, I^{bound}={D} with $\epsilon_D = 1.78$ was made and three different solutions were obtained: $\mathbf{f}(\mathbf{x}^6) = (0.497, 1.41, 93.9, 77.2)^T$, $\mathbf{f}(\mathbf{x}^7) = (0.481, 1.36, 94.2, 77.3)^T$ and $\mathbf{f}(\mathbf{x}^8) = (0.515, 1.46, 93.5, 77.1)^T$. The new solutions had a better recovery but it could only be achieved at the expense of throughput and purity. The DM liked $\mathbf{f}(\mathbf{x}^7)$ best because of the recovery and desorbent consumption. However, the purity was not so good.

In order to get a better understanding of the effects of the purity, the DM wanted to generate three intermediate solutions between $\mathbf{f}(\mathbf{x}^4)$ with the best purity and $\mathbf{f}(\mathbf{x}^7)$. The new solutions obtained were $\mathbf{f}(\mathbf{x}^9) = (0.426, 1.14, 96.3, 72.8)^T$, $\mathbf{f}(\mathbf{x}^{10}) = (0.443, 1.21, 95.6, 74.8)^T$ and $\mathbf{f}(\mathbf{x}^{11}) = (0.461, 1.29, 95.0, 76.2)^T$. The DM found $\mathbf{f}(\mathbf{x}^{11})$ to be very well balanced between all the objectives and selected it as the final, most preferred solution. The solution process was thus terminated. To summarize,

we can say that the DM could conveniently direct the solution process according to his preferences and obtain a satisfactory solution without too much cognitive burden. The information exchanged was intuitive and understandable for the DM. Let us point out that, altogether, only eleven Pareto optimal solutions were generated and, thus, the computational cost was rather low.

Considering an SMB design problem with four objective functions was a novel approach because, previously, only two or three objective functions had been considered. This enabled full utilization of the properties of the SMB model without any unnecessary simplifications. In addition, the DM obtained more thorough understanding of the interrelationships of different objectives considered and, thus, learned more about the problem.

Even better solutions can be found by using the superstructure formulation of an SMB process (when compared to the standard formulation), see Hakanen *et al.* (2008). Although producing Pareto optimal solutions is somewhat more time consuming for the superstructure formulation (because of more complicated formulas used), the model can describe the problem better and the DM could find a very satisfactory solution, as described in Hakanen *et al.* (2008).

6.5.2. *Water Allocation Problem*

Next, we consider an application related to the paper making process and study a water allocation problem for an integrated plant containing a thermomechanical pulping plant and a paper mill. For details of this problem with three objectives, see Hakanen *et al.* (2004).

The water management in paper making is guided by the need to produce paper efficiently. The process requires fresh water to keep the disturbing contaminants on a level that is acceptable for both paper quality and machine runnability. In modern mills, the water consumption has been pressed down to 5-10 m^3 per a ton of paper by matching water sources (e.g. filtrates) with water sinks (e.g. dilution duties) as illustrated in Fig. 6.4. First, wood is processed into pulp (with refining, screening, washing and bleaching). Then, the pulp is led to the paper machine. During the paper making process, (fresh) water is needed for various diluting duties in the washers and screening.

The upper part of Fig. 6.4 represents the thermomechanical pulping plant and the lower part represents the paper mill. The goal is to minimize the amount of fresh water taken into the process and also to minimize

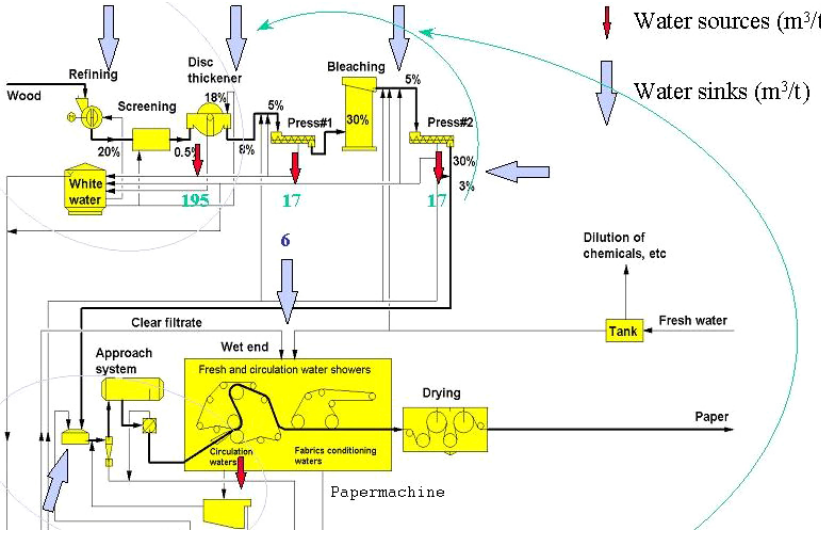


Fig. 6.4. Water sinks and sources in an integrated plant. (This figure is available in color in the file `water_allocation.jpg` in Chapter 6 on the CD.)

the amount of dissolved organic material in critical parts of the process by determining the right recycling of water. These objectives are clearly conflicting because if the amount of fresh water is reduced, then more organic material is accumulated into the water in the process. On the other hand, if the amount of fresh water is increased, then more organic material exits the process in waste water and the concentration of the organic material decreases.

The problem has three objective functions. The first of them describes the concentration of dissolved organic material in the white water of the paper machine. (White water is water that is removed from paper web in the paper machine.) This objective has an impact on the use of chemicals and quality of paper produced. The second objective describes the concentration of dissolved organic material in the pulp entering the bleaching process. This influences the pulp brightness and the use of bleaching chemicals. The third objective describes the amount of fresh water taken into the paper making process. The problem has two inequality constraints that restrict the consistency of the pulp going to the bleaching press and the washing press. The eight decision variables are of two types: splitters and valves.

As described, the water allocation problem is a multi-objective optimization problem by nature. Previously, this problem has typically been solved using the ε-constraint method by optimizing the consumption of fresh water while turning the other two objectives into inequality constraints. However, this approach can produce only one solution at a time corresponding to the upper bounds set for the new inequality constraints. It can also be quite difficult to set correct upper bounds to find the most desirable solution without knowing the behaviour of the problem well enough and the roles of the objective functions and the constraints may be varied. If an interactive approach, like the IND-NIMBUS® process design tool, is used instead, different solutions can be generated according to the preferences of the DM and the study of the interrelationships of the different objective functions is more flexible.

The consistencies of dissolved organic material in the white water of the paper machine and in the pulp entering the bleaching have no exact upper bounds which makes using the ε-constraint method poorly justified. Therefore, the selection of the best solution is not self-evident but the role of the DM is emphasized and, thus, a need for an interactive solution method is obvious. In Hakanen *et al.* (2004), the DM wanted primarily to study the effect of the first two objectives on the fresh water consumption because the approach was more flexible for this purpose than the ε-constraint method (which had been used earlier). The solution process with the IND-NIMBUS® process design tool provided a better understanding of the interrelationships of the objective functions when compared to the previous studies. The DM was able to rigorously study these relationships with various levels for dissolved organic materials and what kind of overall effect they have on the fresh water consumption. For a detailed description of the interactive solution process, see Hakanen *et al.* (2004).

6.5.3. *Heat Recovery System Design*

Finally, we discuss an example of designing a heat recovery system for the process water system of a paper mill (see Fig. 6.5). The problem involving four objective functions has been described and solved in Hakanen *et al.* (2005, 2006). We consider a virtual fine paper mill operating in a climate typical to northern latitudes, where ambient temperature varies according to the season.

The aim is to organize the heat management of the process water system in the most efficient way. A special characteristic of this optimization

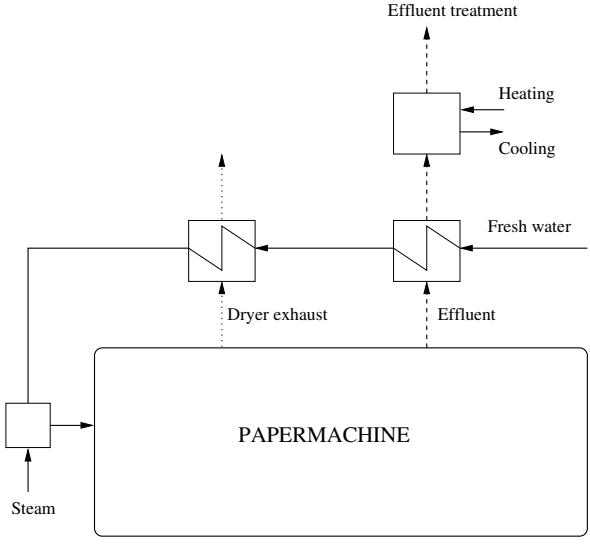


Fig. 6.5. A simplified flowchart of the heat recovery for the process water system of a paper mill.

problem is to consider the effect of seasonal changes in the climate that affect heat management. For example, fresh water taken into the process is much colder in the winter than in the summer. In practice, fresh water taken into the paper making process needs to be heated to the process temperature 60 ^{o}C. The heat sources available are the effluent from the process (at around 50 ^{o}C), which needs to be cooled down to around 37 ^{o}C to be suitable for an effluent treatment process, and dryer exhaust, which is moist air at the temperature 85 ^{o}C. In addition, steam can be used for the final heating of process water and effluent temperature can be controlled by external cooling or heating.

The design task is to determine the amount of heat recovered from the heat sources of the process to the heat sinks, that is, to estimate the size of heat exchangers and the amount of external energy needed for heating or cooling of the effluent coming from the paper machine. Both summer and winter scenarios are included in the model by combining two parallel process models for summer and winter conditions that are solved simultaneously. Ambient air/water temperatures are 20/20 ^{o}C and -5/2 ^{o}C for summer and winter, respectively.

If the heat management was designed only according to winter conditions, the sizing of the heat recovery system would be too large resulting in

large investment costs. On the other hand, if it was designed only for summer conditions, the energy consumption would be very high, because the heat recovery system would then be too small-sized. The higher the degree of heat recovery, the less external energy is required to satisfy the needs of the process. On the other hand, the size of heat exchangers (and hence investment costs) rises with an increased degree of heat recovery. Seasonal changes add to the complexity of the problem, since a recovery system designed for winter conditions can be oversized for summer conditions and lead to a need for external cooling, for instance.

As mentioned, the main trade-off here is between running costs, that is energy, and investment costs. Typically, this trade-off is handled with single objective optimization by formulating an objective function that consists of annualised energy and investment costs with estimated amortisation time and interest rate for the capital. The cooling or heating of effluent before treatment can be primarily either energy or investment cost. In case of heating or cooling with water, the running costs will dominate. However, in many cases, the use of water for cooling is not possible, and then a cooling facility is needed, which in the design phase is mainly an investment cost.

Instead of trying to estimate all relevant aspects to get a single objective function, we can formulate four separate objective functions to be minimized: steam needed for heating water for both summer and winter conditions, estimation of area for heat exchangers (heat exchange from effluent and dryer exhaust in winter conditions, which represents the maximum values for the exchangers), and the amount of cooling or heating needed for the effluent. The first two objectives tell how much, on the average, we need to provide steam for the system and give also an estimation of the size of a steam distribution system needed. The third objective describes the amount of heat exchange area needed. Once the area is known, we can estimate investment costs more accurately from real vendor data rather than using a general cost correlation. The fourth objective, the amount of energy needed to regulate the temperature of the effluent, is really an indication of the goodness of the design, since a value deviating from zero indicates either an additional investment (i.e., a cooling tower) or need for, e.g., steam. Finally, the three decision variables are the area of the effluent heat exchanger and the approach temperatures of the dryer exhaust heat exchangers for both summer and winter operations.

As said, traditionally, this type of a problem has been formulated as a single objective optimization problem hiding the interrelationships between the objectives. Then, monetary values have to be assigned *a priori* to en-

ergy and investments with a large degree of uncertainty in the correlations. Our approach eliminates these uncertainties and leaves it to the DM to assess the costs and their uncertainties *a posteriori* when the required energy and material flows are much better defined. Having four objective functions causes no troubles for an interactive method like NIMBUS and the problem can conveniently be solved, new insight into the problem obtained and a satisfactory solution found. For a detailed description of the interactive solution process with NIMBUS, see Hakanen *et al.* (2005, 2006).

6.6. Conclusions

We have introduced some interactive methods for multi-objective optimization problems and discussed their advantages. Interactive approaches allow the DM to learn about the problem considered and the interrelationships in it. In that way, (s)he gets deeper understanding of the phenomena in question. Because the DM can direct the search for the most preferred solution, only solutions that are interesting to him/her are generated which means savings in computation time. For computationally demanding problems, this may be a significant advantage. It is important that interactive methods can be applied to problems having more than two objective functions and, thus, the true nature of the problem can be taken into account.

If the problem considered has only two objective functions, methods generating a representation of the Pareto optimal set, like EMO approaches can be applied because it is simple to visualize the solutions on a plane. However, when the problem has more than two objectives, the visualization is no longer trivial and interactive approaches offer a viable alternative to solve the problem without artificial simplifications. Because interactive methods rely heavily on the preference information specified by the DM, it is important to select such a user-friendly method where the style of specifying preferences is convenient for the DM. In addition, the specific features of the problem to be solved must be taken into consideration.

We have shown with three applications how interactive multi-objective optimization can be utilized in chemical process design and demonstrated the benefits an interactive approach can offer. In all the cases, it was possible to solve the problems in their true multi-objective character and an efficient tool was created to support the DM (or designer) in the decision making problem.

Besides describing the potential of interactive methods, we have also discussed some properties of widely used methods because their shortcom-

ings do not seem to be generally known. However, it is important that when selecting a method, the limitations set by the method are known. Otherwise, the solutions obtained may not give a truthful impression about the problem in question.

References

Andersson, J. (2000). A survey of multiobjective optimization in engineering design, Tech. Rep. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University.

Bhaskar, V., Gupta, S. and Ray, A. (2000). Applications of multiobjective optimization in chemical engineering. *Reviews in Chemical Engineering* **16**, pp. 1–54.

Branke, J., Deb, K., Miettinen, K. and Slowinski, R. (eds.) (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, *Lecture Notes in Computer Science, State-of-the-Art Survey*, Vol. 4907 (Springer–Verlag), to appear.

Buchanan, J. T. (1986). Multiple objective mathematical programming: A review, *New Zealand Operational Research* **14**, pp. 1–27.

Chakraborty, A. and Linninger, A. (2002). Plant-wide waste management. 1. synthesis and multiobjective design, *Industrial & Engineering Chemistry Research* **41**, pp. 4591–4604.

Chankong, V. and Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology* (Elsevier Science Publishing, New York).

Clark, P. and Westerberg, A. (1983). Optimization for design problems having more than one objective, *Computers & Chemical Engineering* **7**, pp. 259–278.

Coello, C. (2000). Handling preferences in evolutionary multiobjective optimization: A survey, in *Proceedings of the 2000 Congress on Evolutionary Computation* (IEEE Service Center, Piscataway, NJ), pp. 30–37.

Das, I. and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems, *Structural Optimization* **14**, 1, pp. 63–69.

Edgeworth, F. Y. (1881). *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences* (C. Kegan Paul & Co., London), University Microfilms International (Out-of-Print Books on Demand), 1987.

Gass, S. and Saaty, T. (1955). The computational algorithm for the parametric objective function, *Naval Research Logistics Quarterly* **2**, pp. 39–45.

Grauer, M., Lewandowski, A. and Wierzbicki, A. (1984). Multiple-objective decision analysis applied to chemical engineering, *Computers & Chemical Engineering* **8**, pp. 285–293.

Haimes, Y. Y., Lasdon, L. S. and Wismer, D. A. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization, *IEEE Transactions on Systems, Man, and Cybernetics* **1**, pp. 296–297.

Haimes, Y. Y., Tarvainen, K., Shima, T. and Thadathil, J. (1990). *Hierarchical Multiobjective Analysis of Large-Scale Systems* (Hemisphere Publishing Corporation, New York).

Hakanen, J. (2006). *On Potential of Interactive Multiobjective Optimization in Chemical Process Design*, Jyväskylä Studies in Computing 68 (PhD thesis, University of Jyväskylä, Jyväskylä).

Hakanen, J., Hakala, J. and Manninen, J. (2006). An integrated multiobjective design tool for process design, *Applied Thermal Engineering* **26**, pp. 1393–1399.

Hakanen, J., Kawajiri, Y., Miettinen, K. and Biegler, L. (2007). Interactive multiobjective optimization for simulated moving bed processes, *Control & Cybernetics* **36**, 2.

Hakanen, J., Kawajiri, Y., Miettinen, K. and Biegler, L. (2008). Interactive multiobjective optimization of superstructure SMB processes, in V. Barichard, M. Ehrgott, X. Gandibleux and V. T'kindt (eds.), *Multi-Objective Programming and Goal Programming 2006, Proceedings* (Springer–Verlag), to appear.

Hakanen, J., Miettinen, K. and Mäkelä, M. (2004). An application of multiobjective optimization to process simulation, in P. Neittaanmäki, T. Rossi, S. Korotov, E. Onate, J. Periaux and D. Knörzer (eds.), *Proceedings of ECCOMAS 2004*, Vol. II (Jyväskylä, Finland), CD-ROM.

Hakanen, J., Miettinen, K., Mäkelä, M. and Manninen, J. (2005). On interactive multiobjective optimization with NIMBUS in chemical process design, *Journal of Multicriteria Decision Analysis* **13**, pp. 125–134.

Hämäläinen, J., Miettinen, K., Tarvainen, P. and Toivanen, J. (2003). Interactive solution approach to a multiobjective optimization problem in paper machine headbox design, *Journal of Optimization Theory and Applications* **116**, 2, pp. 265–281.

Hashimoto, K., Adachi, S., Noujima, H. and Maruyama, H. (1983). Models for the separation of glucose/fructose mixture using a simulated moving bed adsorber, *Journal of Chemical Engineering of Japan* **16**, 5, pp. 400–406.

Heikkola, E., Miettinen, K. and Nieminen, P. (2006). Multiobjective optimization of an ultrasonic transducer using NIMBUS, *Ultrasonics* **44**, pp. 368–380.

Hwang, C. L. and Masud, A. S. M. (1979). *Multiple objective decision making – methods and applications: A state-of-the-art survey* (Springer–Verlag).

Jaszkiewicz, A. and Slowinski, R. (1999). The 'light beam search' approach – an overview of methodology and applications, *European Journal of Operational Research* **113**, pp. 300–314.

Kawajiri, Y. and Biegler, L. (2006a). A nonlinear programming superstructure for optimal dynamic operations of simulated moving bed processes, *Industrial & Engineering Chemistry Research* **45**, pp. 8503–8513.

Kawajiri, Y. and Biegler, L. (2006b). Optimization strategies for simulated moving bed and powerfeed processes, *AIChE Journal* **52**, 4, pp. 1343–1350.

Klamroth, K. and Miettinen, K. (2007). Integrating approximation and interactive decision making in multicriteria optimization, *Operations Research* To appear (DOI:10.1287/opre.1070.0425).

Klamroth, K., Tind, J. and Wiecek, M. M. (2002). Unbiased approximation in multicriteria optimization, *Mathematical Methods of Operations Research* **56**, pp. 413–437.

Ko, D. and Moon, I. (2002). Multiobjective optimization of cyclic adsorption processes, *Industrial & Engineering Chemistry Research* **41**, 1, pp. 93–104.

Koopmans, T. (1951). Analysis and production as an efficient combination of activities, in T. Koopmans (ed.), *Activity Analysis of Production and Allocation: Proceedings of a Conference* (John Wiley & Sons, New York), pp. 33–97, Yale University Press, New Haven, London, 1971.

Kuhn, H. and Tucker, A. (1951). Nonlinear programming, in J. Neyman (ed.), *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press, Berkeley, LA), pp. 481–492.

Larichev, O. (1992). Cognitive validity in design of decision aiding techniques, *Journal of Multi-Criteria Decision Analysis* **1**, 3, pp. 127–138.

Lim, Y., Floquet, P., Joulia, X. and Kim, S. (1999). Multiobjective optimization in terms of economics and potential environment impact for process design and analysis in a chemical process simulator, *Industrial & Engineering Chemistry Research* **38**, pp. 4729–4741.

Lotov, A. V., Bushenkov, V. A. and Kamenev, G. K. (2004). *Interactive Decision Maps. Approximation and Visualization of Pareto Frontier* (Kluwer Academic Publishers, Boston).

Madetoja, E., Miettinen, K. and Tarvainen, P. (2006). Issue related to the computer realization of a multidisciplinary and multiobjective optimization system, *Engineering and Computing* **22**, pp. 33–46.

Mäkelä, M. and Neittaanmäki, P. (1992). *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control* (World Scientific Publishing Co., Singapore).

Marler, R. and Arora, J. (2004). Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization* **26**, 6, pp. 369–395.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization* (Kluwer Academic Publishers, Boston).

Miettinen, K. (2006). IND-NIMBUS for demanding interactive multiobjective optimization, in T. Trzaskalik (ed.), *Multiple Criteria Decision Making '05* (The Karol Adamiecki University of Economics in Katowice, Katowice), pp. 137–150.

Miettinen, K., Lotov, A. V., Kamenev, G. K. and Berezkin, V. E. (2003a). Integration of two multiobjective optimization methods for nonlinear problems, *Optimization Methods and Software* **18**, 1, pp. 63–80.

Miettinen, K. and Mäkelä, M. (1995). Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS, *Optimization* **34**, pp. 231–246.

Miettinen, K. and Mäkelä, M. (1999). Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation, *Journal of the Operational Research Society* **50**, 9, pp. 949–959.

Miettinen, K. and Mäkelä, M. (2000). Interactive multiobjective optimization system WWW-NIMBUS on the Internet, *Computers & Operations Research* **27**, pp. 709–723.

Miettinen, K. and Mäkelä, M. (2002). On scalarizing functions in multiobjective optimization, *OR Spectrum* **24**, pp. 193–213.

Miettinen, K. and Mäkelä, M. (2006). Synchronous approach in interactive multiobjective optimization, *European Journal of Operations Research* **170**, 3, pp. 909–922.

Miettinen, K., Mäkelä, M. and Toivanen, J. (2003b). Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms, *Journal of Global Optimization* **27**, pp. 427–446.

Miettinen, K. and Mäkelä, M. M. (2002). On scalarizing functions in multiobjective optimization, *OR Spectrum* **24**, 2, pp. 193–213.

Miettinen, K., Mäkelä, M. M. and Kaario, K. (2006). Experiments with classification-based scalarizing functions in interactive multiobjective optimization, *European Journal of Operational Research* **175**, 2, pp. 931–947.

Miettinen, K., Mäkelä, M. M. and Männikkö, T. (1998). Optimal control of continuous casting by nondifferentiable multiobjective optimization, *Computational Optimization and Applications* **11**, 2, pp. 177–194.

Nakayama, H. (1995). Aspiration level approach to interactive multi-objective programming and its applications, in P. Pardalos, Y. Siskos and C. Zopounidis (eds.), *Advances in Multicriteria Analysis* (Kluwer Academic Publishers), pp. 147–174.

Nakayama, H. and Furukawa, K. (1985). Satisficing trade-off method with an application to multiobjective structural design, *Large Scale Systems* **8**, pp. 47–57.

Nakayama, H. and Sawaragi, Y. (1984). Satisficing trade-off method for multiobjective programming, in M. Grauer and A. Wierzbicki (eds.), *Interactive Decision Analysis* (Springer–Verlag), pp. 113–122.

Ojalehto, V., Miettinen, K. and Mäkelä, M. (2007). Interactive software for multiobjective optimization: IND-NIMBUS, *WSEAS Transactions on Computers* **6**, 1, pp. 87–94.

Pareto, V. (1896). *Cours d'Economie Politique* (Rouge, Lausanne).

Pareto, V. (1906). *Manuale di Economia Politica* (Piccola Biblioteca Scientifica, Milan), translated into English by Ann S. Schwier (1971), Manual of Political Economy, MacMillan, London.

Podinovski, V. V. (1994). Criteria importance theory, *Mathematical Social Sciences* **27**, 3, pp. 237–252.

Rajesh, J., Gupta, S., Rangaiah, G. and Ray, A. (2001). Multi-objective optimization of industrial hydrogen plants, *Chemical Engineering Science* **56**, pp. 999–1010.

Roosen, P., Uhlenbruck, S. and Lucas, K. (2003). Pareto optimization of a combined cycle power system as a decision support tool for trading off investment vs. operating cost, *International Journal of Thermal Sciences* **42**, pp. 553–560.

Roy, B. and Mousseau, V. (1996). A theoretical framework for analysing the notion of relative importance of criteria, *Journal of Multi-Criteria Decision Analysis* **5**, 2, pp. 145–159.

Ruthven, D. and Ching, C. (1989). Counter-current and simulated counter-current adsorption separation processes, *Chemical Engineering Science* **44**, pp. 1011–1038.

Sawaragi, Y., Nakayama, H. and Tanino, T. (1985). *Theory of Multiobjective Optimization* (Academic Press, Orlando, FL).

Statnikov, R. B. (1999). *Multicriteria Design: Optimization and Identification* (Kluwer Academic Publishers, Dordrecht).

Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application* (John Wiley & Sons, New York).

Stewart, T. J. (1992). A critical survey on the status of multiple criteria decision making theory and practice, *OMEGA* **20**, 5–6, pp. 569–586.

Subramani, H., Hidajat, K. and Ray, A. (2003). Optimization of reactive SMB and varicol systems, *Computers & Chemical Engineering* **27**, pp. 1883–1901.

Tabucanon, M. T. (1988). *Multiple Criteria Decision Making in Industry* (Elsevier Science Publishers, Amsterdam).

Tanner, L. (1991). Selecting a text-processing system as a qualitative multiple criteria problem, *European Journal of Operational Research* **50**, pp. 179–187.

Tarafder, A., Rangaiah, G. and Ray, A. (2005). Multiobjective optimization of an industrial styrene monomer manufacturing process, *Chemical Engineering Science* **60**, pp. 347–363.

Thiele, L., Miettinen, K., Korhonen, P. J. and Molina, J. (2007). A preference-based interactive evolutionary algorithm for multiobjective optimization, Working Papers W-412, Helsinki School of Economics, Helsinki.

Umeda, T. and Kuriyama, T. (1980). Interactive solution to multiple criteria problems in chemical process design, *Computers & Chemical Engineering* **4**, pp. 157–165.

Vanderpooten, D. and Vincke, P. (1989). Description and analysis of some representative interactive multicriteria procedures, *Mathematical and Computer Modelling* **12**, pp. 1221–1238.

Vincke, P. (1992). *Multicriteria Decision-Aid* (John Wiley & Sons, Chichester).

Wächter, A. and Biegler, L. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* **106**, pp. 25–57.

Wierzbicki, A. (1982). A mathematical basis for satisficing decision making, *Mathematical Modelling* **3**, 25, pp. 391–405.

Wierzbicki, A., Makowski, M. and Wessels, J. (eds.) (2000). *Model-Based Decision Support Methodology with Environmental Applications* (Kluwer Academic Publishers, Dordrecht).

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria, *IEEE Transactions on Automatic Control* **8**, pp. 59–60.

Zhang, Z., Hidajat, K. and Ray, A. (2002). Multiobjective optimization of SMB and varicol process for chiral separation, *AIChE Journal* **48**, pp. 2800–2816.

Exercises

6.1 Consider a nonconvex multi-objective optimization problem *

$$\text{minimize } \{f_1(x), f_2(x)\}$$

$$\text{subject to } x \geq 0, x \in \mathbb{R},$$

where

$$f_1(x) = \sqrt{x^2 + 1},$$
$$f_2(x) = \begin{cases} -x^2 + 16 & \text{for } 0 \leq x \leq \sqrt{15}, \\ 1 & \text{for } x > \sqrt{15}. \end{cases} \tag{6.7}$$

The feasible objective region for the problem (6.7) is shown in Figure 6.6. Try to find four different Pareto optimal solutions for problem (6.7) by using the weighting method. What do you observe and why?

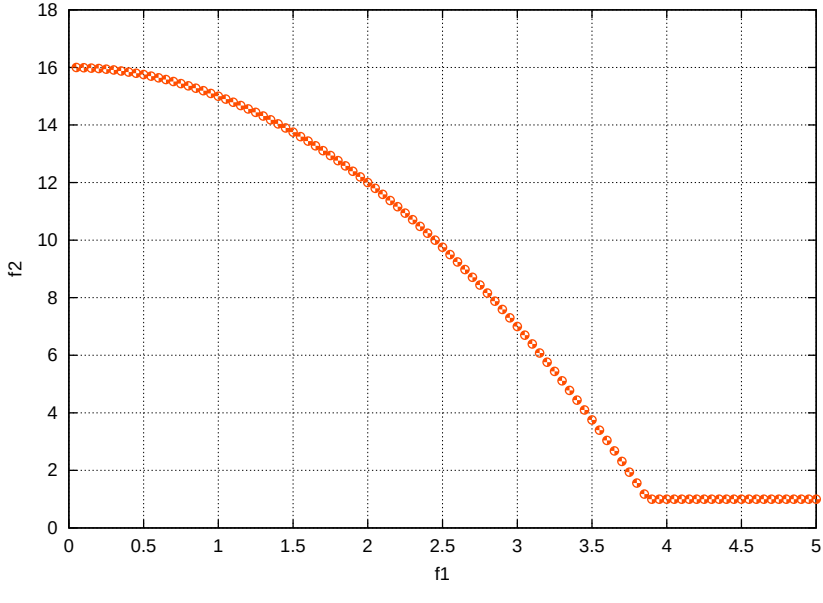


Fig. 6.6. Feasible objective region.

*V. Chankong & Y.Y. Haimes, Multiobjective Decision Making: Theory and Methodology (Elsevier Science Publishing, New York)

6.2 Consider a multi-objective optimization problem

$$\begin{aligned} &\text{minimize } \{-x, -y\} \\ &\text{subject to } 2x + 5y \leq 40, \\ &\qquad\qquad 2x + 3y \leq 28, \\ &\qquad\qquad x \leq 11, \\ &\qquad\qquad x, y \geq 0. \end{aligned}$$

Use the ε-constraint method to generate Pareto optimal solutions for this problem. Try using different upper bounds and generate four different Pareto optimal solutions.

6.3 WWW-NIMBUS is an implementation of the interactive NIMBUS method operating on the Internet (`http://nimbus.it.jyu.fi/`). Study the tutorial of the WWW-NIMBUS system (`http://nimbus.it.jyu.fi/N4/tutorial/index.html`) and answer the following questions.

a) What are the objective functions?
b) What is meant by classification?
c) What are the aspiration levels and upper bounds?
d) How can you generate new alternatives?

6.4 Input the problem described in the tutorial to the WWW-NIMBUS system and generate some Pareto optimal solutions by using both classification and generating of intermediate solutions. (Generate at least 5-10 solutions.) Compare the different solutions obtained by using the different visualizations available in WWW-NIMBUS. Which visualization did you prefer?

6.5 Use WWW-NIMBUS and solve the nutrition problem (saved in the system for guest users). Use both the symbolic and graphical classification in WWW-NIMBUS. Which one do you prefer? Why? For some classification, use both local and global (underlying) optimizer. Study the similarity of the results obtained and analyze the reasons for the similarity.

Chapter 7

Net Flow and Rough Sets: Two Methods for Ranking the Pareto Domain

Jules Thibault
Department of Chemical Engineering
University of Ottawa
Ottawa (Ontario) Canada K1N 6N5

Tel: (613) 652-5800 x6094; E-mail: Jules.Thibault@uottawa.ca

Abstract

This chapter presents a description of two multi-objective optimization (MOO) methods, Net Flow Method (NFM) and Rough Set Method (RSM), with a particular focus on engineering applications. Each of the methods provides its own algorithm for ranking the Pareto domain. NFM, which is a hybrid of the ELECTRE and PROMETHEE optimization schemes, employs the preferences of decision-makers in the form of three threshold values for each criterion and one set of relative weights that are used to classify the entire Pareto domain. RSM uses a set of decision rules that are based on the preferences of decision-makers that are established through the ranking of a small, diverse sample set extracted from the Pareto domain. These rules are then applied to the entire Pareto domain to determine the preferred zone of operation. Both methods require the intervention of experts to provide their knowledge and their preferences regarding the operation of the process.

The two methods are used to optimize the production of gluconic acid for multiple objectives. In this fermentation process, it is desired mainly to maximize the productivity and the final concentration of gluconic acid. Other objective functions, the final substrate concentration and the initial inoculum biomass concentration, can also be added to make a three- and four-objective optimization problem. It is shown that NFM and some variants of RSM performed similarly and possess good robustness.

Keywords: Net flow method, rough set method, thresholds, concordance index, discordance index, preference and non-preference rules, ranking, gluconic acid production.

7.1 Introduction

In recent years, the development and application of MOO techniques in chemical engineering have received wide attention in the literature. In complex chemical processes, the ability to select optimum operating conditions in the presence of multiple conflicting objectives, given the various economical and environmental constraints, is of paramount importance for the profitability of chemical plants. For this reason, MOO has been applied to many chemical process optimization problems. Excellent reviews on the applications of MOO in chemical engineering have been presented by Bhaskar *et al.* (2000) and Masuduzzaman and Rangaiah (2008).

Despite the numerous successful applications in chemical engineering, the implementation of MOO methods still remains a major challenge because of the inherent conflicting nature of the multiple objectives, whereby determining a compromised solution is far from being trivial. Traditional optimization techniques have dealt with multiple objectives by combining them into a single objective function composed of their weighted sum, or by focusing on a single objective while transforming the others into constraints. These techniques assume that the objective functions are well behaved, in the sense that they are either concave-shaped or convex-shaped and continuous, and that there

exists an optimal solution that will resolve any issues pertaining to conflicting objectives. Although single objective algorithms give satisfactory solutions in many cases, there are many drawbacks to their use, especially when the objectives under consideration are conflicting. Some of these issues are as follows. (1) Combining multiple objectives into a single objective function does not provide the decision-maker with information about trade-offs amongst the various objectives, or about alternative operating conditions. (2) Transforming a multi-objective problem into a single objective function composed of the weighted sum of the objectives strictly relies on the appropriate weights for the objectives, where the application of different weights can lead to a variety of solutions. (3) Standard single-objective optimization (SOO) techniques provide only one optimal solution even if other possible solutions exist. Hence these techniques are often plagued with the problem of finding global optima or multiple global optima, and can miss possible solutions if the functions are non-convex, multi-modal or discontinuous. They also require information about function derivatives and an initial estimate of the solution, which may not be readily available. On the other hand, global optimization techniques for SOO, such as evolutionary algorithms, have now been developed to overcome this problem. (4) Traditional optimization techniques do not incorporate the practical experience and knowledge of the decision-maker in regard to the overall behavior of the process.

In reality, many chemical processes are defined by complex equations where the application of SOO techniques does not provide satisfactory results in the presence of multiple conflicting objectives. Instead, the solution lies with the use of MOO techniques. MOO refers to the simultaneous optimization of multiple, often conflicting objectives, which produces a set of alternative solutions called the Pareto domain (Deb, 2001). These solutions are said to be Pareto-optimal in the sense that no one solution is better than any other in the domain when compared on all criteria simultaneously and in the absence of any preferences for one criterion over another. The decision-maker's experience and knowledge are then incorporated into the optimization procedure in order to classify the available alternatives in terms of his/her preferences (Doumpos and Zopounidis, 2002). MOO techniques

have many advantages: (1) they can simultaneously optimize multiple and conflicting objectives; (2) they usually have a global perspective and are not affected by multiple global or local optima, do not require information about function derivatives, and can be applied to functions that are non-convex, non-concave and discontinuous; and (3) they have the ability to generate multiple solutions that span the entire search space.

In recent years, new methodologies for generating and classifying the Pareto domain have been developed. With regards to generating the Pareto domain, a general class of MOO techniques called Evolutionary Algorithms has gained increasing attention in the literature and has been successfully applied to many engineering problems (Coello Coello, 1999; Deb, 2001; Fonseca *et al.*, 1995; Jaimes and Coello Coello, 2008; Liu *et al.*, 2003; Ramteke and Gupta, 2008; Shim *et al.*, 2002; Silva and Biscaia, 2003; Viennet *et al.*, 1996). However, a single universal technique acceptable to all does not yet exist, and results from current methodologies can vary significantly in terms of the proposed Pareto domain. For this reason, many standard benchmark test cases with varying degrees of difficulty have been developed to allow researchers to compare their techniques to others reported in the literature (Deb, 2001; Kursawe, 1990; Poloni *et al.*, 2000; Silva and Biscaa, 2003; Viennet *et al.*, 1996).

In fact, most decisions associated with daily human activities involve multiple objectives. Very often, the optimization process leading to a decision is so well integrated in the daily routine that it becomes natural and transparent to the decision-maker, and one is oblivious to the individual steps that are required to come to a decision. The same analogy can be extended to more complex situations where human intervention occurs. This is the case of many industrial decision-makers or experts who have a profound knowledge of their processes and who succeed in making appropriate decisions in order to render their process optimal despite the multiple conflicting objectives and constraints. These experts have, through a comprehensive knowledge of their processes, achieved an acceptable compromised solution of their multi-objective problems.

To easily implement MOO methods to these problems, it is necessary to devise a technique for incorporating as naturally as possible the knowledge of the decision-makers with regards to their preferences, given a set of conflicting objectives and constraints. At the same time, it is important to realize that there is inevitably a degree of uncertainty and vagueness in trying to capture the knowledge and preferences of the decision-makers. It is the purpose of this chapter to present two MOO techniques that allow the decision-maker's preferences to be encapsulated mathematically and used to determine an optimal solution. These two methods, described in turn, are Net Flow and Rough Sets.

Specifically, rest of this chapter begins by very briefly considering MOO in general, followed by a more detailed discussion of the algorithm of net flow and rough set methods, whilst illustrating these with simple examples. The last section of the chapter is devoted to the application of this method to the production of gluconic acid.

7.2 Problem Formulation and Solution Procedure

The first step in the optimization process is the formulation of the problem. A MOO problem is one in which each of the multiple objectives is maximized or minimized, subject to various constraints that the feasible solution must satisfy, and where the ranges for the input space variables are defined. Mathematically, the MOO problem can be expressed as follows.

$$\underbrace{F(X)}_{Min/Max} = \left[f_1(x_1..x_N),\, f_2(x_1..x_N),\, ...f_n(x_1..x_N) \right] \tag{7.1}$$

$$\begin{aligned} &subject\ to \quad G(X) \geq 0, \\ &\qquad\qquad H(X) = 0 \\ &where\ x_{i=1...N}^{(Lower\ Bound)} \leq x_{i=1...N} \leq x_{i=1...N}^{(Upper\ Bound)} \end{aligned}$$

The input space is predefined by the ranges associated with the independent variables $X = (x_1, x_2, x_3...x_N)^T$. This input space is often called the decision variable space or the actions, while the output or solution space, expressed by $F(x)$, is called the objective function space or performance space. From an engineering point of view, to determine

an optimal solution for a given system, one must obtain the set of objective function values that best satisfies the preferences of the decision-maker or the expert of the system. The decision variables or actions associated with the optimal set of objective functions are then implemented in the process. These usually correspond to process variables that the decision-maker can change to influence outcome of the multiple objectives.

A flow chart of the typical procedure for the MOO of a process is presented in Fig. 7.1. After obtaining a proper model of the process, the optimization method reduces to (1) circumscribing the Pareto domain, approximated by a sufficiently large number of non-dominated solutions; and (2) ranking the entire Pareto domain by order of preferences. Note that the Pareto domain represents the collection of solutions taken from the total solution set that are not dominated by any other solution within this set. In this respect, one solution is said to be dominated by another if the values of all optimization criteria of the first are worse than those of the second (Deb, 2001; Thibault *et al*., 2003). A genetic algorithm is often used to obtain the desired number of non-dominated solutions in order to adequately represent the entire Pareto domain (Halsall-Whitney and Thibault, 2006; Viennet *et al*., 1996). This first step is common to the majority of MOO techniques and is performed without any biased preference of a decision-maker. It is only required to know if a given criterion should be minimized, maximized or set as close as possible to a given target value.

The second step in the optimization process consists of ranking the entire Pareto domain in order of preferences based on the conscious (and often intuitive) knowledge that an expert has of his/her process. While the Pareto domain is first established from a domination perspective without the bias of *a priori* knowledge concerning the relative importance of the various objectives, its ranking requires that the expert incorporates his/her knowledge of the process into the optimization routine. There exist many MOO methods designed to accomplish this ranking, Net Flow Method (NFM) and Rough Set Method (RSM) being the two that are considered in this chapter.

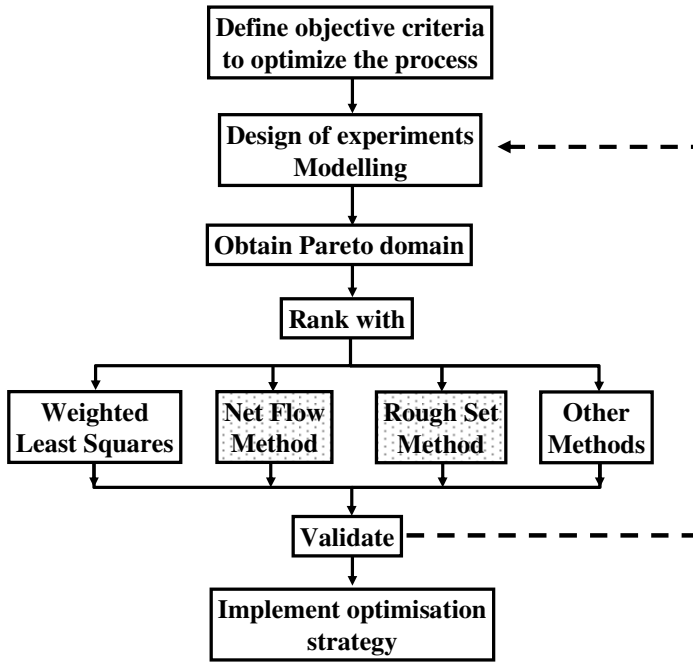


Fig. 7.1 Flow chart of MOO (Electronic copies of all figures are included in Chapter 7 folder on the accompanying CD).

The success of either NFM and RSM requires that a sufficiently large number of discrete solutions be identified to adequately represent the Pareto domain. The discrete solutions can originate from either experiments or simulations obtained from a model of the process. In engineering applications, the latter is preferred because of the typically large cost associated with generating experimental points. Very often, an experimental design will be used to allow modeling of the process in a manner that relates each of the objectives to all the input process variables.

Example 7.1 Consider a four-objective optimization problem. The four objectives are denoted as C_1, C_2, C_3, and C_4. It is desired to maximize C_1 and C_3, and minimize C_2 and C_4. Perform pair-wise comparisons of the three solutions listed in Table 7.1 to determine which solutions are dominated and which solutions are non-dominated.

Table 7.1 List of three solutions of Example 7.1.

Solution	Objectives			
	C_1	C_2	C_3	C_4
	Max	Min	Max	Min
1	40	20	5	100
2	31	18	3	200
3	30	25	2	300

Solution Comparing solution 1 to solution 2, objectives C_1, C_3, and C_4 are better for solution 1, but worse for C_2. The two solutions are non-dominated with respect to each other because each solution is better for at least one objective. However, the values of the four objectives of solution 3 are worse than those of solutions 1 and 2 such that solution 3 is a dominated solution with respect to the other two solutions and will be discarded. Nonetheless, it does not mean that solutions 1 and 2 will be part of the Pareto domain because they may be dominated by other solutions. In the end, the Pareto domain only contains solutions that are better for at least one of its objective criteria than all the other solutions within that domain.

7.3 Net Flow Method

This section concentrates on a class of outranking techniques called ELECTRE (**EL**imination **E**t **C**hoix **T**raduisant la **RE**alité, which when translated becomes ELimination and Choice Expressing the REality) that originated in the field of economics and finance (Brans *et al*., 1984; Derot *et al*., 1997; Doumpos and Zopounidis, 2002; Roy, 1991; Roy, 1978; Scarelli and Narula, 2002; Triantaphyllou, 2000), and which has also been successfully applied to the field of chemical engineering (Couroux *et al*., 1995; Halsall-Whitney *et al.*, 2003; Perrin *et al*., 1997; Renaud *et al.*, 2007; Thibault *et al*., 2001; Viennet *et al*., 1996). NFM, which was developed as a result of modifications made to the ELECTRE III method, is considered in this section as a method to efficiently rank all solutions of the Pareto domain based on the preferences of the decision-maker (Derot *et al.*, 1997). Another similar class of outranking methods, also derived from ELECTRE, to deal with MOO that has

attracted considerable interest is based on the so-called PROMETHEE methods (**P**reference **R**anking **O**rganization **METH**od for **E**nrichment **E**valuations) (Brans *et al.*, 1984, 1986). NFM is in fact an amalgam of these two powerful MOO methods.

In NFM, *a priori* knowledge of the process expressed by the decision-maker is incorporated into the optimization routine using four sets of ranking parameters listed below.

1. The first parameter gives the relative importance of each objective function or criterion k, expressed as a relative weight (W_k). In this algorithm, the various weights are normalized:

$$\sum_{k=1}^{n} W_k = 1 \tag{7.2}$$

2. The second parameter refers to the indifference threshold (Q_k), which defines the range of variation of each criterion for which it is not possible for the decision-maker to favor the criterion of one solution over the corresponding criterion of another solution. It therefore represents the range of values over which two objective functions are indiscernible.
3. The third parameter refers to the preference threshold (P_k). If the difference between two values for a given criterion exceeds this threshold, a preference is given to the better criterion. If the objective is to maximize a particular criterion, then the better solution is that with the larger value for that criterion, and vice versa.
4. The fourth parameter refers to the veto threshold (V_k), which serves to ban a solution relative to another solution if the difference between the values of a criterion is too high to be tolerated. A solution is banned if the veto threshold is violated for at least one of the objective functions, even if the other criteria are acceptable.

The three thresholds are established for each objective function such that the following relationship holds:

$$0 \le Q_k \le P_k \le V_k \tag{7.3}$$

The three thresholds represent a reference range set by the decision-maker to assess the values of the objective functions for each alternative

in the Pareto domain (Roy, 1978). In this context, the NFM algorithm is executed as follows.

1. First, for each combination of solutions in the Pareto domain, the difference between the values F_k of each objective function k is calculated by comparing solution i with solution j using the following relationship:

$$\Delta_k[i,\ j] = F_k(i) - F_k(j) \quad \begin{cases} i \in [1, M] \\ j \in [1, M] \\ k \in [1, n] \end{cases} \tag{7.4}$$

where M is the number of solutions approximating the Pareto domain. In subsequent equations, minimizing a criterion considers $\Delta_k[i, j]$, while maximizing a criterion considers its negative value, $-\Delta_k[i, j]$. When an objective criterion needs to meet a specified target, $F_k(i)$ and $F_k(j)$ correspond to the absolute values of the differences between the values of the criterion k and its target value; then, $\Delta_k[i, j]$ is used directly since it is desired to minimize the distance of the criterion to its target value.

2. Using the values of $\Delta_k[i, j]$, the individual concordance index $c_k[i, j]$ for each criterion is first determined for all n objective criteria and for each pair of solutions using the following relationships:

$$c_k[i, j] = \begin{cases} 1 & \text{if } \Delta_k[i, j] \le Q_k \\ \dfrac{P_k - \Delta_k[i, j]}{P_k - Q_k} & \text{if } Q_k < \Delta_k[i, j] \le P_k \\ 0 & \text{if } \Delta_k[i, j] > P_k \end{cases} \tag{7.5}$$

The individual concordance index measures the strength of the argument that, when comparing solution i to solution j for a given objective criterion k, the value of *' $F_k(i)$ is at least as good as $F_k(j)$ '* when compared to values specified by the decision-maker in the reference range for a given criterion (Roy, 1978). Fig. 2(a) illustrates how the individual concordance index is determined using the values of the calculated differences, the indifference threshold, and the preference threshold. For a difference smaller than the indifference threshold, the corresponding individual concordance index is unity.

Between the indifference and preference thresholds, it varies linearly from 1 to 0. For a difference larger than the preference threshold, the concordance index is set to 0.

3. The weighted sum of individual concordance indices is calculated to determine the global concordance index when comparing solution *i* to solution *j*.

$$C[i, j] = \sum_{k=1}^{n} W_k c_k [i, j] \qquad \begin{cases} i \in [1, M] \\ j \in [1, M] \end{cases} \tag{7.6}$$

4. A discordance index $D_k[i, j]$ is then calculated for each criterion *k* using the preference and veto thresholds:

$$D_k[i, j] = \begin{cases} 0 & \text{if } \Delta_k[i, j] \le P_k \\ \dfrac{\Delta_k[i, j] - P_k}{V_k - P_k} & \text{if } P_k < \Delta_k[i, j] \le V_k \\ 1 & \text{if } \Delta_k[i, j] > V_k \end{cases} \tag{7.7}$$

The discordance index measures the strength of the argument that when comparing solution *i* to solution *j* for a given criterion k the value of $'F_k(i)$ *is significantly worse than* $F_k(j)'$ when compared to values specified by the decision-maker in the reference range for a given criterion (Roy, 1978). Fig. 2(b) illustrates how the discordance index is determined using the preference and veto thresholds. For a difference smaller than the preference threshold, the discordance index is 0. Between the preference and veto thresholds, it varies linearly from 0 to 1, and for a difference larger than the veto threshold, the discordance index is set to 1.

5. Using the global concordance and discordance indices, the relative performance of each pair of domain solutions is finally evaluated by calculating each element of the outranking matrix $\sigma[i, j]$ using the following equation:

$$\sigma[i, j] = C[i, j]\left(\prod_{k=1}^{n} \left[1 - (D_k[i, j])^3 \right] \right) \qquad \begin{cases} i \in [1, M] \\ j \in [1, M] \end{cases} \tag{7.8}$$

Each element $\sigma[i, j]$ measures the quality of solution *i* relative to solution *j* in terms of the *n* objective functions. An element $\sigma[i, j]$ close to 0 indicates that solution *j* outranks solution *i*. If the value is

near 1, then solution *i* may outrank solution *j* or simply be located in the vicinity of solution *j*. In the absence of discordant criteria, the outranking matrix is identical to the global concordance matrix. However, it only takes one discordant criterion to make an element of the outranking matrix equal to zero. The definition of such a relation, called an outranking relation, involves the three thresholds mentioned above, and its function reflects the respective role played by each objective (Roy, 1971).

6. The final ranking score for each solution in the Pareto domain is obtained by summing individual outranking elements associated with each domain solution as follows:

$$\sigma_i = \sum_{j=1}^{M} \sigma[i, j] - \sum_{j=1}^{M} \sigma[j, i] \tag{7.9}$$

The first term evaluates the extent to which element *i* performs relative to all the other solutions in the Pareto domain, while the second term evaluates the performance of all the other solutions relative to solution *i*. The solutions are then sorted from highest to lowest according to the ranking score. The solution with the highest ranking is the one that best satisfies the set of preferences provided by the decision-maker.

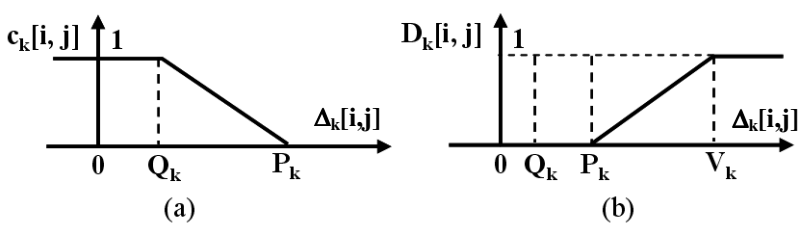


Fig. 7.2 (a) Individual concordance index, and (b) discordance index calculations used in NFM algorithm to determine ranking scores for the Pareto domain solutions.

Finally, instead of relying on the unique solution of the Pareto domain having the best ranking score, it is preferable to use the results of NFM to divide the Pareto domain into zones containing high-ranked, mid-ranked, and low-ranked domain solutions in order to identify

graphically where the optimal region is located. The decision space variables associated with the optimal objective function zone are then implemented in the process.

It may be tempted to believe that the NFM reduces to a simple least-squares method, where only the relative weights (W_k) are used to rank the entire Pareto domain, when the three thresholds (Q_k, P_k, and V_k) are either made all equal to zero or to very high values. This is not the case and, in fact, the three threshold values play an important role in the ranking of the Pareto domain over the whole range of threshold values. The role of thresholds is to use the distance between two values of a given criterion to create a zone of preference around each solution of the Pareto domain and to identify the solutions that are systematically better than the other solutions.

The threshold values could vary over the range of data or the location of Pareto-optimal solutions. For instance, in the pulping process example that was studied by Renaud *et al.* (2007) and Thibault *et al.* (2002), the expert expressed a very stringent tolerance, resulting in small threshold values, when the ISO brightness of the paper was low. It was considered that one percentage point in the lower range of the ISO brightness was critical whereas, in the upper range, its level of tolerance was lax. It is indeed possible to define thresholds which are not constant along the range of the objective function but this would obviously require a greater participation from the decision-maker. Further, functions for the concordance and discordance indices can be easily generalized to consider other functions that take into account uncertainty, conflicts and contradictions of the decision-maker.

Example 7.2 Consider a four-objective optimization problem. The four objectives are denoted as C_1, C_2, C_3, and C_4. It is desired to maximize C_1 and C_3, and minimize C_2 and C_4. Rank the Pareto-optimal solutions contained in Table 7.2 using NFM subject to the ranking parameters of Table 7.3.

Solution Using a pair-wise comparison, the difference $\Delta_k[i, j]$, the individual concordance index $c_k[i, j]$, and the discordance index $D_k[i,j]$ are calculated using Eqs. (7.4), (7.5), and (7.7), respectively. The values

calculated for this example are presented in Table 7.4. From the values calculated in Table 7.4, it is possible to calculate the global concordance index $C[i,j]$ and the outranking matrix $\sigma[i, j]$ using Eqs. (7.6) and (7.8), respectively. Results are presented in Table 7.5.

Table 7.2 List of three Pareto-optimal solutions of Example 7.2.

Solution	Objectives			
	C_1	C_2	C_3	C_4
	Max	Min	Max	Min
1	40	20	5	100
2	31	18	3	200
3	43	23	2	70

Table 7.3 Parameters used to rank the Pareto domain using Net Flow.

Objective	W_i	Q_i	P_i	V_i	Type
1	0.30	2	5	10	Max
2	0.30	2	4	8	Min
3	0.20	0.5	2	4	Max
4	0.20	20	40	80	Min

Table 7.4 Calculated values of $\Delta_k[i, j]$, $c_k[i, j]$, and $D_k[i,j]$ for Example 7.2.

Difference $\Delta_k[i, j]$												
	C_1			C_2			C_3			C_4		
j \ i	1	2	3	1	2	3	1	2	3	1	2	3
1	0	-9	3	0	-2	3	0	-2	-3	0	100	-30
2	9	0	12	2	0	5	2	0	-1	-100	0	-130
3	-3	-12	0	-3	-5	0	3	1	0	30	130	0
Individual concordance index $c_k[i, j]$												
j \ i	1	2	3	1	2	3	1	2	3	1	2	3
1	1	0	1	1	1	0.5	1	0	0	1	0	1
2	1	1	1	1	1	0	1	1	0.667	1	1	1
3	0.667	0	1	1	1	1	1	1	1	0.5	0	1
Discordance index $D_k[i,j]$												
j \ i	1	2	3	1	2	3	1	2	3	1	2	3
1	0	0.522	0	0	0	0	0	0	0.125	0	1	0
2	0	0	0	0	0	0.016	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	1	0

Finally, Eq. (7.9) is used to calculate the score of each solution. It is obtained by subtracting the sum of all elements in the column by the sum of all elements in the row for each diagonal element. The scores are 1.231, -1.623, and 0.392 for solutions 1, 2, and 3, respectively. The solution with the highest score is the best solution. In this example, the ranking in order of preference gives solution 1 as the best solution, followed by solutions 3 and 2. This example was performed with only three solutions. In practice, the same analysis is performed with thousands of Pareto-optimal solutions that are generated to adequately circumscribe the entire Pareto domain.

Table 7.5 Global concordance index $C[i,j]$ and the outranking matrix $\sigma[i, j]$.

	Global concordance index $C[i,j]$			*Outranking matrix* $\sigma[i, j]$		
j \ i	1	2	3	1	2	3
1	1	0.30	0.65	1	0	0.569
2	1	1	0.63	1	1	0.623
3	0.80	0.5	1	0.80	0	1

7.4 Rough Set Method

Rough set theory, introduced by Pawlak (1982; 1991; 2004), utilizes a model of approximate reasoning. It has attracted considerable attention through numerous and diverse applications, maturing to the point where more than 4000 papers have been published on the subject since 1982 (Orlowska *et al.*, 2007). Rough set theory has been used in knowledge discovery in databases (Düntsch and Gediga, 2000), environmental engineering applications (Warren *et al.*, 2004), emulsion polymerization processes (Fonteix *et al.*, 2004), single cell oil production (Muniglia *et al.*, 2004), high yield pulping (Thibault *et al.*, 2003; Renaud *et al.*, 2007), and the optimization of the quality of beer (Vafaeyan *et al.*, 2007).

The rough set concept is a mathematical approach to deal with data imprecision, fuzziness, vagueness, and uncertainty. The theory of rough sets is complex mathematically and often considered to be too obscure for widespread implementation in engineering applications. The purpose of this section is to present rough sets as a MOO method that can be easily understood and implemented. The objective of RSM is to assess

and rank by order of preferences all solutions within the Pareto domain. RSM employs the following procedure as used by Thibault *et al.* (2003) and Renaud *et al.* (2007) for a pulping process application.

A handful of solutions, usually 3 to 7, from different regions of the Pareto domain are selected and presented to a decision-maker who has an intimate knowledge of the process being optimized. The decision-maker is given the task of ordering the small set of solutions from the most preferred to the least preferred. This ranking procedure captures and encapsulates the expert's knowledge of the process, which RSM will use to rank all solutions in the Pareto domain.

Following the creation of the small ranked set of Pareto-optimal solutions, the expert may, although not essential, establish ranges of indifference for each objective, which account for possible error in the measurement of the criteria as well as possible limits of human detection regarding differences for a given criterion. Specifically, the range of indifference for a particular criterion is defined as the difference between two values of the criterion that is not considered significant enough to rank one value over another. This indifference parameter is similar to the indifference threshold of the NFM and was added to generalize RSM. It can obviously be set to zero for any or all criteria. Once the small solution set is ranked and the ranges of indifference are chosen, the expert's input into RSM is complete, apart from the validation, acceptance, and implementation of the resulting optimal solution.

The next step is to establish a set of rules that are based on the expert's ranked set and ranges of indifference. Here, each solution in the ranked set is compared to every other solution in order to define "rules of preference" (P rules) and "rules of non-preference" (NP rules). A rule takes the form of a set of values, one for each criterion, where each value may be either 1 or 0. A value of 1 indicates that the first solution is better than the second with respect to that criterion, while a value of 0 indicates that the first solution is worse than, or not significantly different from, the second with respect to that criterion. If it is desired to include the threshold of indifference in the elaboration of the rules, ranking the second solution in the comparison is always considered in the worst possible light by adding or subtracting the range of indifference, as appropriate, to it. In other words, the first solution in the comparison is

always given the benefit of the doubt with respect to the range of indifference for a given criterion. This implies that, when the threshold of indifference is active, comparing solution one to solution two is not the reciprocal operation of comparing solution two to solution one such that the NP rule is not necessarily the complementary rule of the P rule. When the thresholds of indifference are ignored, the P and NP rules are complementary.

When the entire set of P and NP rules has been established, some rules need to be eliminated for two reasons. First, if two P rules are the same, or if two NP rules are the same, then the rule needs to be included only once since it represents only one type of preference. Second, if a P rule is identical to a NP rule, they are both eliminated since the expert cannot rank one solution better than another solution for the same reason that he considers a solution worse than another solution.

Finally, the solutions that comprise the Pareto domain are ranked using the P and NP rules. Each solution in the Pareto domain is given a ranking score, which starts at zero, and then compared with every other solution in the same manner that the solutions in the small ranked set were compared to one another. For each comparison that matches a P rule, the ranking score of the first solution is incremented by one and the ranking score of the second solution is decremented by one; for each comparison that matches a NP rule, the ranking score of the first solution is decremented by one, and the ranking score of the second solution is incremented by one. Upon completing the ranking, the solution with the highest score represents the optimal solution. Akin to the procedure used in NFM, the Pareto domain can then be divided into zones containing high-ranked, mid-ranked, and low-ranked domain solutions in order to easily identify graphically where the optimal region is located. The selected solution is then validated by the decision-maker.

RSM is able to capture the knowledge that the decision-maker has of his/her process in a very straightforward manner. However, the main limitation of RSM is its reliance on the decision-maker's ranking of a very small set of solutions selected from the Pareto domain. Some of the issues associated with the generation of ranking rules and the subsequent ranking process are summarized below.

The number of solutions presented to the decision-maker should be large enough to determine a sufficient number of rules to adequately rank all solutions in the Pareto domain. If the number of solutions presented to the user is too small, only a fraction of the possible number of rules will be generated. On the other hand, if the number of solutions is too high, some of the rules will be generated more than once with the risk of a given rule appearing in both the P and NP rules, resulting in its eventual elimination. In addition, when a large number of solutions are presented, the decision-maker may be overwhelmed by the amount of data and may not be able to easily rank these solutions. To guide the optimizer in determining the number of solutions to present to the decision-maker, Tables 7.6 and 7.7 present the maximum number of rules associated with a given number of objectives of an optimization problem, and the maximum number of rules that can be generated given the number of solutions in the decision-maker's small ranked data set, respectively.

The process of subtracting two rules in the expression $(2^k - 2)$ in Table 7.6 accounts for the removal of rules (00...00) and (11…11) from the possible rule set, since these two rules imply a solution that suffers complete dominance in the first case and enjoys total domination in the second, and so de facto cannot be part of the Pareto domain. For example, in the case of a four-objective optimization problem, a maximum of 14 rules can be generated. It might seem reasonable in this case to present the decision-maker with a data set containing four solutions, resulting in a minimum of two rules that will not appear in the final set. Meanwhile, if the expert's ranked data set contains five solutions, a total of 20 rules will be generated while the maximum number of possible distinct rules would be only 14, implying that some rules will be duplicated in the P and NP sets, and therefore eliminated.

Table 7.6 Maximum number of rules as a function of the number of objective criteria.

Number of objectives	Maximum number of rules
2	2
3	6
4	14
…	…
k	$2^k - 2$

Table 7.7 Number of rules generated as a function of the number of solutions presented to the decision-maker for ranking.

Number of solutions	Number of rules generated
2	2
3	6
4	12
…	…
m	$2\sum_{i=1}^{m}(i-1)$

By way of illustrating this latter point, for a four-objective optimization problem, Renaud *et al.* (2007) used an expert's ranked data set containing seven solutions taken from the Pareto domain, resulting in a total of 42 rules. Removing the duplicate rules in each of the P and NP rule sets and identical rules appearing simultaneously in both the rule sets, only three rules remained in both rule sets (a total of 6) in one case considered, and five rules in each set (a total of 10) in the other case. Under ideal conditions, seven rules in each of the P and NP rule sets would be found. Obviously, some rules were eliminated because they appeared in both rule sets and, therefore, they will not used to rank the entire Pareto domain.

When the values of a particular criterion associated with two solutions contained in the expert's ranked data set are close to each other, ranking of the two solutions will invariably not be based on this particular criterion. This situation may lead to a rule that will not be significant for that criterion and which may subsequently bias the final ranking process. There exist a few approaches to partly alleviate this problem:

1. To use a threshold of indifference as suggested above (Thibault *et al*., 2003).
2. To adopt a ternary rule (0; 0.5; 1) or (-1; 0; 1) instead of a binary rule (0; 1) to account for the uncertainty created by having two objectives being within the threshold of indifference (Zaras and Thibault, 2007).
3. To select solutions within the Pareto domain that are sufficiently discriminative to the decision-maker to make a clear choice. Vafaeyan *et al*. (2007) developed one such algorithm that ensures the

solution set presented to the expert includes a maximum number of rules, by choosing those solutions with a high discriminative level.

4. To make sure all possible rules, or at least a reasonable number of rules, appear in the final rule set, an alternative approach would be to present the decision-maker with only one pair of solutions at a time, rather than the entire subset of solutions all at once. Since two solutions can generate only two rules (one P rule and one NP rule), each pair of solutions can be selected in such a way to ensure distinct pairs of rules, thus overcoming rule duplication and elimination. This process is continued until the desired number of rules or all possible rules are generated. If only a fraction of all possible rules is desired, the selection process would start with the most frequently-encountered rule and progress towards the least frequent rules occurring in the Pareto domain. In this case, it is important that all criteria of the selected pairs of solutions be sufficiently spaced to allow proper discrimination between the two solutions. By way of example, in Vafaeyan *et al*. (2007), the pair of solutions was selected so that the value of one criterion for one solution was as close as possible to one quarter of the total range of the criterion and the value of the other solution was as close as possible to three quarters of the total range. For a complete set of rules, it is necessary to present a number of pairs of solutions equal to half the value given in Table 7.6, that is $(2^{k-1} - 1)$.
5. To present the decision-maker with a pair of solutions rather than all of the sample solutions at once can certainly help in the determination of the rules. However, the decision-maker may still become overwhelmed when the number of objectives increases. An alternative would be to segment the problem by considering a smaller number of objectives at a time and devise an algorithm to reconstruct the whole set of rules from the partial information.
6. It may not be necessary to present the decision-maker with pairs of solutions that will generate all possible rules, since some rules may safely be assumed to be in the P set and their binary complement in the NP set. For example, for a five-objective optimization problem, rules (11110), (11101), (11011), (10111), and (01111) will certainly be part of the P set, and their counterpart (00001), (00010), (00100),

(01000), and (10000) will undoubtedly be part of the NP set. Logical arguments, done in agreement with the decision-maker, can advantageously help to construct a part of the P and NP rules and thus reduce the amount of work he/she has to perform in ranking a small set of solutions from the Pareto domain. Extending this argument, it can realistically be concluded that, for a three-objective optimization problem, the P rules would contain (110; 101; 011) and the NP rules will contain the complementary rules (001; 010; 100), which means that the intervention of a human expert is only required to validate this choice and it may not be necessary to have him rank a small data set. For a four-objective optimization, the P rules would undoubtedly contain (1110; 1101; 1011; 0111), and the NP rules would normally contain (0001; 0010; 0100; 1000), which would leave only six rules (1100; 0011; 1010; 0101; 1001; 0110) to be decided by the decision-maker.

Expressing preferences of a single decision-maker may be difficult. When several decision-makers participate in the decision process, it is unrealistic to expect all of them to agree on the ranking of the small data set, leading inevitably to a higher degree of fuzziness. At the same time, the observed contradictions may bring out enriched knowledge about the process that will, in the end, lead to a better understanding of the process and selecting a more desirable optimum.

In a pair-wise comparison of two solutions within the Pareto domain, a rule will contain at least one zero and, as a result, a minimum of one objective function will always be sacrificed. RSM cannot be used for a two-objective optimization because the decision-maker will have to make a clear choice between one of the two objective functions and the preference rule can only be (10) or (01). For instance, choosing (01) automatically means that the optimal solution is the lowest possible value of the second objective in the case of a minimization problem, and the highest possible value for a maximization problem. RSM reduces a two-objective problem to a SOO problem. As the number of objectives increases, the overall effect of losing at least one dimension in objective space diminishes significantly.

It may be advantageous to ask the decision-maker to classify the various rules into the P and NP sets rather than examining a series of solutions from the Pareto domain. Since all objectives do not have equal importance, the possibility of adding relative weights to the rules may further improve the ranking of the entire Pareto domain.

Several steps leading to the final ranking of the Pareto domain also contain a certain level of uncertainty. This is certainly true for process modeling and rule determination based on the expert's ranked set of solutions. In the case of NFM, the variability of the relative weight and the three threshold values associated with each objective must also be assessed. It is therefore important for the decision-maker to feel confident that the optimal zone of operation and the resulting objective space are in agreement with his/her knowledge of the process. A sensitivity analysis must be performed on the parameters used in the whole sequence that led to the optimal solution. In chemical engineering, the aspect of controllability and robustness must also be considered. Yanofsky *et al.* (2006) used a technique called the drift group analysis to determine the robustness of the final solution from a ranked Pareto domain. This step offers an additional opportunity to build robustness while considering the natural variability of the different variables and the control system performance. If the final solution lies on the edge of the action space, it may be wise to accept a compromised optimum by displacing the solution away from the edge of the action space to ensure that the objective function remains within the Pareto domain and in a zone that is as optimal as possible.

Example 7.3 Assuming the small set of Pareto-optimal solutions of Table 7.2 were presented to the decision-maker who ranked solution 1 as the best one and then solutions 3 and 2 in order of preference. Determine the resulting set of preference and non-preference rules. The threshold of indifference is not taken into account.

Solution Performing pair-wise comparisons of the three solutions, a total of six rules can be generated: three P and three NP rules, as shown in Table 7.8. Comparing solution 1 to solution 3, since criteria C_1 and C_4 are worst for the better solution (i.e. solution 1), the vector elements

corresponding to these criteria assume a value of zero for the P rule and one for the NP rule. On the other hand, criteria C_2 and C_3 are better for solution 1, and so one will appear in corresponding elements of the P rule and a zero in the NP rule. This procedure is performed for the three possible pair-wise comparisons. However, amongst the six rules generated (Table 7.8), only four rules are distinct. Indeed, two rules were generated twice. In addition, two rules appear simultaneously in the P and NP sets such that they must be eliminated. Only two rules remained: 1011 and 0100. This example shows the importance of properly choosing the small Pareto-optimal solution set that is presented to the decision-maker in order to generate a representative set of rules that can reliably be used to rank the entire Pareto domain.

Table 7.8 Rules generated from Pareto-optimal solution of Example 7.2.

Pair-wise Comparison	Preference Rules				Non-Preference Rules			
	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
1 →3	0	1	1	0	1	0	0	1
1 →2	1	0	1	1	0	1	0	0
3 →2	1	0	0	1	0	1	1	0

7.5 Application: Production of Gluconic Acid

7.5.1 *Definition of the Case Study*

NFM and RSM are used in this case study to optimize the gluconic acid production. The primary objectives of this process are to maximize simultaneously the overall production rate and the final concentration of gluconic acid. The simulation of the fermentation of glucose to gluconic acid by the micro-organism *Pseudomonas ovalis* in a batch stirred tank reactor is performed. The overall biochemical reaction can be expressed as:

$$\text{Cells} + \text{Glucose} + \text{Oxygen} \longrightarrow \text{More cells}$$
$$\text{Glucose} + \text{Oxygen} \xrightarrow{\text{Cells}} \text{Gluconolactone}$$
$$\text{Gluconolactone} + \text{Water} \longrightarrow \text{Gluconic Acid}$$

This process has been thoroughly studied and the following state space model has been derived to represent, respectively, the concentrations of cells (X), gluconic acid (p), gluconolactone (l), glucose substrate (S) and dissolved oxygen (C) (Ghose and Ghosh, 1976):

$$\frac{dX}{dt} = \mu_m \frac{SC}{k_s C + k_0 S + SC} X \tag{7.10}$$

$$\frac{dP}{dt} = k_P l \tag{7.11}$$

$$\frac{dl}{dt} = v_l \frac{S}{k_l + S} X - 0.91 k_P l \tag{7.12}$$

$$\frac{dS}{dt} = -\frac{1}{Y_s} \mu_m \frac{SC}{k_s C + k_0 S + SC} X - 1.011 v_l \frac{S}{k_l + S} X \tag{7.13}$$

$$\frac{dC}{dt} = K_L a \left(C^* - C\right) - \frac{1}{Y_0} \mu_m \frac{SC}{k_s C + k_0 S + SC} X - 0.09 v_l \frac{S}{k_l + S} X \tag{7.14}$$

In this example, the values of the coefficients used to perform the simulation were identical to those used by Johansen and Foss (1995), and are given in Table 7.9. For the purpose of optimization, the batch fermentation process can be considered as a four-input and multi-output system. The four input or decision variables are the duration of the batch fermentation (t_B), the initial substrate concentration (S_0), the overall oxygen mass transfer coefficient ($K_L a$), and the initial biomass concentration (X_0). While numerous objectives could be defined, this investigation focuses initially on the following two process outputs: the overall productivity of gluconic acid, defined by the ratio of the final gluconic acid concentration over the duration of the batch (P_f/t_B), and the final gluconic acid concentration (P_f). To consider three- and four-objective optimization problems, two other objective functions will later be incorporated: the final substrate concentration (S_f) and the initial biomass concentration (X_0). Then, the optimization seeks to maximize both the overall production rate and the final concentration of gluconic acid, and to minimize the residual substrate concentration and initial biomass concentration. This case study will be done progressively, starting with a two-objective optimization and adding in turn the last two

objectives. Fig. 7.3 presents a schematic of the optimization process when all four objective functions are included.

Table 7.9 Values of parameters for the simulation of gluconic acid production.

Parameter	Value	Unit
μ_m	0.39	h^{-1}
k_s	2.50	g/L
k_0	0.00055	g/L
k_P	0.645	h^{-1}
v_l	8.30	mg/UOD h
k_l	12.80	g/L
Y_s	0.375	UOD/mg
Y_0	0.890	UOD/mg
C^*	0.00685	g/L

In the first step, the Pareto domain is approximated by a large number of feasible solutions. To establish this domain the following procedure was used. For each of the four input variables, a value is randomly selected from within their predefined range of variation ($t_B \in$ [5-15 h]; $S_0 \in$ [20-50 g/L]; $K_La \in$ [50-300 h^{-1}]; $X_0 \in$ [0.05-1.0 UOD/mL]). The model is then solved for this set of input variables to generate the two, three or four objective functions. A diploid genetic algorithm was used to generate a large number of solutions, 12000 in this case study, to adequately circumscribe the Pareto domain (Fonteix *et al.*, 1995; Perrin *et al.*, 1997).

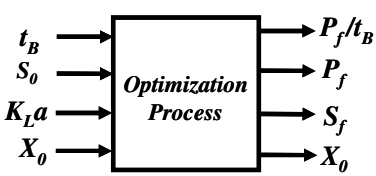


Fig. 7.3 Schematic of the optimization of the gluconic acid production.

7.5.2 *Net Flow Method*

The Pareto domain will first be ranked with the NFM. NFM uses four parameters for each objective criterion to express the preferences of the decision-maker: the relative weight and three thresholds (indifference, preference, and veto). Table 7.10 provides, for each objective function

and for the two-, three- and four-objective optimization problems, values of the relative weights and the three thresholds used in this case study.

Table 7.10 Values of the four optimization parameters for each objective function.

Criterion k	Relative Weight W_k # Objective Functions			Threshold Values		
	2	3	4	Q_k	P_k	V_k
1 (P_f/t_B)	0.50	0.40	0.40	0.1	0.2	0.6
2 (P_f)	0.50	0.40	0.40	0.5	1.0	1.5
3 (S_f)	-	0.20	0.10	0.01	0.1	0.2
4 (X_0)	-	-	0.10	0.05	0.15	0.3

Fig. 7.4 shows the ranked Pareto domain for the two-objective optimization of gluconic acid. This plot of P_f/t_B versus P_f is a typical Pareto domain for a two-objective optimization where both objective functions need to be maximized. It may be tempted to believe that maximizing P_f and minimizing t_B would be equivalent to this two-objective problem. However, this is not the case. Indeed, a very different Pareto domain, which would include very low and very high values of the batch time, would be obtained. Using the productivity (P_f/t_B) is truly the best way to define the desired objective.

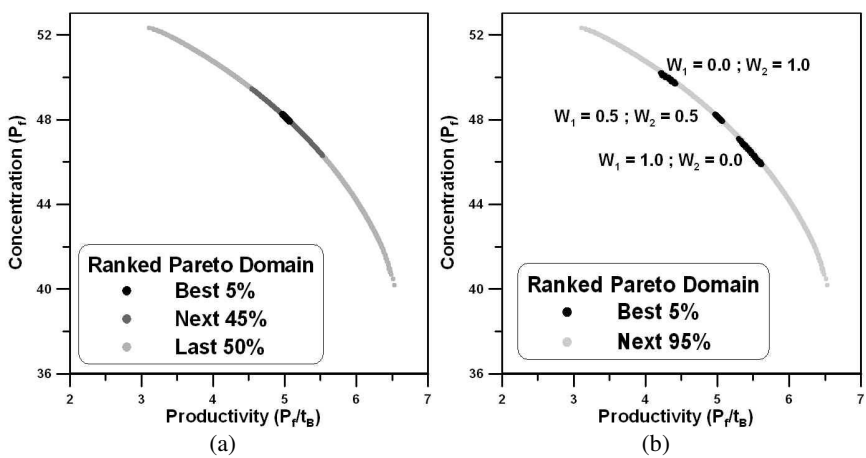


Fig. 7.4 Ranked Pareto domain using NFM for the two-objective optimization of gluconic acid: (a) Nominal case; (b) Influence of the relative weights of the two objective functions.

The optimal values of the gluconic acid productivity and concentration are in the vicinity of 4.7 g/L.h and 49 g/L, respectively. As Fig. 7.4 suggests, the objectives seeking to maximize P_f / t_B and P_f are contradictory; increasing the latter (total production of gluconic acid in a batch) generally results in a lower productivity, due to longer batch fermentation time that would be required. The highest ranked solutions (best 5% plotted using dark symbols) represent a good compromise between the two objective functions. Fig. 7.4(a) shows the ranking of the Pareto domain using the parameters of Table 7.10. Fig. 7.4(b) shows the influence of the relative weights on the ranking of the best 5% of all solutions of the Pareto domain. With a relative weight of unity for one objective and zero for the other, one could expect that the optimal zone would shift along the Pareto domain towards the extreme of the criterion. However, this does not occur to the full extent that would be expected because, even though the relative weight for one criterion and its resulting contribution to the concordance index are both zero, the preference and veto thresholds still play the same role with respect to the discordance index and, therefore, will affect the ranking of a solution with respect to another when, in a pair-wise comparison, the difference of a given criterion exceeds the preference or the veto thresholds. These results clearly show that the NFM is significantly robust to changes in the relative weights, and the thresholds play a very important role in the ranking of the Pareto domain. Additional tests (not shown) were performed to investigate the influence of the threshold values on the ranking of the Pareto domain. The three threshold values of one objective criterion were all set to zero while they were kept at their nominal values for the other criterion. It was observed that decreasing the threshold values to zero for one objective moves the optimal (best 5%) zone along the Pareto front towards higher values of that criterion akin to increasing the relative weight.

To achieve the optimal compromised solution, the four input variables must be set at the values that have given rise to the optimal region of the Pareto domain. As expected, to maximize the concentration and productivity of gluconic acid, it was found that the initial concentrations of both the initial substrate and initial biomass must be at their maximum values, i.e. 50 g/L and 1.0 UOD/mL, respectively.

Fig. 7.5 presents the values of the other two input variables, t_B and K_La, that led to the Pareto domain. It is clear that the performance of the fermentation is sensitive to the duration of fermentation and the overall oxygen mass transfer coefficient. In this case, the optimum fermentation conditions favor somewhat lower values for K_La (between 80 to 100 h^{-1}) while the least favored conditions share values near the upper limit of 240 h^{-1}. The corresponding optimal batch time (t_B) is around 10-12 h. The results of Fig. 7.5 clearly show the trade-off that must exist between t_B and K_La to remain inside the Pareto domain, and illustrate the advantage of using MOO techniques that are based on the Pareto domain. It is possible to observe without ambiguity the trade-offs that must be made to always remain within the Pareto domain. Indeed, any operating conditions that lie outside the zone shown on Fig. 7.5 would lead to a dominated point and would therefore be worse than all solutions that are contained in the Pareto domain. In practice, K_La would be controlled by manipulating the agitation speed of the mixer and/or the air flow rate.

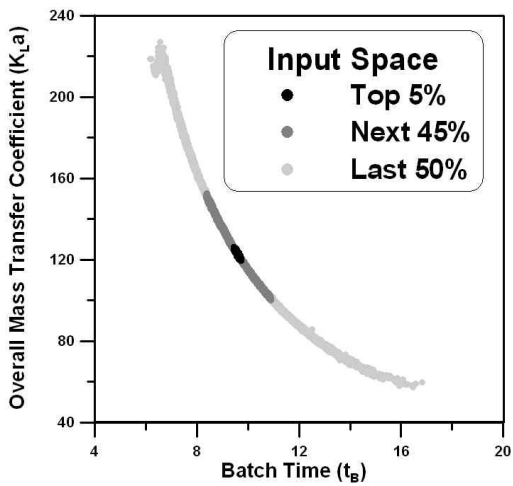


Fig. 7.5 Plot of the input space for the two-objective optimization problem associated with the Pareto domain: K_La versus t_B.

In this fermentation process, in addition maximizing the productivity and the concentration of gluconic acid, it may also be desired to include a third objective function whereby the final substrate concentration would

be minimized in order to fully use the available substrate and to eliminate post-fermentation separation. The two-dimensional projections of the ranked Pareto domain for this three-objective optimization problem are presented in Fig. 7.6. Adding the final substrate concentration did not change significantly the outer boundary of the compromise between the productivity and concentration of gluconic acid. However, due to the domination constraint for the final substrate concentration, the projected Pareto domain extends inward significantly. This behavior is more pronounced at low productivity and higher concentration because, to obtain very low final substrate concentration, it is necessary to conduct the batch fermentation over a longer period of time, thus reducing the gluconic acid productivity.

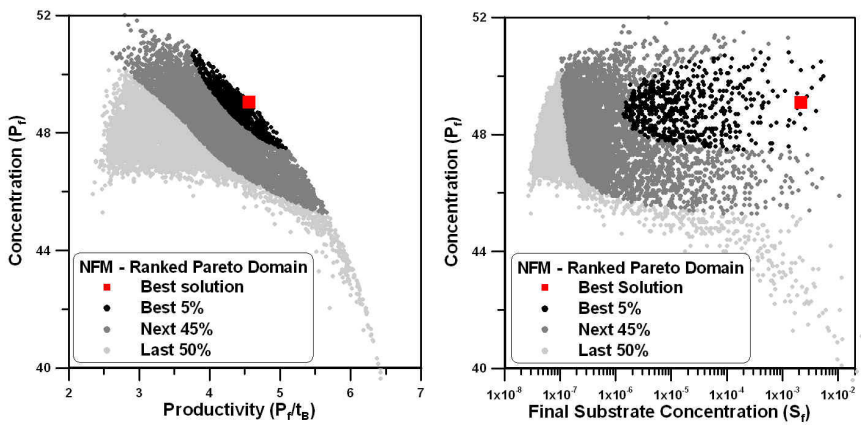


Fig. 7.6 Ranked Pareto domain for the three-objective optimization of gluconic acid using NFM: (a) concentration versus productivity of gluconic acid; (b) concentration of gluconic acid versus final substrate concentration.

Using the final substrate concentration is undoubtedly excessive in this case study because, by forcing the highest gluconic acid productivity and concentration as in the two-objective optimization problem, a small final substrate concentration was also obtained. It was added in this case study to explore the influence of this objective function on the performance of NFM to find a judicious compromise for a three-objective optimization problem. The location of the best 5% of the

ranked Pareto domain for the compromise between productivity and concentration (Fig. 7.6(a)) is nearly the same as the optimal region found for the two-objective optimization (Fig. 7.4(a)). The best solution is also shown on Fig. 7.6. NFM was able to find an excellent compromise given that the first two criteria are the most important ones. Fig. 7.6(b) shows that the third objective function was well satisfied. Indeed, very low values of the final substrate concentration were obtained for the entire Pareto domain. Even though very low final substrate concentration was always obtained, the optimal region is located towards higher substrate concentration in order to reduce the batch fermentation time and favor higher productivity. The final substrate concentration for the best solution is close to its maximum value.

Fig. 7.7 presents two of the associated input variables, the batch time and the overall oxygen mass transfer coefficient, that gave rise to the Pareto domain. The input space corresponding to the best 5% is located at nearly the same position as the one that was obtained for the two-objective optimization, except that it has been slightly inflated. Akin to the two-objective optimization problem, it was found that the initial concentrations of both the initial substrate and initial biomass are very close to their maximum values (50 g/L and 1.0 UOD/mL, respectively).

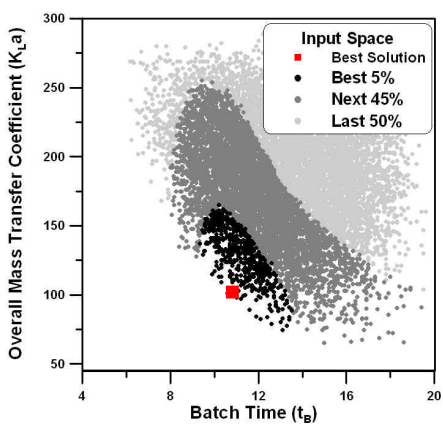


Fig. 7.7 Plot of the input space for the three-objective optimization problem associated with the Pareto domain: K_La versus t_B.

It was observed from the previous results that the initial biomass and substrate concentrations had to be at their maximum allowable values in order to achieve the better performance. It could be envisaged, due to cost consideration, to incorporate the initial biomass concentration X_0 as a fourth objective function even though it also appears as an input to optimization of the process. The ranked Pareto domain obtained for this four-objective optimization problem is presented in Fig. 7.8. The shape of the Pareto domain has drastically changed because of the addition of the initial inoculum biomass concentration. Since the inoculum is minimized, the Pareto domain contains solutions over the entire range of initial inoculum concentration X_0, and it has therefore been greatly expanded. The best 5% of the Pareto-optimal solutions moved inward with respect to the first two objectives, and are located around a concentration of 47 g/L and a productivity of 4.2 g/L.h (Fig. 7.8(a)).

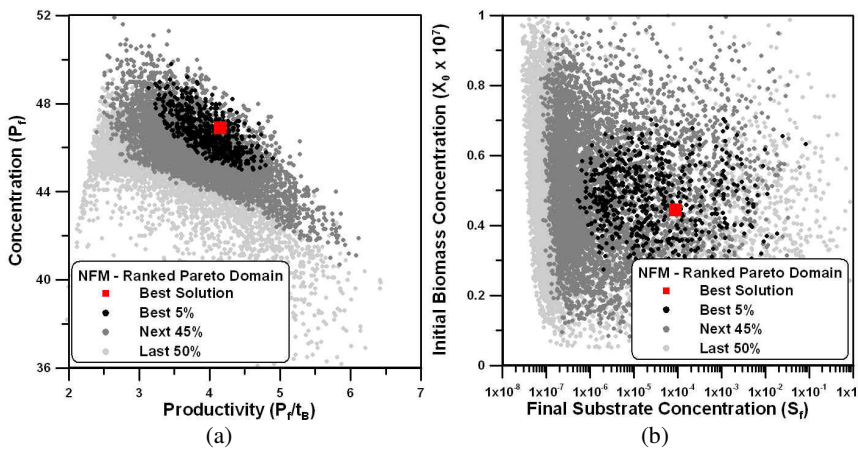


Fig. 7.8 Ranked Pareto domain for the four-objective optimization of gluconic acid using NFM: (a) concentration versus productivity of gluconic acid; (b) Initial inoculum biomass concentration versus final substrate concentration.

Fig. 7.8(b) shows that the initial biomass concentration is around 0.45 UOD/mL and the final substrate concentration was still maintained at very low values. Using a lower initial inoculum biomass concentration naturally led to a small decrease of the first two and, most important, objectives compared to the two-objective optimization problem. NFM

was able to find a suitable compromise between the four objectives. However, in practice, it is desired to use the minimum number of objectives, and the initial biomass concentration would only be included as an objective function if the cost of preparing the inoculum was very high. The plot of the batch time and the overall oxygen mass transfer coefficient associated with the four-objective Pareto domain is presented in Fig. 7.9. The zone associated with the best 5% of all Pareto-optimal solutions is much larger than the two- and three-objective optimization problems considered earlier.

The optimization problem can also be transformed by adding new input space variables. For instance, Halsall-Whitney *et al.* (2003) used NFM for the three-objective optimization problem discussed above but, instead of using a constant overall oxygen mass transfer coefficient (K_La) throughout the fermentation, they used a series of step changes in K_La to optimize the fermentation process with the goal of approximating the overall oxygen mass transfer coefficient profile throughout the fermentation process.

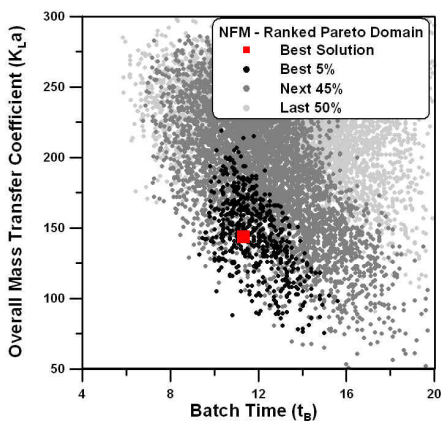


Fig. 7.9 Plot of the input space for the four-objective Pareto domain: K_La versus t_B.

7.5.3 *Rough Set Method*

The identical Pareto domains that were ranked with NFM are now ranked with RSM. As noted above, RSM cannot be used for a two-

objective optimization problem. For the three-objective optimization problem without using a threshold of indifference, the P and NP rule sets used to rank the Pareto domain are presented in Table 7.11. As expected, the rules of preference contain elements that favor two of the three objectives. The selection of a P rule that contains two zeros would mean that one of the objectives is clearly preferred over the other two, and it would logically be possible to reduce the problem to a one-objective optimization problem. It is however possible to choose a set of rules that does not contain all the elements.

The ranked Pareto domain, obtained using the set of rules of Table 7.11, is presented in Fig. 7.10. The nice compromise that was observed for the three-objective optimization using NFM (Fig. 7.6) is not achieved with RSM. The best 5% amongst the ranked Pareto domain spans over a very wide range of gluconic acid productivity and concentration. It is not possible to clearly identify narrow ranges similar to those obtained with NFM. The best solution (see Fig. 7.10) is obtained at nearly the highest possible gluconic acid concentration and, as a consequence, at a very low productivity. The reason for this must be examined taking into consideration the third objective function that needs to be satisfied. The third objective function, the final substrate concentration, was very well satisfied as it is close to its lowest possible concentration. It can be concluded that two objectives (gluconic acid and final substrate concentrations) are very close to their optimum values but the productivity was sacrificed. Indeed, in a pair-wise comparison of the best ranked solution with all the other solutions in the Pareto domain, the rule (011) predominates with a percentage occurrence of 80%. This rule is also the most frequent rule encountered in all pair-wise comparisons with a percentage of more than 25% (Table 7.11).

Table 7.11 P and NP rule sets for the three-objective optimization.

Preference		Non-preference	
Rule	% Occurrences	Rule	% Occurrences
1 1 0	18.99	0 0 1	17.41
1 0 1	7.38	0 1 0	7.14
0 1 1	25.25	1 0 0	23.83

Fig. 7.11 presents the graph of the overall mass transfer coefficient (K_La) as a function of the batch time (t_B) associated with the results of Fig. 7.10. The distribution of the input space is significantly different from the results that were obtained with NFM (Fig. 7.7) where a significantly larger batch time is required and slightly higher oxygen mass transfer coefficient. The other two variables (S_0 and X_0) remained nearly identical, at their upper bounds.

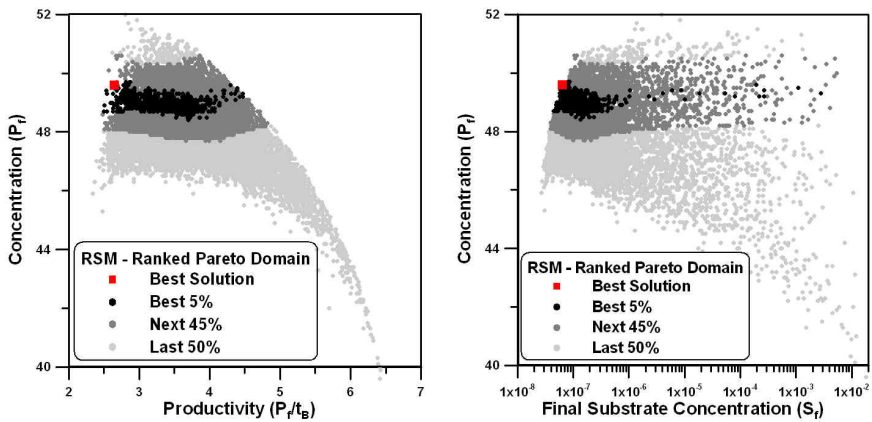


Fig. 7.10 Ranked Pareto domain for the three-objective optimization with RSM using the P and NP rules of Table 7.11: (a) concentration versus productivity of gluconic acid; (b) gluconic acid concentration versus final substrate concentration.

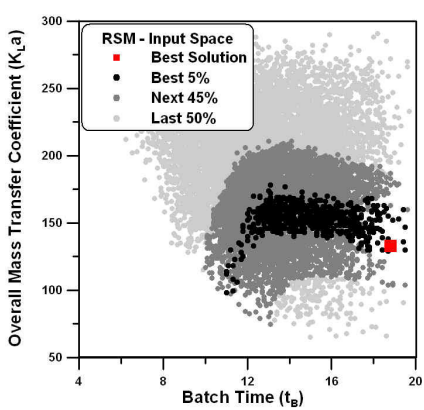


Fig. 7.11 Plot of the input space for the three-objective Pareto domain using RSM: K_La versus t_B.

In NFM, the various thresholds were able to reduce the importance of a given solution, even vetoed a solution relative to another, when the difference between the values of a given objective increased. RSM is not capable to make such a subtle compromise. In fact, the contrary is observed as in the case of the best identified solution. The best solution is located nearly at the extremes of all criteria: very low productivity, high gluconic acid concentration, and very low final substrate concentration. The ranking of RSM favors solutions with extreme criteria such that a given rule will prevail for the majority of the pair-wise comparisons, and the best compromise is achieved, akin to what was discussed for a two-objective optimization, by selecting a combination of the best and worst objectives. NFM penalizes excessive differences between values of objectives whereas RSM rewards large differences by favoring objectives located close to their extreme ranges. The latter ensures that some objectives will be very well satisfied while completely sacrificing the other objectives.

With RSM, each criterion has equal importance as opposed to NFM for which the relative weights partly account for the importance of the objectives. To account for the relative importance of each objective using RSM, a few methods can be proposed. A simple method is to remove some of the rules to favor one criterion over another one. In this particular case, the gluconic acid productivity was drastically sacrificed in favor of the other two objectives. One way to place a greater emphasis on that important criterion is to remove P rule (011) and NP rule (100) from the set of rules of Table 7.11. These two rules accounted for roughly 50% of all rules within the Pareto domain. The P rule and its complementary NP rule significantly reward a solution which has a lower productivity and high values for the other two objectives, and to penalize the productivity when it was high while the other two criteria were low. Results obtained following the elimination of these two rules are presented in Figs. 7.12 and 7.13. These results are now very similar to those obtained by NFM (Figs. 7.6 and 7.7). In this case study, there was too much emphasis placed on the final substrate concentration. Indeed, the presence of zero or one in the establishment of a rule was decided based on a very small, almost negligible, final substrate concentration. Eliminating only rule (100) from the NP rule set gave

results (not shown) that were intermediate between the results of Figs. 7.10 and 7.11, and those of Figs. 7.12 and 7.13. Removing some rules to favor one criterion is thus one of the methods but a task that may be more difficult for the decision-maker and may induce unwanted biases.

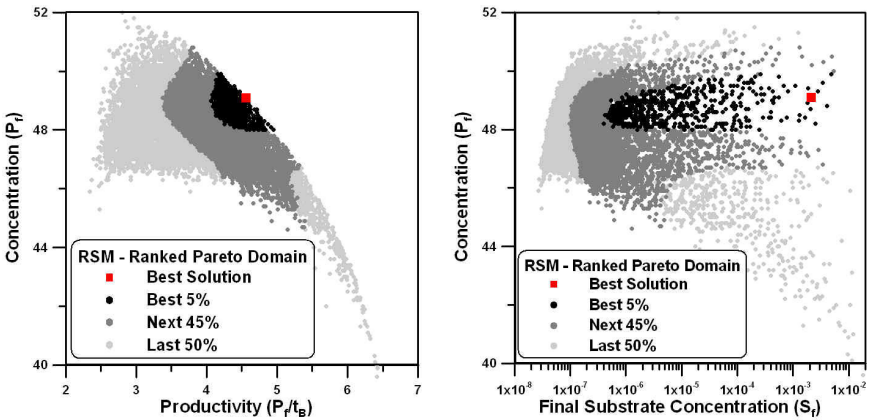


Fig. 7.12 Ranked Pareto domain for the three-objective optimization of gluconic acid using RSM without rules (011) and (100): (a) concentration versus productivity of gluconic acid; (b) gluconic acid concentration versus final substrate concentration.

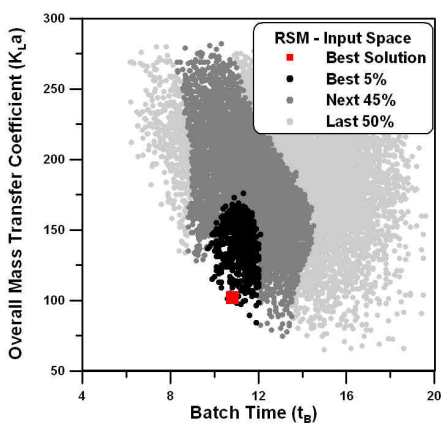


Fig. 7.13 Plot of the input space for the four-objective Pareto domain using RSM without rules (011) and (100): K_La versus t_B.

An alternative method to place more emphasis on some objective functions is to attribute relative weights to each of them and use these

weights in the calculation of the scores of each solution of the Pareto domain. Currently, each solution within the Pareto domain is compared with every other solution and the rule that exists between two solutions is determined. For each comparison that matches a P rule, the ranking score (initially set to zero) of the first solution is incremented by one and the ranking score of the second solution is decremented by one; for each comparison that matches a NP rule, the ranking score of the first solution is decremented by one, and the ranking score of the second solution is incremented by one. Upon completing the ranking, the solution with the highest score represents the optimal solution.

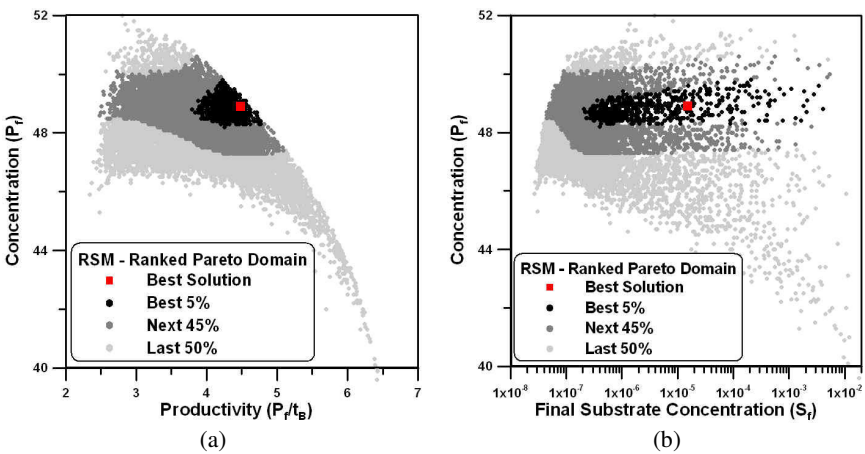


Fig. 7.14 Ranked Pareto domain for the three-objective optimization of gluconic acid using RSM with relative weights: (a) concentration versus productivity of gluconic acid; (b) gluconic acid concentration versus the final substrate concentration.

To include the relative weights in the ranking of the Pareto domain, it is proposed to add the contribution of relative weights as follows. If a P rule is matched, the score would be incremented by an amount equal to the weighted sum of each element of the rule. For example, matching P rule (110), the score of the solution would be incremented by ($1*W_1$ + $1*W_2$ + $0*W_3$) and the score of the other solution, normally satisfying rule (001), would be decremented by the same value. If a NP rule is matched, the score would be decremented by an amount equal to the weighted sum of each element of the complementary rule. For example,

matching non-preference rule (010), the score of the solution would be decremented by ($1*W_1 + 0*W_2 + 1*W_3$) and the score of the other solution, normally satisfying rule (101), would be incremented by the same value. The results obtained using the relative weights of Table 7.10 in the calculation of the scores of each solution in the Pareto domain are presented in Figs. 7.14 and 7.15. These results are now very similar to those obtained by NFM (Figs. 7.6 and 7.7) and those obtained by RSM when two rules were removed from the rule sets (Figs. 7.12 and 7.13).

The weighted RSM provides very good results for the three-objective optimization problem, and offers a much simpler way to the decision-maker to consider the relative importance of each objective function. The weighting of each objective allows moving from the crisp ranking of the nominal RSM to a method that is able to make greater compromise.

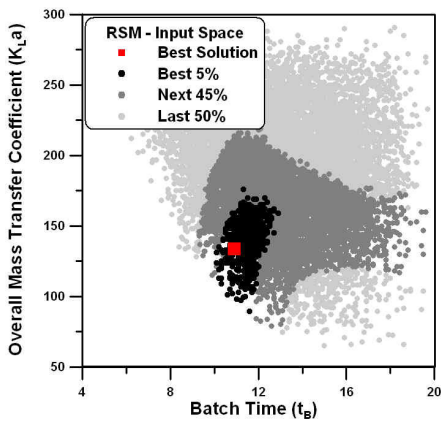


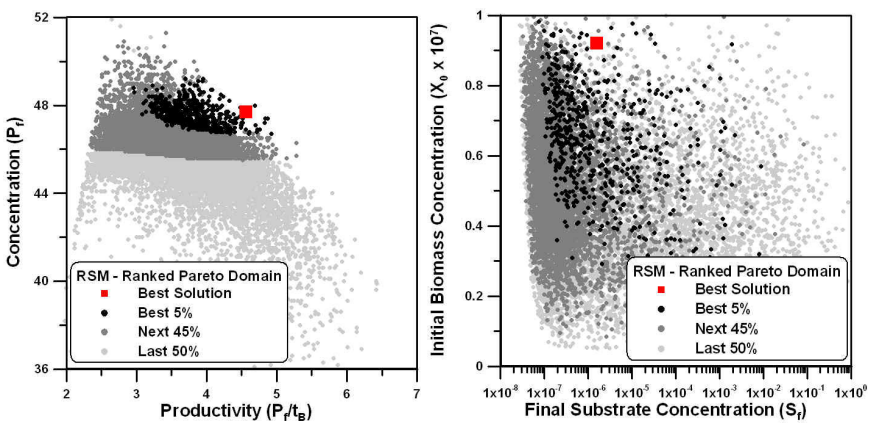
Fig. 7.15 Plot of the input space for the three-objective Pareto domain using RSM with relative weights: K_La versus t_B.

A similar analysis using RSM was performed for the four-objective optimization problem where the initial inoculum concentration was added to the list of objective functions. The P and NP sets of rules that were used to rank the entire Pareto domain are presented in Table 7.12. The results obtained with the 14 rules without considering relative weights were very similar to those obtained for the three-objective optimization problem (Figs. 7.10 and 7.11). The best 5% of all solutions

for the plot of the productivity and concentration of gluconic acid was almost identical for the first three objectives where the gluconic acid productivity was again sacrificed in favor, this time, of the other three objectives. In addition, the best solution for the fourth objective, the initial inoculum concentration, was at its maximum permissible concentration.

Table 7.12 P and NP rule sets for the four-objective optimization.

Preference		Non-preference	
Rule	% Occurrences	Rule	% Occurrences
0 1 1 1	12.32	1 0 0 0	11.78
1 1 1 0	6.09	0 0 0 1	7.14
0 1 1 0	16.60	1 0 0 1	16.40
0 1 0 1	3.61	1 0 1 0	3.43
1 1 0 0	7.98	0 0 1 1	7.51
1 0 1 1	0.86	0 1 0 0	0.82
1 1 0 1	3.67	0 0 1 0	3.23



(a) (b)

Fig. 7.16 Ranked Pareto domain for the four-objective optimization of gluconic acid using RSM with relative weights: (a) concentration versus productivity of gluconic acid; (b) initial biomass concentration versus final substrate concentration.

Three objectives were very well satisfied but, with RSM, one objective always appears to be sacrificed. The optimization was then performed using RSM with the relative weights of Table 7.10 and the

P and NP rules of Table 7.12. The results of the ranked Pareto domain and the associated input space are presented in Figs. 7.16 and 7.17, respectively. The results are similar to those obtained with NFM (Figs. 7.8 and 7.9). It is interesting to note that the best solution achieved a very good compromise amongst all objectives. The first three objectives were better satisfied with the weighted RSM than with NFM; only the initial inoculum concentration was worst. The associated input space was also very similar to the one obtained with NFM.

Similar results were also obtained when some of the rules were eliminated. However, it is much easier from the standpoint of view of the decision-maker to specify relative weights of each objective than to eliminate or interchange some rules. The weighted RSM was found to be relatively robust and achieved systematically, as was the case for NFM, an excellent compromise.

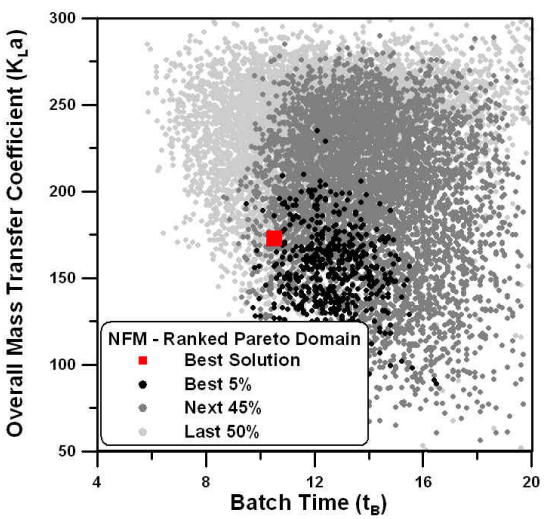


Fig. 7.17 Plot of the input space for the four-objective Pareto domain using RSM with relative weights: K_La versus t_B.

Table 7.13 presents a summary of the main results obtained for the MOO of the production of gluconic acid using NFM and RSM. The results obtained by the two methods, except RSM in its nominal form, are very similar. The two modifications that were implemented for RSM,

which are the elimination of some rules from the P and NP sets, and the addition of relative weights in calculating the ranking scores of RSM, gave excellent results. Table 7.13 presents the best solution for all cases studied. Prior to implementing this solution, it is important to examine the location of the associated process input variables to ensure that it will lead to a robust solution despite natural disturbances that can affect the process. This subject was previously discussed by Yanofsky *et al.* (2006).

Consider for example the results of the two-objective optimization problem (Figs. 7.4 and 7.5). There exists an intimate and relatively stringent relationship between the batch time (t_B) and the overall oxygen mass transfer coefficient (K_La). Deviating slightly from the set point of these variables may bring the operating conditions outside the input space that gives rise to Pareto-optimal solutions. As a result, these new operating conditions would lead to a dominated solution, and therefore to a solution that would be worse than any solution within the Pareto domain. Process control in this case will need to be designed carefully to ensure to remain within the Pareto domain especially that K_La is not easily measured in practice and can only be controlled indirectly by manipulating the speed of rotation of the agitator, varying the oxygen concentration of the input gas, and/or changing the input gas flow rate.

Extending this analysis to the three-objective optimization problem, it can be observed in Fig. 7.7 that the operating point associated with the best Pareto-optimal solution is located on the edge of the input space delimitated by t_B and K_La. Again, a slight decrease in the batch time or the overall oxygen mass transfer coefficient will lead to a dominated solution, and therefore, a solution that would be worse than any other solution within the Pareto domain. The decision-maker will undoubtedly accept a slightly suboptimal solution and choose to increase K_La in order to remain inside the central zone of the best 5% despite outside disturbances and to ensure more robust operating conditions. This unambiguously demonstrates the strength of using optimization methods that are based on Pareto domain because it is possible to directly visualize the trade-offs that must be made when implementing the results of an optimization study. It is almost tempting to argue that, for a low number of objectives, the decision-maker could perform an optimization

graphically by selecting a solution that would meet his/her preferences and see the impact on all variables simultaneously.

Table 7.13 Summary of the MOO of gluconic acid production – best solutions.

Number of Objectives	Method	Input Process Variables				Objective Functions		
		t_B (h)	S_0 (g/L)	K_La (h^{-1})	X_0 (*UOD/ml*)	P_f/t_B (g/L.h)	P_f (g/L)	S_f (g/L)
2	NFM	10.8	49.9	102	0.99	4.57	49.3	-
	RSM	-	-	-	-	-	-	-
3	NFM	10.8	49.7	102	0.98	4.56	49.1	2.1×10^{-3}
	RSM	18.8	49.8	133	0.95	2.65	49.6	6.4×10^{-8}
	RSM-E[1]	10.8	49.7	102	0.98	4.56	49.1	2.1×10^{-3}
	RSM-W[2]	10.9	49.8	134	0.97	4.48	48.9	1.5×10^{-5}
4	NFM	11.3	49.9	144	0.45	4.15	46.9	9.8×10^{-5}
	RSM	17.5	49.9	160	0.996	2.79	49.0	5.7×10^{-8}
	RSM-W[2]	10.5	49.8	173	0.92	4.56	47.7	1.5×10^{-6}

[1]Rules (011) and (100) were eliminated from rule set.
[2]RSM using weighted rules.

7.6 Conclusions

This chapter has presented two MOO methods that are useful to rank the Pareto domain. These two methods are able to capture and encapsulate within the optimization procedure itself the knowledge that the decision-maker possesses on his/her process. The first method, Net Flow, is based on an outranking relation that is implemented in the form of pair-wise comparisons of Pareto-optimal solutions. The ranking of all solutions contained in the Pareto domain is achieved on the basis of concordance and discordance indices defined in terms of relative weights and three thresholds (indifference, preference and veto). The second method, Rough Sets, is based on the determination of a set of preference and non-preference rules that are derived from a small representative Pareto-optimal solution set that is ranked by the decision-maker. A set of rules is then used to rank the entire Pareto domain. These methods, which have been primarily used up to now in the field of Operations Research, are gaining considerable interest in engineering applications.

The two optimization methods were applied using the batch simulation model of the production of gluconic. The optimization was formulated as two-, three- and four-objective problems. NFM has been shown to be very robust in all cases. RSM cannot be implemented for a two-objective problem but is easily used for problems of higher dimension. RSM provides crisp ranking of the Pareto domain, especially when the number of objectives is low. Some RSM variants having a smoother ranking were discussed and were shown to be equivalent to NFM.

Acknowledgements

The author would like to thank the Natural Science and Engineering Research Council (NSERC) for the financial assistance.

Nomenclature

C	Dissolved oxygen concentration in the broth, $g.L^{-1}$
c_k	Individual concordance index
C_k	Global concordance index or individual objective
C^*	Equilibrium liquid oxygen concentration, $g.L^{-1}$
D_k	Discordance index
$f(X)$	Individual objective function
$F(X)$	Multi-objective function space
G	Inequality constraint
H	Equality constraint
(i, j)	Indices for pair of solutions
k	Criterion number
K_La	Volumetric oxygen transfer coefficient, h^{-1}
k_l	Michaelis constant for lactone production, $g.L^{-1}$
k_P	Gluconolactone hydrolysis rate constant, h^{-1}
k_o	Monod rate constant of growth with respect to oxygen, $g.L^{-1}$
k_s	Monod rate constant of growth with respect to glucose, $g.L^{-1}$
l	Gluconolactone concentration, $g.L^{-1}$
n	Number of objectives or criteria
M	Number of solutions approximating the Pareto domain
N	Number of input variables

P	Gluconic acid concentration, $g.L^{-1}$
P_f	Final Gluconic acid concentration, $g.L^{-1}$
P_f/t_B	Gluconic acid productivity, $g.L^{-1}.h^{-1}$
P_k	Preference threshold
Q_k	Indifference threshold
S	Substrate concentration, $g.L^{-1}$
S_o	Initial substrate concentration, $g.L^{-1}$
S_f	Final substrate concentration, $g.L^{-1}$
t_B	Batch time, h
t	Time, h
v_l	Velocity constant for lactone production, $mg.UOD^{-1}.h^{-1}$
V_k	Veto threshold
W_k	Relative weight
x	Individual decision variable or action
X	Vector of decision variables or cell concentration, $UOD.ml^{-1}$
X_o	Initial cell concentration, $UOD.ml^{-1}$
Y_o	Yield of growth based on oxygen, $UOD.mg^{-1}$
Y_s	Yield of growth based on glucose, $UOD.mg^{-1}$

Greek Characters

Δ	Difference between values of a criterion for a pair of solutions
σ	Outranking matrix
μ_m	Maximum specific growth rate, h^{-1}

References

Bhaskar, V., Gupta, S. K., and Ray, A. K. (2000). Applications of multiobjective optimization in chemical engineering. Reviews in Chem. Eng., 16, 1-54.

Brans, J.P., Mareschal, B., and Vincke, Ph. (1984). PROMETHEE: A new family of outranking methods in multicriteria decision making, *Operations Research*, 84, 477-498.

Brans, J.P., Vincke, Ph., and Mareschal, B. (1986). How to select and how to rank projects: The PROMETHEE method. European J. Oper. Research, 24, 2228-238.

Coello Coello, C.A. (1999). A comprehensive survey of evolutionary-based multi-objective optimization techniques, Knowledge Information Systems, 1: 269-308.

Couroux, P., Qannar, E.M., Melcion, J.P., and Morat, J. (1995). Optimisation multiréponse: Application à un procédé de granulation d'aliments, Récents Progrès en Génie des Procédés: Stratégie Expérimentale et Procédés Biotechnologiques, 36 : (9) 41-47.

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms, Wiley, NY.

Derot, B., Gareau, J., Kiss, L.N., and Martel, J.M. (1997). The solver of Volvox multicriteria table. Multicriteria Analysis, Climaco J., (ed) Springer-Verlag, 113-126.

Doumpos, M., and Zopounidis C. (2002). Multicriteria decision aid classification methods, Applied Optimization, Vol. 3, Kluwer Academic, 2002.

Düntsch, I., and Gedida, G. (2000). Rough sets data analysis: A road to non-invasive knowledge discovery. Methoδos Primers, Vol. 2.

Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multi-objective pptimization, Evolutionary Computation, 3, 1, 1-16.

Fonteix, C., Massebeuf, S., Pla, F., and Kiss, L.N. (2004). Multicriteria optimization of emulsion polymerization process, European J. Oper. Research, 153, 2, 350-359.

Ghose, T.K. and P. Gosh (1976). Kinetic analysis of gluconic acid production by Pseudomonas ovalis. J. App. Chem. Biotechnol., 26, 768-777.

Halsall-Whitney, H., Taylor, D., and Thibault, J. (2003). Multicriteria optimization of gluconic acid production using Net Flow, Bioproc. Biosyst. Eng., 25, 5, 299-307.

Halsall-Whitney H. and Thibault J. (2006). Multi-objective optimization for chemical processes and controller design: Approximating and classifying the Pareto domain. Comp. Chem. Engng., 30, 1155-1168.

Jaimes, A.L. and Coello Coello, C.A. (2008). Multiobjective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering. Chapter 2 in G.P. Rangaiah (Ed.) Multi-Objective Optimization: Techniques and Applications in Chemical Engineering, World Scientific.

Johansen, T.A. and Foss, B.A. (1995). Semi-empirical modeling of non-linear dynamic systems through identification of operating regimes and locals models. In: Neural Network Engineering in Control Systems, K Hunt, G Irwin and K Warwick, Eds., pp. 105-126, Springer-Verlag.

Kursawe, F. (1990). A variant of evolution strategies for vector optimization, Parallel Problem Solving From Nature I (PPSN-I), 193-197.

Liu, G.P., Yang, J. B., and Whidborne, J. F. (2003). Multiobjective pptimisation and control, Research Studies Press, England.

Masuduzzaman and Rangaiah G.P. (2008). Multi-Objective optimization applications in chemical engineering. Chapter 2 in G.P. Rangaiah (Ed.) Multi-Objective Optimization: Techniques and Applications in Chemical Engineering, World Scientific.

Muniglia, L., Kiss, L.N., Fonteix, C., and Marc, I. (2004). Multicriteria optimization of a single-cell oil production, European J. Oper. Research, 153, 2, 360-369.

Orlowska, E., Peters, J.F., Rozenberg, G., and Skowron, A. (2007). In Memory of Professor Zdzislaw Pawlak**,** Fundamenta Informaticae 75, i-ii.

Pawlak, Z. (1982). Rough Sets. Intern. J. Comput. Inform. Sci., 11, 341-356.

Pawlak, Z. (1991). Rough Sets: Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Boston, MA.

Pawlak, Z. (2004). Some issues on rough sets. J.F. Peters *et al.* (Eds.): Transactions on Rough Sets I, LNCS 3100, Springer-Verlag Berlin, 1–58.

Perrin, E., Mandrille, A., Oumoun, M., Fonteix, C., and Marc I. (1997). Optimisation globale par stratégie d'évolution: Technique utilisant la génétique des individus diploïdes. *RAIRO-Recherche Opérationnelle*, 31, 161-201.

Poloni, C., Giurgevich, A., Onesti, L., and Pediroda, V. (2000). Hybridization of a multi-objective genetic algorithm, a neural network, and a classical optimizer for complex design problem in fluid dynamics, Comp. Methods App. Mech. and Eng., 186, (2-4), 403-420.

Ramteke, M. and Gupta, S.K. (2008). The jumping gene adaptations of multi-objective genetic algorithm and simulated annealing. Chapter 4 in G.P. Rangaiah (Ed.) Multi-Objective Optimization: Techniques and Applications in Chemical Engineering, World Scientific.

Renaud, J., Thibault, J., Lanouette, R., Kiss, L.N., Zaras, K., and Fonteix, C. (2007). Comparison of two multicriteria methods : Net Flow and Rough Set methods for aid to decision in a high yield pulping process. European J. Oper. Research, 177: 1418-1432.

Roy, B. (1971). Problems and methods with multiple objectives functions, Mathematical Programming, 1, 239-266.

Roy, B. (1978). ELECTRE III: Algorithme de classement basé sur une représentation floue des préférences en présence de critères multiples. Cahiers du CERO, 20, 3-24.

Roy, B. (1991). The outranking approach and the foundations of the ELECTRE methods, *Theory and Decision*, 31, 49-73.

Scarelli, A., and Narula, C. N. (2002). A multicriteria assignment problem. *Journal of Multi-Criteria Decision Analysis*, 11, 65-74.

Shim, M., Suh, M., Furukawa, T., Yagawa, G., and Yoshimura, S. (2002). Pareto-based continuous evolutionary algorithms for multi-objective optimization. Engineering Computations, 19, 1, 22-48.

Silva, C. M., and Biscaia, E. C. (2003). Genetic Algorithm development for multi-objective optimization of batch free-radical polymerization reactors. Comp. Chem. Engng, 27, 1329-1344.

Thibault, J., Lanouette, R., Fonteix, C., and Kiss, L.N. (2002). Multicriteria optimization of high yield pulping process. Can. J. Chem. Eng., 80, 5, 897-902.

Thibault, J., Taylor, D., and Fonteix, C. (2001). Multicriteria optimization for the production of gluconic acid. 8th International Conference on Computer Applications in Biotechnology: Modeling and Control of Biological Processes, Quebec City, Canada, 24-27.

Thibault, J., Taylor D., Yanofsky, C., Lanouette, R., Fonteix, C., and Zaras, K. (2003). Multicriteria pptimization of a high yield pulping process with Rough Sets. Chem. Eng. Sci. 58, 203-213.

Triantaphyllou, E. (2000). Multicriteria decision making methods: A comparative study. *Applied Optimization Series*, Kluwer Academic Publishers.

Vafaeyan, S., Thibault, J., and Titica, M. (2007). Multi-criteria optimization of beer quality using rough set method. IFAC 10th Conference on Computer Applications in Biotechnology, Cancun, Mexico, 4-6 June.

Viennet, R., Fonteix, C., and Marc, I. (1996). Multicriteria optimization using a genetic algorithm for determining a Pareto set, International. J. Syst. Sci., 27, 2, 255-260.

Warren, R.H., Johnson J.A., and Huang, G.H. (2004). Application of rough sets to environmental engineering models. J.F. Peters *et al.* (Eds.): Transactions on Rough Sets I, LNCS 3100, Springer-Verlag Berlin, 356–374.

Yanofsky. C., Taylor, D.G., and Thibault, J. (2006). Novel methodology for assessing a ranked Pareto domain: Drift Group Analysis. Chem. Eng. Sci., 61, 1312-1320.

Zaras, K. and Thibault, J. (2007). Ranking by rough approximation of preferences for decision engineering applications. Proc. of the Second Conference of Rough Sets and Knowledge Technology, Toronto, Canada, 14-16 May.

Exercises

7.1 Add three more solutions to the three solutions of Table 7.1: one solution that is dominated by the existing three solutions, one solution that dominates all three solutions, and one solution that is neither dominating nor dominated.

7.2 Using NFM, rank the four-objective solutions of Table 7.14 using the relative weights and thresholds of Table 7.15.

Table 7.14 List of Pareto-optimal solutions of Exercise 7.2.

Solution	Objectives			
	C_1	C_2	C_3	C_4
	Max	Min	Min	Max
1	68.1	7.97	0.132	3.72
2	64.9	6.66	0.125	4.24
3	66.4	7.12	0.123	3.97
4	66.7	7.70	0.118	4.55

Table 7.15 Parameters used to rank the Pareto domain using Net Flow.

Objective	W_i	Q_i	P_i	V_i	Type
1	0.27	0.5	1.0	3.0	Max
2	0.20	0.4	0.8	2.0	Min
3	0.20	0.05	0.1	0.2	Min
4	0.33	0.3	0.5	1.0	Max

7.3 Using RSM for a five-objective optimization problem, (a) determine the maximum number of rules that can be defined, and (b) suggest the number of solutions that should be presented as a single batch to the decision-maker for ranking.

7.4 The small set of Pareto-optimal solutions of Table 7.14 is presented to the decision-maker who ranked the four solutions in the following order: solution 1 as the best one and then solutions 3, 4, and 2 in order of preference. Determine the resulting set of preference and non-preference rules that would be used to rank the entire Pareto domain using RSM.

Chapter 8

Multi-Objective Optimization of Multi-Stage Gas-Phase Refrigeration Systems

Nipen M. Shah[1], Gade Pandu Rangaiah[2] and Andrew F. A. Hoadley[1*]
[1]Department of Chemical Engineering, Monash University, Clayton, VIC 3800 Australia
[2]Department of Chemical & Biomolecular Engineering, National University of Singapore, Singapore 117576, Singapore

[*]Andrew.hoadley@eng.monash.edu.au

Abstract

Liquefied natural gas (LNG) is a clean burning fossil fuel which offers an energy density comparable to petrol and diesel fuels. There are many commercial processes available for the liquefaction of natural gas, for example single mixed refrigeration, and cascade refrigeration. Another alternative is gas-phase refrigeration processes. These processes are very flexible and inherently safer than condensing refrigeration processes. A shaftwork targeting method has recently been developed for multi-stage gas-phase refrigeration systems (Shah and Hoadley, 2007). In this chapter, a study has been carried out to optimize these systems for multiple objectives, by varying the operating parameters such as the minimum temperature driving force (ΔT_{min}) and pressure ratio. An interesting aspect of this study is the use of a superstructure model for the process simulation in order to allow for different numbers of refrigeration stages (from 2 to 7 stages). The two objective functions considered in the present optimization study are the capital cost and the energy requirement. A Non-dominated Sorting Genetic Algorithm

(NSGA-II) is used to generate the Pareto-optimal front, and an extended range of process parameters including the number of refrigeration stages is tested. The multi-objective optimization results are presented and discussed for two cases: cooling of a nitrogen stream using a nitrogen gas refrigerant, and the dual nitrogen/natural gas refrigerant process for LNG.

Keywords: Gas-phase Refrigeration, Dual Independent Expander Process, Liquefied Natural Gas, NSGA-II.

8.1 Introduction

Refrigeration systems are extensively used in the chemical industries in low temperature processes such as liquefaction of natural gas, ethylene purification and cryogenic air separation. In these kinds of low temperature systems, heat is rejected from the process by refrigeration to heat sinks being other process streams or refrigeration systems at the expense of mechanical work. The refrigeration systems employed are complex, and energy and capital intensive, and therefore, play a critical role in the overall plant economics.

Liquefied Natural Gas (LNG) is a clean burning fossil fuel. It offers an energy density comparable to petrol and diesel fuels, and produces less pollution. LNG is about 0.2% of the volume of natural gas at standard temperature and pressure, thus making it more economical to transport over long distances. The refrigeration and liquefaction are the key sections of the LNG plant, which typically account for 30-40% of the capital cost of the overall plant (Shukri, 2004). There are several licensed processes available for LNG, but most fall into either a cascade refrigeration or mixed refrigerant scheme.

In the Phillips optimized cascade process shown in Fig. 8.1a, refrigeration and liquefaction of the natural gas are achieved in a cascade system comprising of three different pure refrigerants: propane, ethylene and methane (Houser and Krusen, 1996). Each refrigeration stage requires one compression stage. In contrast, the PRICO process, as shown in Fig. 8.1b, uses a single mixed refrigerant made up of

nitrogen, methane, ethane, propane and iso-pentane (Swenson, 1977). The composition of the mixed refrigerant is selected in such a way that the liquid refrigerant evaporates at different temperature levels and provides cooling, which matches the condensation curve of the natural gas. In the PRICO process, only one compressor train is used to achieve the desired refrigeration and liquefaction. The mixed refrigerant systems require a careful selection and control of the refrigerant composition; on the other hand, the cascade systems are expensive to build and maintain.

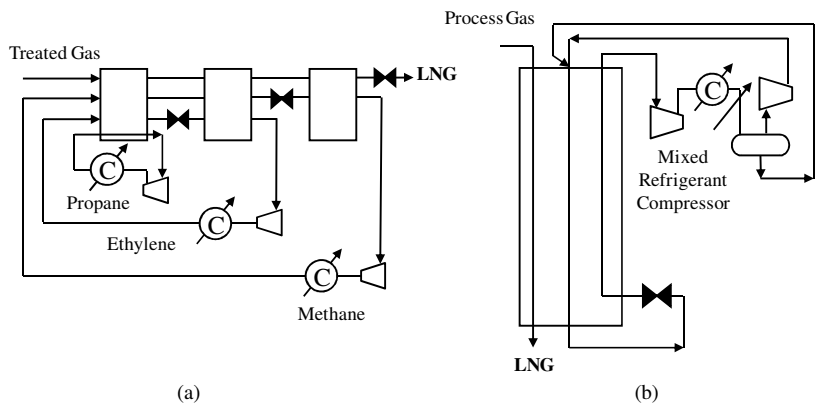


Fig. 8.1 (a) Phillips optimized cascade process for LNG (Houser and Krusen, 1996); (b) PRICO process for LNG (Swenson, 1977). Rectangular boxes are the multi-stream heat exchangers.

Gas-phase refrigeration systems can provide near-isentropic expansion and auto refrigeration. Moreover, they also have the ability to closely match the hot composite curve which is essential for achieving high energy efficiency. In gas-phase refrigeration systems, turbo-expanders provide expansion of a gas with the recovery of shaftwork. Moreover, they can handle considerable amounts of condensing as well as flashing liquid without a significant loss in efficiency or mechanical damage. Therefore, as turbo-expanders technology improves, gas-phase systems will compete and may overtake other refrigeration systems for LNG.

The overall design of any process should consist of two stages: targeting and design. Targeting is mainly an analysis tool that provides the initial screening of the process to identify the feasible design options. Targets for refrigeration systems provide compressor power requirements, estimates of capital and operating costs, and a platform to assess all the design options. In the design stage, the promising options selected from the targeting phase are developed further and subjected to detailed simulation and optimization.

A lot of research has been done on the synthesis and design of refrigeration processes, which includes the design of integrated refrigeration systems (Wu and Zhu, 2002), the selection of the mixed refrigerant composition (Duvedi and Achenie, 1997; Lee et al., 2002), the synthesis of refrigeration cycles to minimize capital and operating costs (Shelton and Grossmann, 1986; Vaidyaraman and Maranas, 1999), the retrofit of refrigeration systems integrated with heat exchanger networks (Wu and Zhu, 2001), and the optimization of natural gas plants to maximize ethane recovery (Diaz et al., 1997; Konukman and Akman, 2005). Jang et al. (2005) recently proposed a hybrid algorithm (written in MATLAB) based on a genetic algorithm and a quadratic search method for the economic optimization of a refrigeration plant modeled in ASPEN Plus. A turbo-expander plant for the recovery of natural gas liquids was taken as the example problem. However, with the exception of natural gas treatment processes, there has been very little research on gas-phase refrigeration processes. Moreover, to the best of our knowledge, there has not been any research on the single or multi-objective optimization (MOO) of multi-stage gas-phase refrigeration systems.

In this chapter, the design and optimization of two multi-stage gas-phase refrigeration systems (one for nitrogen cooling and another for liquefaction of natural gas) for two objectives is presented. A Multi-Platform, Multi-Language Environment (MPMLE) has been used as an interface to optimize the refrigeration processes simulated in HYSYS (Bhutani et al., 2007), and a Non-dominated Sorting Genetic Algorithm (NSGA-II, Deb et al., 2002) has been used to generate the Pareto-optimal solutions.

8.2 Multi-Stage Gas-Phase Refrigeration Processes

Fig. 8.2 shows a simple compression refrigeration cycle that involves an evaporator (or a process heat exchanger), a compressor, a condenser and an expansion valve. The saturated refrigerant is compressed to a higher pressure in the compressor (1→2). The superheated fluid is then condensed in the condenser to saturated liquid (2→3) before being expanded across the valve (3→4). The refrigerant vaporizes partially across the valve and passes through the evaporator where it exchanges heat with the process and leaves as a saturated vapor (4→1). In this cycle, the shaftwork is supplied to the compressor. The expansion across the valve is isenthalpic and no work is recovered.

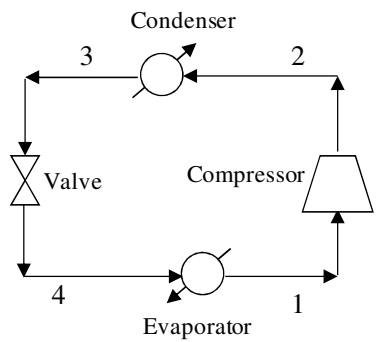


Fig. 8.2 A simple compression refrigeration cycle.

8.2.1 *Gas-Phase Refrigeration*

The gas-phase refrigeration system works on the concept of the Reverse-Brayton cycle, shown in Fig. 8.3. In this, unlike both gas and liquid phases in the compression refrigeration cycle (Fig. 8.2), the refrigerant remains in the gas phase throughout the system. The refrigerant at a high pressure is expanded across a turbo-expander where some work (W_{out}) is extracted and, simultaneously, the refrigerant temperature is reduced (1→2 in Fig. 8.3). The refrigerant then exchanges heat ($Q_{Process}$) with the hotter process stream (2→3), after which it passes through a heat exchanger (3→4) where it pre-cools the warm refrigerant before being

compressed back to the original pressure (4→5). The compressed refrigerant is then cooled down to the ambient temperature in a cooler (5→6). This process is called auto refrigeration as there is no external utility used for the pre-cooling of the warm refrigerant (6→1).

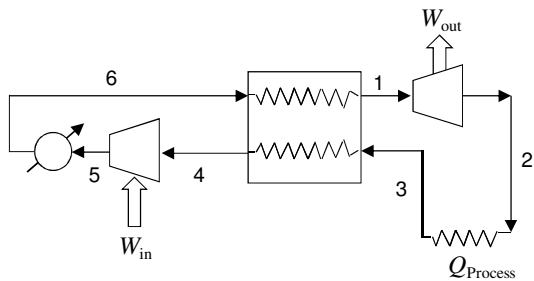


Fig. 8.3 Reverse-Brayton refrigeration cycle.

Shah and Hoadley (2007) proposed a multi-stage gas-phase auto refrigeration process for sub-ambient processes like liquefaction of natural gas, which is shown in Fig. 8.4 with n number of stages. In this figure, each multi-stream heat exchanger, H_i, along with a compressor, K_i, and an expander, E_i, represent one refrigeration stage. The process stream which requires cooling enters from the left and leaves from the last exchanger after passing through n heat exchangers. The warm refrigerant also enters from the left and exchanges heat with the cold refrigerant. In each stage i, the pre-cooled refrigerant is expanded to a lower pressure P_1 in the expander, which then cools the hot process stream. The cold refrigerant now at T^*_{i-1} is used to pre-cool the warm refrigerant. The refrigerant is then compressed back to P_0. The pressure ratio, which is the ratio of the inlet refrigerant pressure (P_0) and the pressure of the refrigerant at the outlet of the expander (P_1), is kept constant in the various stages throughout the process in order to avoid the losses associated with non-ideal mixing of streams.

During the pre-cooling of warm refrigerant, the cold refrigerant (at lower pressure, P_1) exchanges heat with the warm refrigerant (at higher pressure, P_0). The specific heat has a positive relationship with pressure. Therefore, in order to maintain the minimum driving force (ΔT_{min})

between hot and cold streams, a small amount of supplementary flow is required on the low pressure side to compensate for the difference in the specific heats. It is returned back after compression. For example, as shown in Fig. 8.4, the stream leaving the expander E_1 supplies supplementary flows to colder stages at T_1^* and lower; the flow in the reverse direction occurs after the coolers following compression.

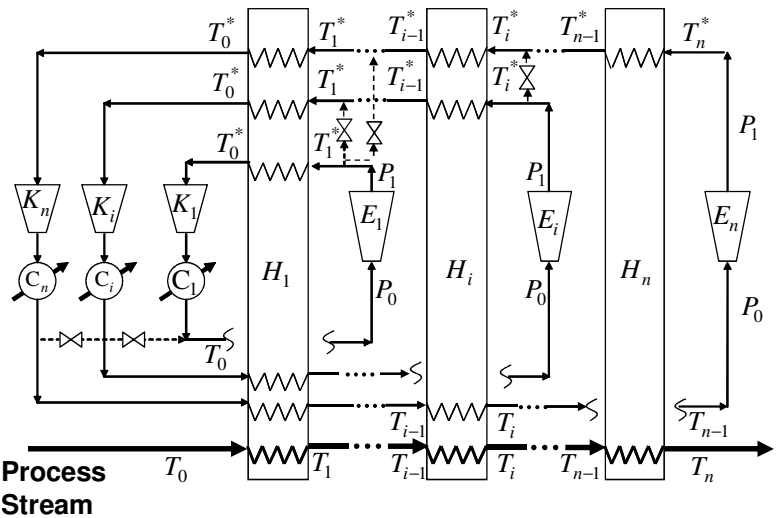


Fig. 8.4 A theoretical multi-stage refrigeration process.

8.2.2 *Dual Independent Expander Refrigeration Process for LNG*

The grand composite curve (cooling curve) for the liquefaction of 1000 kmol/h of natural gas consisting of 96.929 mol% methane, 2.938 mol% ethane, 0.059 mol% propane, 0.01 mol% n-butane and 0.064 mol% nitrogen is shown in Fig. 8.5. There are three steps involved in the process: gas-phase cooling, liquefaction of gas, and sub-cooling of liquid. One of the major considerations while designing the refrigeration systems is to closely approach the cooling curve of the process gas being liquefied, by using refrigerants that will match the different regions involved in the cooling curve. A close match results in the high refrigeration efficiency and reduces the energy consumption. A major

difficulty in matching the cooling curve lies in the liquefaction region where the curve becomes flatter.

Foglietta (2002) proposed a dual independent expander refrigeration system for the liquefaction of natural gas (Fig. 8.6). In this process, natural gas or methane is used to cool the feed natural gas stream to an intermediate temperature level – preferably up to the liquefaction temperature – and then nitrogen refrigerant is used to sub-cool the liquefied gas. The natural gas refrigerant partially condenses in the liquefaction stages, which provides a good match between the hot and cold composite curves in the liquefaction region.

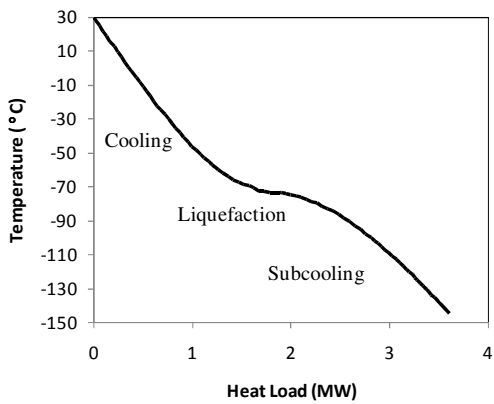


Fig. 8.5 Grand composite curve for LNG.

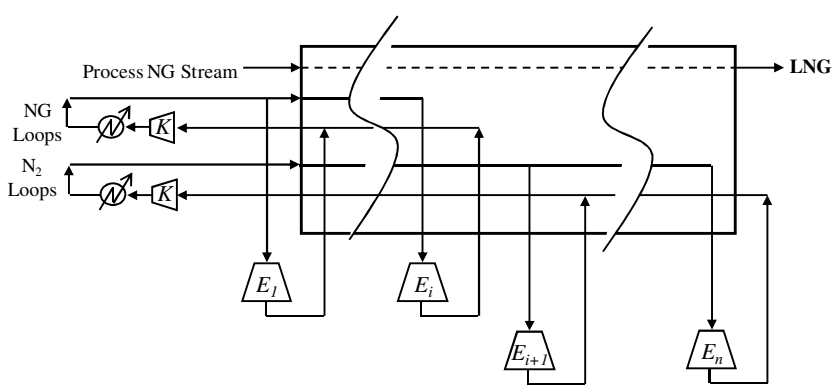


Fig. 8.6 Schematic of the dual independent expander refrigeration process.

The number of natural gas and the number of nitrogen refrigeration stages provide the degrees of freedom for the design of the dual independent expander refrigeration process. In addition, the temperature at which the nitrogen refrigerant is introduced is also an important parameter for optimization. In the present work, the shift from natural gas refrigerant to nitrogen refrigerant occurs when the process natural gas stream is completely liquefied.

8.2.3 *Significance of ΔT_{min}*

The minimum temperature driving force for heat transfer ($\Delta T_{\min}$) is a very important parameter in the design of refrigeration systems. By maintaining a constant $\Delta T_{\min}$ at both ends of a heat exchanger, it is ensured that the exchanger will be operating with a minimum driving force. This will appear as parallel curves on a temperature-heat flow diagram, as shown in Fig. 8.7, assuming there is no change of phase in either stream. The choice of $\Delta T_{\min}$ has a significant effect on the capital cost of a heat exchanger. Zero $\Delta T_{\min}$ would require infinite heat exchanger area (i.e. infinite capital cost) although it would require lower energy cost. On the other hand, the larger the $\Delta T_{\min}$, the lower is the capital cost, but higher is the energy cost. Hence, there is a trade-off between the capital cost and the energy efficiency. For cryogenic processes, the value of $\Delta T_{\min}$ ranges from 1 to 6 °C. Such small values of $\Delta T_{\min}$ can be achieved with either plate or plate-fin heat exchangers (Polley, 1993).

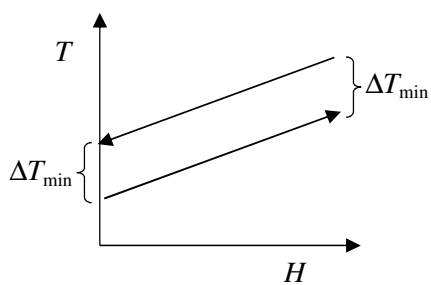


Fig. 8.7 Composite curves of a heat exchange process showing $\Delta T_{\min}$.

8.3 Multi-Objective Optimization (MOO)

As discussed throughout this book, many real-world chemical engineering problems require simultaneous optimization of several objectives. Often, these objectives conflict with one another (i.e., the improvement in one objective is accompanied by the deterioration in another objective), and so there is no single optimum. In such situations, the multi-objective problem will have a set of optimal solutions called Pareto-optimal solutions, in which, on moving from any one point to another, at least one objective improves while at least one other objective worsens. Thus, they are non-dominated or equally good solutions. Traditionally, MOO problems are converted to single objective optimization problems by different methods and then solved as single objective optimization problems. These methods include the weighting method and the ε-constraint method discussed in Chapters 1 and 6, and require several optimization runs to find the Pareto-optimal solutions. In the past two decades, evolutionary algorithms such as genetic algorithms have been modified for MOO, and they have found many chemical engineering applications. These applications since 2000 have been summarized in Chapter 2.

In this chapter, MOO of two refrigeration systems using NSGA-II and a process simulator is described. Commercial process simulation packages like HYSYS, ASPEN Plus and PRO/II are widely used by the chemical and related industries for designing new plants and for analyzing and retrofitting existing plants. The main advantage of using such simulators is the availability of many, rigorous process models and property packages. Some of the process simulators also provide built-in tools for optimization; for example, a sequential quadratic programming (SQP) algorithm is available in HYSYS. However, in the chemical and related industries, there are many complex operations that cannot be modeled or optimized using the modules and/or optimization techniques provided within the simulator.

Simulators like HYSYS and ASPEN Plus can be interfaced with Microsoft Excel or Visual Basic because of their ActiveX compliance. This feature can be used to optimize the process modeled in such simulators using powerful optimization algorithms written in high-level

programming languages such as C++ and FORTRAN. Bhutani et al. (2007) developed an interface program: multi-platform multi-language environment (MPMLE), which is written in Visual Basic (VB), to perform the optimization of processes simulated using a simulator such as HYSYS and using generic optimization programs such as NSGA-II.

The architecture and working principle of MPMLE are shown in Fig. 8.8. In the beginning, the user supplies the number of objectives, decision variables, constraints, and the NSGA-II parameters to the VB interface. The parameters include the population size, number of generations, crossover and mutation probabilities, and a seed for random number generation. The interface passes all these parameters to the optimizer (NSGA-II) which generates the first set of values for the decision variables, and passes them back to the interface. The HYSYS case is then called, and the set of values for the decision variables is copied to the built-in HYSYS spreadsheet which is directly connected to the HYSYS flowsheet. From the HYSYS results, the objectives are evaluated in the built-in spreadsheet and the values are sent back to the NSGA-II code through the VB interface. The NSGA-II code ranks the individuals according to their objective values. After a series of operations like selection, crossover and mutation, a new set of values for the decision variables is generated and supplied back to the VB interface. The iterative procedure continues until it reaches the maximum number of generations.

8.4 Case Studies

In this section, optimization of nitrogen cooling with a single gas-phase refrigerant (N_2) for multiple objectives using MPMLE, HYSYS and NSGA-II is demonstrated first. Then, the MOO of the dual independent expander process for liquefaction of natural gas is presented. Both processes are considered for typical industrial conditions.

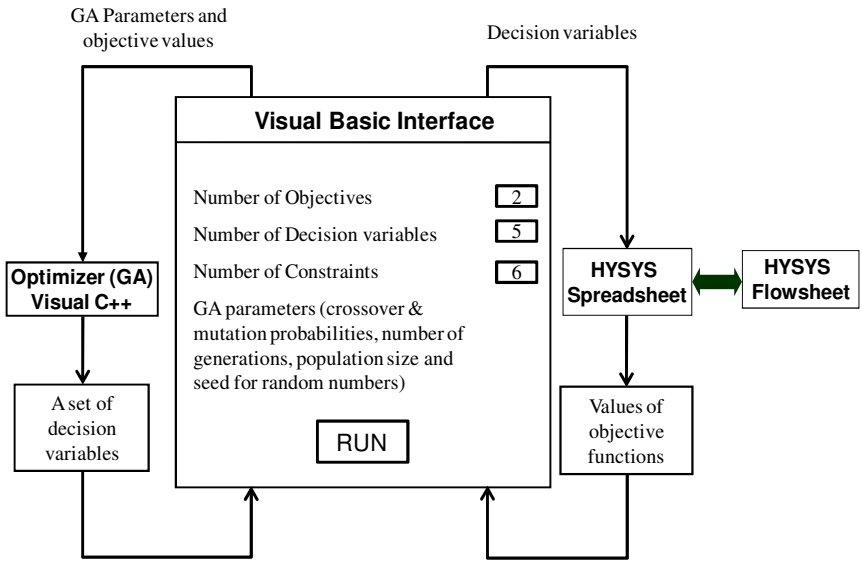


Fig. 8.8 Architecture and working principle of MPMLE.

8.4.1 *Nitrogen Cooling using N_2 Refrigerant*

It is required to cool 1000 kmol/h of the feed nitrogen gas at 2850 kPa from 30 °C to -145 °C with N_2 gas as the refrigerant using the multi-stage refrigeration process shown in Fig. 8.9. The hot process stream enters from the left and is cooled to the desired temperature using n refrigeration stages. The first objective is the minimization of total, net shaftwork requirement given by Eq. 8.1. Shaftwork is a very important objective in the refrigeration system optimization as it is directly associated with the operating cost.

$$\text{Minimize} \qquad W_{total} = W_{in} - \left(\sum_{i=1}^{n} W_{i,out} \right) \tag{8.1}$$

Here, W_{in} is the amount of work required by the compressor and $W_{i,out}$ is the amount of work extracted from the ith expander.

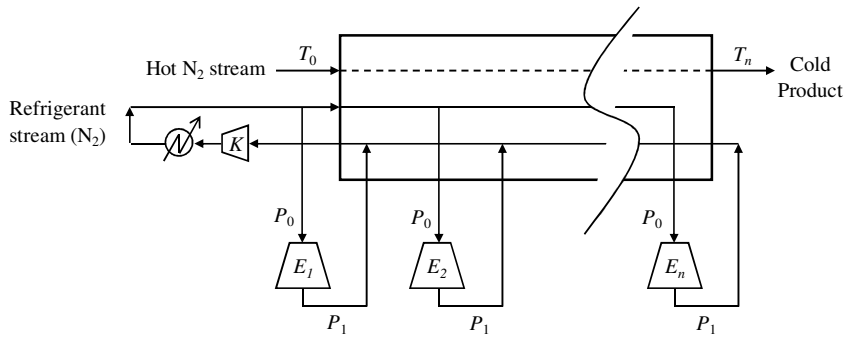


Fig. 8.9 Process flow diagram for nitrogen cooling with nitrogen gas as the refrigerant.

The second objective is the minimization of total heat exchanger area:

$$\text{Minimize} \qquad (UA)_{total} = \sum_{i=1}^{n} \sum_{k=1}^{i} (UA)_{i,k} \tag{8.2}$$

where $(UA)_{i,k}$ represents the product of the heat exchanger area and the overall heat transfer coefficient for the kth heat exchanger in the ith stage. Here, the overall heat transfer coefficient, U is assumed to be constant throughout the process. The number of refrigeration stages, n depends on the pressure ratio across the expanders. The method for deciding the number of stages is discussed later. It had been found that the pressure ratio across rotating equipments and the heat exchanger $\Delta T_{\min}$ are very important design parameters for the multi-stage gas-phase refrigeration process (Shah and Hoadley, 2007). The three decision variables and their bounds are:

$$3 \le P_{0,\,N2} \le 5 \text{ MPa (abs)} \tag{8.3}$$

$$1.5 \le P_{1,\,N2} \le 3.5 \text{ MPa (abs)} \tag{8.4}$$

$$1 \le \Delta T_{\min} \le 5 \text{ °C} \tag{8.5}$$

Here, $P_{0,\,N2}$ and $P_{1,\,N2}$ represent the refrigerant pressure at the inlet and the exit, respectively, of each of the n expanders. The optimization is subject to the following constraints:

$$-145.5 \le T_n \le -144.5 \ \text{°C} \tag{8.6}$$

$$\text{Cold pinch } T \le \text{Hot pinch } T \tag{8.7}$$

$$1 \le n \le 7 \tag{8.8}$$

Here, T_n is the final temperature of the cold product as shown in Fig. 8.9. The cold and hot pinch temperatures represent the temperatures on the cold and hot sides of a heat exchanger where the composite curves are the closest. If the condition in Eq. 8.7 is not true, then it means there is a temperature cross (or internal pinching) within the heat exchanger as shown in Fig. 8.10. When this occurs, the heat exchanger area (or capital cost) becomes infinite. Hence, Eq. 8.7 ensures a finite, positive temperature difference between the hot and cold composite curves (or, in other words, both composite curves are not overlapping as shown in Fig. 8.10).

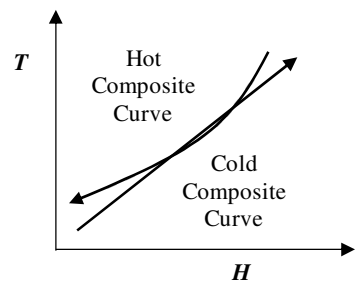


Fig. 8.10 Composite curves showing internal pinching in a heat exchanger.

The constraint in Eq. 8.7 on the temperatures in the heat exchangers has been implemented within the HYSYS spreadsheet. HYSYS does not calculate the value of *UA* when there is any internal pinching in the exchanger. In such a case, a very large value of *UA* (= 27.8 MW/°C) is supplied to the built-in spreadsheet where the objectives are calculated. Since one of the objectives is the minimization of *UA*, the optimizer will give lower priority for selecting the set of values for the decision variables giving such a large *UA* value in the next generations. So by doing this, we are actually discarding the solutions with internal pinching. The constraint on the maximum number of stages is also

implemented in HYSYS along with the algorithm in Fig. 8.11. Further, the superstructure is created for 7 stages only. Thus, only the constraint in Eq. 8.6 is handled by the optimizer, NSGA-II, which uses constrained-dominance principle for solving constrained MOO problems (Deb et al., 2002). In this method, two solutions are picked from the population and the better solution is preferred which is either feasible in terms of the constraints or is infeasible but has a smaller overall constraint violation. If both the solutions are feasible, it is checked whether they belong to different non-dominated fronts. In that case, the one that belongs to the better non-dominated front is selected. If both of them are from the same non-dominated front, the one which belongs to the least crowded region is selected.

Shah and Hoadley (2007) proposed a shaftwork targeting methodology for multi-stage gas-phase refrigeration processes. In this method, values of ΔT_{min} and the number of refrigeration stages are first selected. Then, based on the initial process stream temperature and the final product temperature, the value of the pressure ratio is calculated. In the present work, a modified targeting methodology has been used for estimating the number of refrigeration stages based on the values of decision variables selected by the optimizer. The algorithm for this is shown in Fig. 8.11. Firstly, the values of the decision variables ($P_{0,N2}$, $P_{1,N2}$ and ΔT_{min}) generated by the optimizer are passed to the HYSYS flowsheet through the VB interface. In addition, the values of the initial process temperature, T_0, $T_{desired}$ (which is ΔT_{min} colder than the final product temperature), and the isentropic efficiency (η_{isen}) are also supplied. In all simulations, the value of η_{isen} for compression as well as expansion is taken as 80%. A superstructure-based flowsheet considering the maximum possible number of stages (7) has been created in HYSYS. Starting from one refrigeration stage (i = 1), the temperature of the refrigerant at the exit of the expander/entering the process heat exchanger (T_{out}) is calculated using HYSYS simulation and then checked against $T_{desired}$. If the difference between the two is within the given tolerance limit or T_{out} is less than $T_{desired}$, the value of i is taken as the number of refrigeration stages, the targeting calculations terminate, and the objectives are calculated for optimization. If the difference between the two temperatures is outside the tolerance, then another stage is added and

the iterations are continued from calculating the temperature of the refrigerant at the end of this added expander.

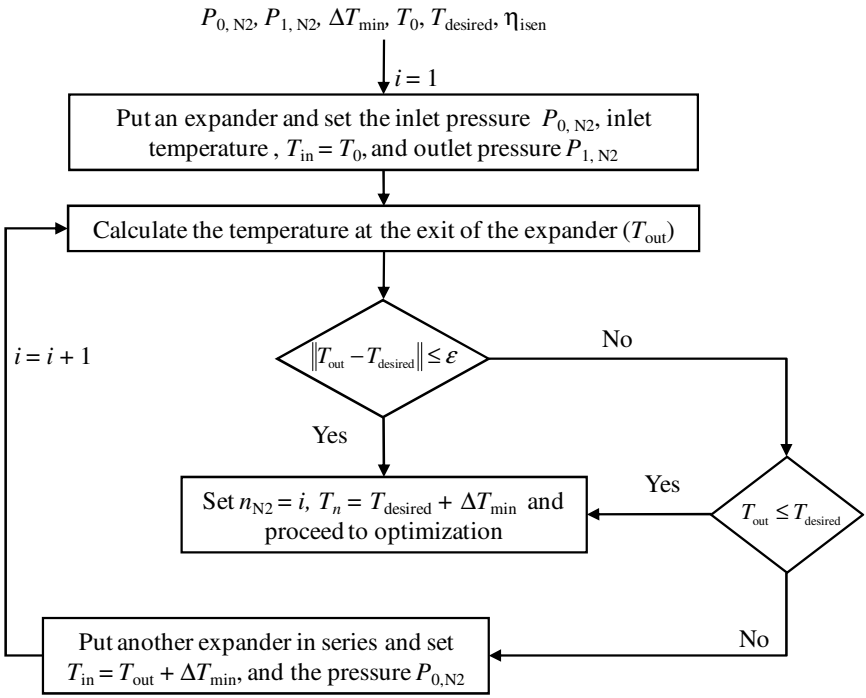


Fig. 8.11 Algorithm to calculate the number of refrigeration stages.

The Pareto-optimal solutions for the above optimization problem for nitrogen cooling are shown in Fig. 8.12. These results were obtained with NSGA-II after 400 generations with a population size of 100. The other GA parameters were: crossover probability = 0.65, mutation probability = 0.25 and random number seed = 0.8. These values were chosen after around 15 trials with different values of the GA parameters, in order to get a very good spread of the Pareto-optimal solutions.

The Pareto-optimal solutions are spread over $(UA)_{total}$ from 1 to 7 MW/°C and W_{total} from 2.6 to 3.2 MW. The solutions with $1 \le (UA)_{total} \le 2$ MW/°C correspond to 4 refrigeration stages, and those with a higher $(UA)_{total}$ correspond to 7 refrigeration stages (Fig. 8.13). The shaftwork requirement for 4 stages is higher than that for 7 stages. This is expected

as a lower number of stages requires a higher pressure ratio across expanders to achieve the same final refrigeration temperature. Hence, the work required to compress the refrigerant back to $P_{0,N2}$ would also be higher. Interestingly, there are no Pareto-optimal solutions other than with 4 or 7 stages (Fig. 8.13), since the solutions for 5 and 6 stages were all eliminated.

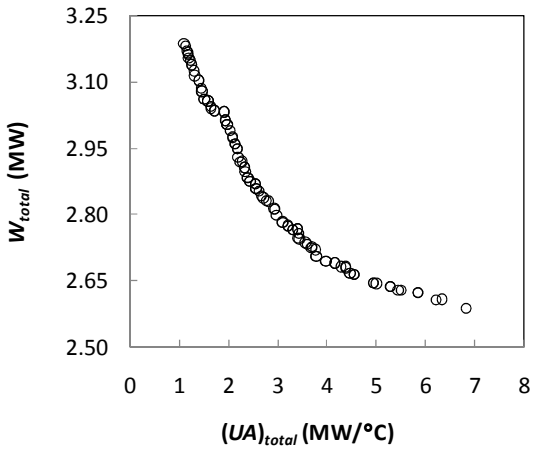


Fig. 8.12 Pareto-optimal front for nitrogen cooling process.

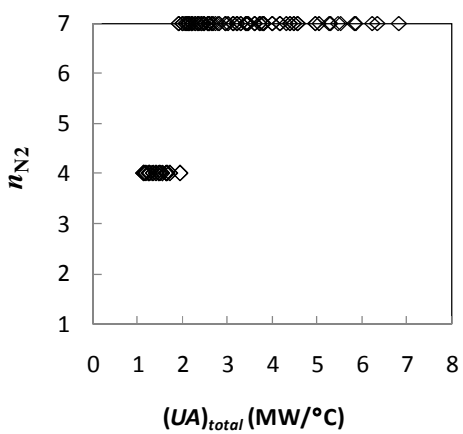


Fig. 8.13 Number of stages corresponding to the Pareto-optimal front in Fig. 8.12.

The values of ΔT_{min} corresponding to the Pareto-optimal solutions are shown in Fig. 8.14a. It is evident that as the ΔT_{min} decreases, $(UA)_{total}$ increases or vice versa. The discontinuity and sudden increase in ΔT_{min} at $(UA)_{total}$ ~ 2 MW/°C when the optimal number of stages changes from 4 to 7, are interesting. It can be seen that $P_{0,N2}$ should be at its highest allowable value for minimizing both the objectives (Fig. 8.14b). The values of $P_{1,N2}$ corresponding to the Pareto-optimal solutions are shown in Fig. 8.14c. All the values near 2 MPa correspond to 4 refrigeration stages, and the values between 2.5 and 3 MPa correspond to 7 stages.

Interestingly, there seem to be multiple solutions – two different sets of $P_{0,N2}$ and $P_{1,N2}$ which give the same objective values as shown in Fig. 8.14b and Fig. 8.14c. The number of refrigeration stages and the objective values largely depend on the pressure ratio ($P_{0,N2}/P_{1,N2}$). For $2 \leq (UA)_{total} \leq 4$ in Fig. 8.14b and Fig. 8.14c, $P_{0,N2}$ and $P_{1,N2}$ attain two different sets of values ($P_{0,N2}$ ~ 4.8 and 4.4 MPa, $P_{1,N2}$ ~ 2.9 and 2.7 MPa). These values result in almost the same pressure ratio for both sets (1.66 and 1.63) giving the same number of refrigeration stages (7). To understand this further, the same problem was optimized for a fixed value of $P_{0,N2}$ = 5 MPa. The results are shown in the Appendix.doc in the folder: Chapter 8 on the CD. In this case, there are no multiple optimal solutions and the Pareto-optimal front is almost the same as that in Fig. 8.12. All these are consistent with the finding of Shah and Hoadley (2007) that pressure ratio and ΔT_{min} are very important design parameters for multi-stage gas-phase refrigeration processes. Fig. 8.14d shows the values of the final product temperature. The desired temperature for this process was -145±0.5 °C, and all its optimal values are near the upper limit in order to minimize the shaftwork required.

Fig. 8.15 shows the composite curves for UA = 1.12 MW/°C among the Pareto-optimal solutions in Fig. 8.12 and Fig. 8.14. The y-axis represents the temperature profiles of the hot nitrogen process stream being cooled from 30 °C to -144.5 °C, and of the cold nitrogen refrigerant stream; the x-axis represents the heat load. It can be seen that there is a very good match of the composite curves with no indication of internal pinching.

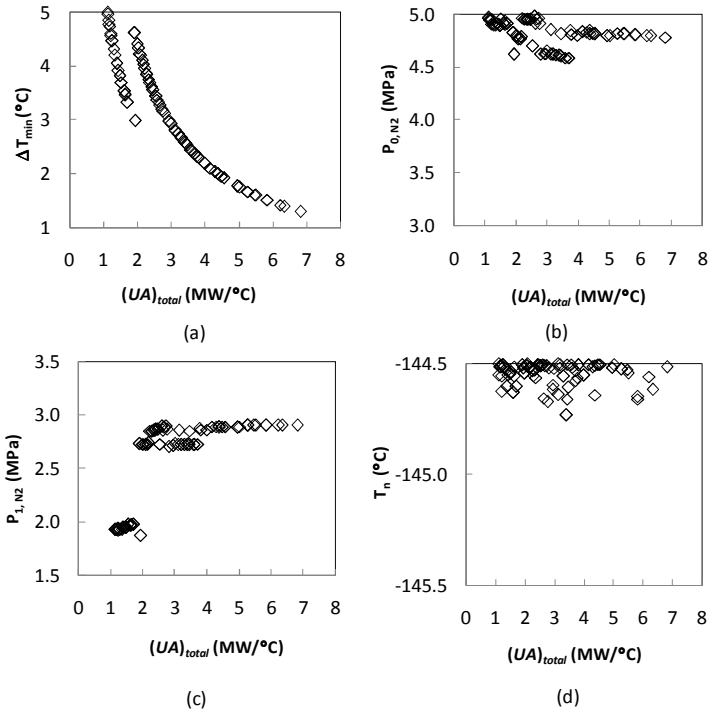


Fig. 8.14 Values of the decision variables and constraint corresponding to the Pareto-optimal front in Fig. 8.12: (a) heat exchanger $\Delta T_{\min}$; (b) initial refrigerant pressure, $P_{0,N2}$; (c) final refrigerant pressure, $P_{1,N2}$; and (d) final product temperature, T_n.

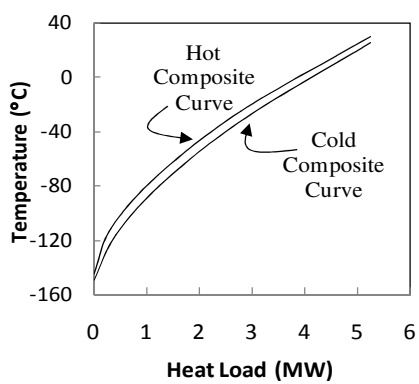


Fig. 8.15 Composite curves for one Pareto-optimal solution corresponding to the minimum UA = 1.12 MW/°C in Fig. 8.12.

8.4.2 *Liquefaction of Natural Gas using the Dual Independent Expander Process*

In this section, MOO of the dual independent expander refrigeration process, which has been explained in Section 8.2.2, is described. The feed natural gas stream at 5.5 MPa consisting of 96.929 mol% methane, 2.938 mol% ethane, 0.059 mol% propane, 0.01 mol% n-butane and 0.064 mol% nitrogen is to be liquefied and sub-cooled using natural gas (having the same composition as the feed natural gas) and N_2 as the refrigerants (Fig. 8.16). The LNG product rate is kept constant at 22.6 kg/s. In the process shown in Fig. 8.16, the liquefied gas is expanded across a multi-phase turbine (E_{MPT}) to 0.11 MPa. Multi-phase turbines are generally used to produce low temperature liquefied gases by expanding fluids. They can recover some work from the expansion of fluids and increase the cold liquid yield. In this case, the LNG is produced at around -160 °C. Because of the expansion across the multi-phase turbine, the LNG stream vaporizes partially. The cold gas is then separated from the LNG stream, and compressed to 3.5 MPa in the fuel gas compressor (K_{GC}) before being used in the gas turbine unit as a combustion fuel. The gas turbine unit is, however, not considered in the present study. The temperatures with asterisks in Fig. 8.16 represent the temperatures on the refrigerant side which are $\Delta T_{\min}$ colder than the temperatures on the hot process side (for example, $T_n^* = T_n - \Delta T_{\min}$).

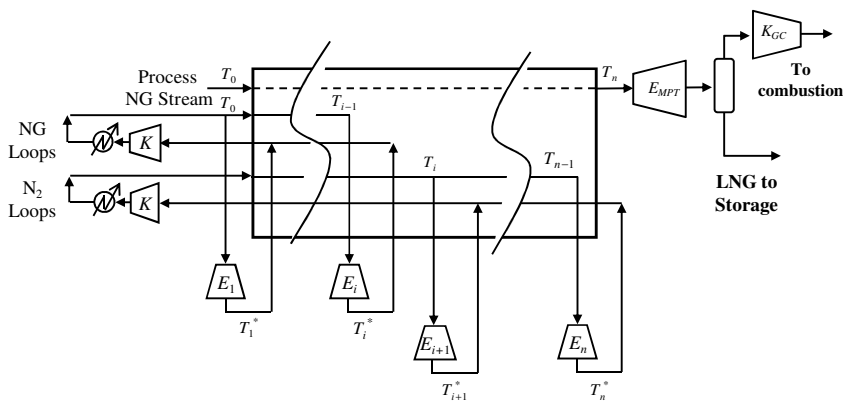


Fig. 8.16 Schematic of the dual independent expander refrigeration process.

The two objectives for the optimization are: total shaftwork requirement and total capital cost.
Minimize

$$W_{total} = W_{in,\mathrm{NG}} + W_{in,\mathrm{N2}} + W_{in,GC} - \left[\sum_{i=1}^{n} W_{i,out} + W_{MPT}\right] \tag{8.9}$$

Minimize

$$C_{total} = C_{\mathrm{NG},Comp} + C_{\mathrm{N2},Comp} + C_{GC} + C_{HE} + C_{MPT} + C_{EXP} \tag{8.10}$$

Here, $W_{in,\mathrm{NG}}$, $W_{in,\mathrm{N2}}$ and $W_{in,\mathrm{GC}}$ represent the shaftwork required in the natural gas, N_2 and fuel gas compressors respectively, $W_{i,out}$ is the shaftwork recovered from the ith turbo-expander, W_{MPT} is the amount of work recovered from the multi-phase turbine (E_{MPT}, Fig. 8.16), and C_{total} is the total capital cost in US$. $C_{\mathrm{NG},comp}$, $C_{\mathrm{N2},comp}$, C_{GC}, C_{HE}, C_{MPT} and C_{EXP} denote the cost of natural gas compressor, nitrogen compressor, fuel gas compressor, heat exchangers, multi-phase turbine and n turbo-expanders respectively. In this study, only major equipment items that dominate the capital cost have been considered in the calculation of the total cost. It should also be noted that these costs are based on empirical cost models rather than industrial quotations, and are probably only accurate to ± 20%.

The cost of a centrifugal compressor (C_{comp} US$) can be calculated from the following correlation (Calise et al., 2007):

$$C_{comp} = 91562\left(\frac{W_{in}}{445}\right)^{0.67} \tag{8.11}$$

where W_{in} is in kW. Plate-fin heat exchangers are very suitable for gas-phase cryogenic processes. The cost of a plate-fin heat exchanger (C_{HE} US$) of area A m^2 can be calculated from the following correlation (Calise et al., 2007):

$$C_{HE} = 130\left(\frac{A}{0.093}\right)^{0.78} \tag{8.12}$$

Using this equation, the capital cost of heat exchangers based on their area is:

$$C_{HE} = 130\left(\frac{\sum_{i=1}^{n_{NG}}\sum_{k=1}^{i}(UA)_{i,k} + \sum_{j=n_{NG}+1}^{n_{N2}}\sum_{k=1}^{j}(UA)_{j,k}}{0.093U}\right)^{0.78} \tag{8.13}$$

Here, $(UA)_{i,k}$ represents the product of overall heat transfer coefficient and the area for the kth heat exchanger in the ith natural gas refrigeration stage. Similarly, $(UA)_{j,k}$ represents the product of U and A for the kth heat exchanger in the jth nitrogen refrigeration stage. Here, a constant value of $U = 0.3$ kW/m^2 °C is assumed throughout the process.

The cost data of turbo-expanders have been obtained from Mafi-Trench Corporation who is a supplier of turbo-expander systems (Guerrero, 2007). The cost of turbo-expanders range from $750,000 to $2,500,000 for the maximum power output range between 1000 kW and 15000 kW. These cost data were fitted to a linear correlation as below:

$$C_{EXP} = 126.9W_{out} + 661{,}111 \tag{8.14}$$

where C_{EXP} is the cost of a turbo-expander in US$, and W_{out} is the power output from the turbo-expander in kW. Eq. 8.14 is used separately for each of the n turbo-expanders. The cost of a multi-phase turbine has been approximated with the cost function for a gas turbine (Calise et al., 2007), as

$$C_{MPT} = \left[-98.328\ln\left(W_{MPT}\right)+1318.5\right]\cdot W_{MPT} \tag{8.15}$$

where W_{MPT} is in kW and C_{MPT} is in US$.

The optimization problem of the liquefaction of natural gas using the dual independent expander process has five decision variables. The following bounds are used for these decision variables:

$$\left.\begin{array}{c} 4 \le P_{0,\,NG} \le 6 \text{ MPa (abs)} \\ 0.7 \le P_{1,\,NG} \le 3.1 \text{ MPa (abs)} \\ 3.5 \le P_{0,\,N2} \le 5 \text{ MPa (abs)} \\ 0.7 \le P_{1,\,N2} \le 3 \text{ MPa (abs)} \\ 1 \le \Delta T_{\min} \le 6 \text{ °C} \end{array}\right\} \tag{8.16}$$

Here, $P_{0,NG}$ and $P_{1,NG}$ are the refrigerant pressure at the inlet and the exit, respectively, of each turbo-expander in the natural gas refrigeration loop.

The upper bounds on $P_{0,NG}$ and $P_{0,N2}$ are dependent on the critical pressure of natural gas and N_2 respectively. The bounds on ΔT_{min} are chosen based on the normal operating range for gas-phase systems. The values of the initial and the final refrigerant pressure along with the value of ΔT_{min} decide the number of refrigeration stages. Moreover, ΔT_{min} has also got a significant effect on the heat exchanger area and hence the capital cost. Therefore, these variables are very important in the economic optimization. The constraints for the optimization problem are:

$$\left.\begin{array}{c} -156.5 \le T_n \le -139.5 \ \text{°C} \\ \text{Yield} \ge 0.9 \\ UA < 27.8 \ \text{MW/°C} \\ \text{Cold pinch } T \le \text{Hot pinch } T \\ 1 \le n_{NG} \le 3 \\ 1 \le n_{N2} \le 4 \\ T_{comp} \le 100 \ \text{°C} \end{array}\right\} \tag{8.17}$$

The constraints are very similar to those discussed in the first case study. T_n is the final refrigeration temperature as shown in Fig. 8.16. This constraint directs the amount of sub-cooling required by the nitrogen refrigeration loop. The colder the final refrigeration temperature or T_n, the higher the yield obtained, and, at the same time, a lower amount of liquid vaporizes across the multi-phase turbine. The maximum temperature of -139.5 °C is required to get the desired amount of liquid product. A constraint on the yield of the liquefied product is applied to control the amount of the LNG vaporized across the multi-phase turbine. There is a very high probability of obtaining solutions with internal pinching in heat exchangers (especially in the liquefaction region). To prevent this, as explained in the previous case study, *UA* is set equal to 27.8 MW/°C for the heat exchangers that do not satisfy the constraint: Cold pinch $T \le$ Hot pinch T. This increases the value of the total heat exchanger area by a factor of 10, and consequently the cost of heat exchangers increases tremendously. As a result, the optimizer rejects this particular set of values for the decision variables giving very large *UA* in

the next iterations. There could be many points in the decision variable space that can result in the internal pinching. The very high value of *UA* itself is sufficient to avoid these points. However, an additional constraint on *UA*, as shown in Eq. 8.17, is provided which eliminates any chances of getting solutions with internal pinching.

The constraints on T_n, the yield and *UA* are handled by the optimizer. The constraints on n_{NG}, n_{N2} and T_{comp} are implemented within the HYSYS spreadsheet. The numbers of natural gas (n_{NG}) and nitrogen (n_{N2}) refrigeration stages are limited to a maximum of 3 and 4, respectively. T_{comp} is the temperature at the outlet of a compressor. For a given pressure ratio, if this temperature goes above 100 °C then inter-stage cooling is introduced. In that case, the refrigerant stream going out of the first compression stage is cooled to ambient using a cooler before it enters the second compression stage. Here, each of the two compression stages will have the same pressure ratio. In the case of a very high pressure ratio, three compression stages (with two inter-stage coolers) may be needed. In that case, the stream leaving the second compression stage will be cooled to ambient before entering the third compression stage. Again, here the pressure ratio across all compressors will be the same. All these calculations are performed within the built-in HYSYS spreadsheet.

The targeting method to determine n_{NG} and n_{N2} within the optimization is shown in Fig. 8.17. It is similar to that in Fig. 8.11 for nitrogen cooling. Initially, the values of all decision variables chosen by the optimizer, along with the values of T_0, $T_{desired}$ (which is ΔT_{min} colder than T_n) and the isentropic efficiency, are supplied to the HYSYS flowsheet for simulation. Firstly, natural gas is used as the refrigerant. The liquid fraction on the process side is checked after each refrigeration stage. As soon as the process natural gas stream liquefies completely, the natural gas loop terminates. The nitrogen refrigerant is then introduced and a similar procedure is repeated until the desired product temperature is achieved. The numbers of natural gas and nitrogen refrigeration stages are then passed to the VB interface to continue with the optimization.

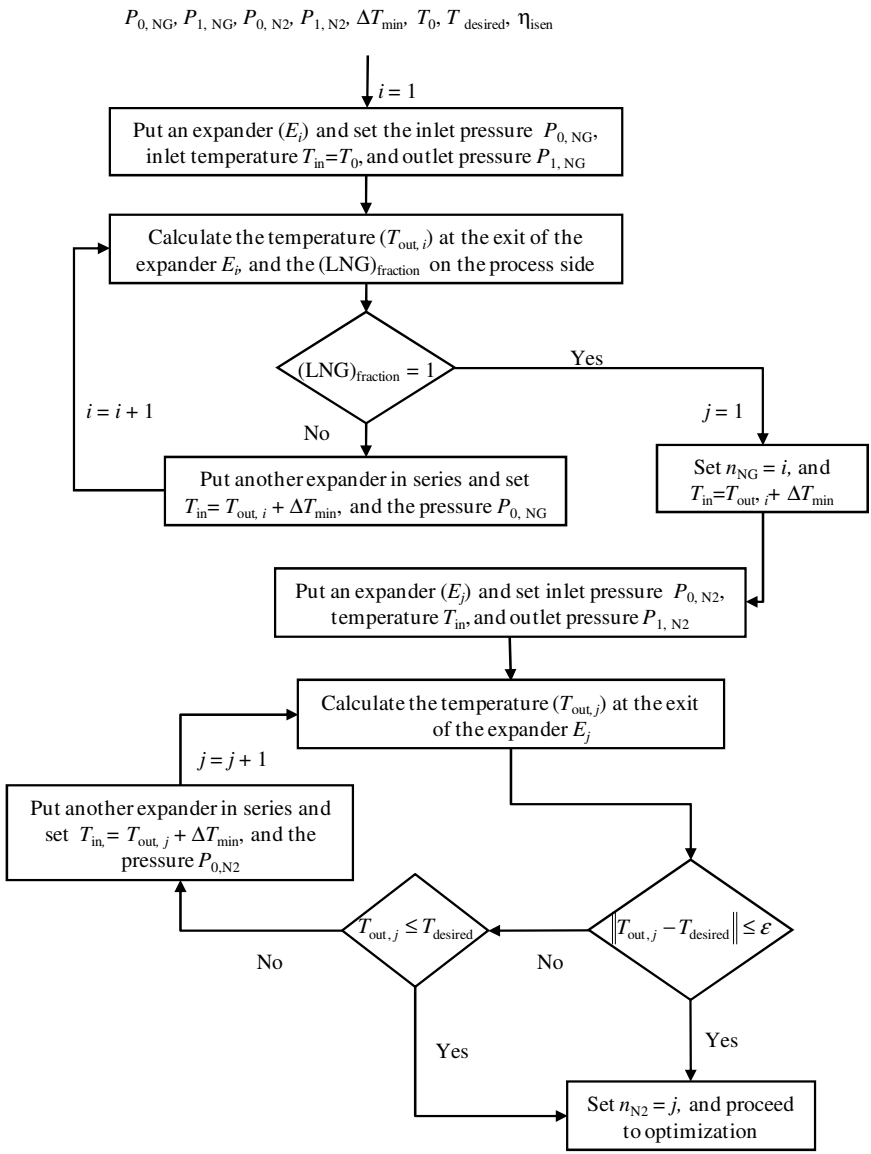


Fig. 8.17 Algorithm to calculate the number of natural gas and number of nitrogen refrigeration stages for the dual independent expander refrigeration process for LNG.

Twelve different flowsheets have been created in HYSYS considering all possible combinations of n_{NG} = 1 to 3 and n_{N2} = 1 to 4. One of these (depending on i and j values) is called during the steps in the algorithm shown in Fig. 8.17. The values of the decision variables are first passed to a HYSYS flowsheet where a spreadsheet is created that calculates the number of nitrogen and the number of natural gas stages using the targeting method shown in Fig. 8.17. In addition, the number of compression stages is also calculated in a separate spreadsheet created in the same HYSYS flowsheet. The values of n_{NG} and n_{N2} are passed back to the VB interface which then calls the relevant HYSYS flowsheet and passes all the decision variables to its built-in spreadsheet. The objective and constraint values are then passed back to the VB interface. This is a very useful feature of the MPMLE as a complicated superstructure-based MOO problem could be solved very easily.

The Pareto-optimal solutions for the above optimization problem are shown in Fig. 8.18. These results were obtained using NSGA-II with the following parameters: number of generations = 800, population size = 100, crossover probability = 0.65, mutation probability = 0.08 and seed for random number generation = 0.35. The Pareto-optimal solutions are spread over C_{total} from \$8.5 to 13 million and W_{total} from 28 to 32 MW. In general, as W_{total} (which is a major component of the operating cost) decreases the capital cost increases, or vice versa. This can be explained better with Fig. 8.19 which shows the numbers of natural gas and nitrogen refrigeration stages; two and three natural gas stages are selected as the Pareto-optimal solutions. As explained earlier, it is very difficult to match the hot and cold composite curves in the liquefaction region. The probability of getting solutions without internal pinching is higher with three natural gas stages than two stages. Therefore, there are fewer optimal solutions with two natural gas stages than with three natural gas stages. Moreover, since a higher pressure ratio is required for lower number of stages, the Pareto-optimal solutions with two natural gas stages require more shaftwork than three natural gas stages. At the same time, three natural gas stages require higher capital cost than two stages, which is understandable as the former would require an extra turbo-expander. Interestingly, in the case of nitrogen stages, 1 stage is found to be optimum for all the Pareto-optimal solutions. The composite

curves for one Pareto-optimal solution corresponding to C_{total} =\$12.9 million, and the lowest $\Delta T_{\min}$ (= 2.73 °C) is shown in Fig. 8.20. It can be seen that, despite being the optimal solution with the lowest $\Delta T_{\min}$, there is a very good match of the composite curves, even in the liquefaction region (shown in Fig. 8.20 with an oval) where there is a maximum probability of having a temperature cross.

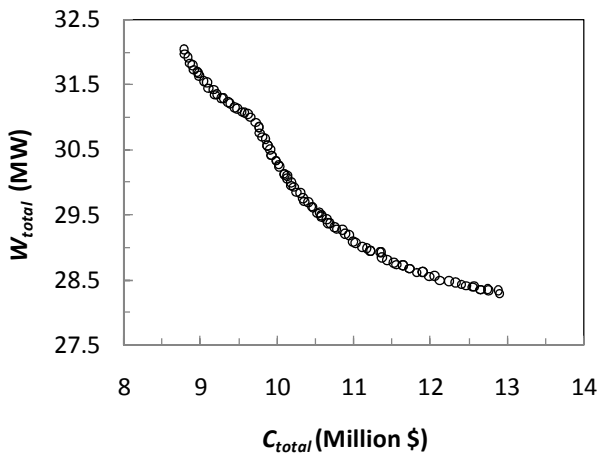


Fig. 8.18 Pareto-optimal front for the optimization of dual independent expander refrigeration process.

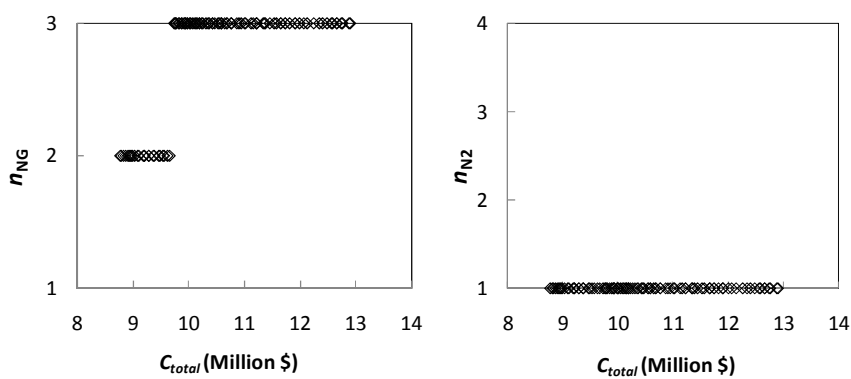


Fig. 8.19 Number of natural gas and number of nitrogen refrigeration stages corresponding to the Pareto-optimal front in Fig. 8.18.

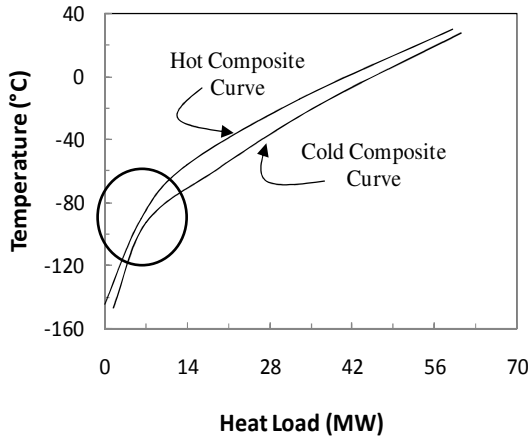


Fig. 8.20 Composite curves for the Pareto-optimal solution corresponding to the lowest shaftwork and the highest capital cost in Fig. 8.18: C_{total} = \$ 12.9 Million, W_{total} = 28.29 MW, ΔT_{min} = 2.73 °C, n_{NG} = 3, n_{N2} = 1, A_{total} = 92574 m^2.

Fig. 8.21 shows the values of decision variables corresponding to the Pareto-optimal front in Fig. 8.18. The total capital cost strongly depends on the heat exchanger area which in turn depends on ΔT_{min}. The heat exchanger area and hence the capital cost increases as ΔT_{min} decreases (Fig. 8.21a). At around C_{total} = \$ 9.75 million, ΔT_{min} suddenly increases from 5 °C to 6 °C and then decreases from 6 °C to 2.73 °C. This sudden increase happens due to an increase in the number of natural gas refrigeration stages from 2 to 3 (Fig. 8.19). The optimal $P_{0,NG}$ is between 5.7 and 6 MPa for minimizing both the objectives (Fig. 8.21b). Interestingly, the majority of the solutions prefer the highest possible value of $P_{0,NG}$, 6 MPa. The optimal $P_{1,NG}$ lies in two regions: nearly constant at about 1.4 MPa and at 2.6 MPa as shown in Fig. 8.18c. All the optimal solutions corresponding to ~1.4 MPa and ~2.6 MPa are, respectively, those with two and three natural gas refrigeration stages. The pressure ratio is lower in the case of $P_{1,NG}$ ~ 2.6 MPa, and hence it requires three stages to reach the liquefaction temperature. As expected, operating at a higher pressure ratio, the solutions with two natural gas refrigeration stages require more shaftwork than the solutions with three natural gas stages. The optimal $P_{0,N2}$ is a constant value of 5 MPa for all the solutions (not shown in Fig. 8.21). Optimal $P_{1,N2}$ lies in two different

regions (Fig. 8.21d): values between 1 and 1.35 MPa which correspond to two natural gas stages, and between 1 and 1.24 MPa corresponding to three natural gas stages.

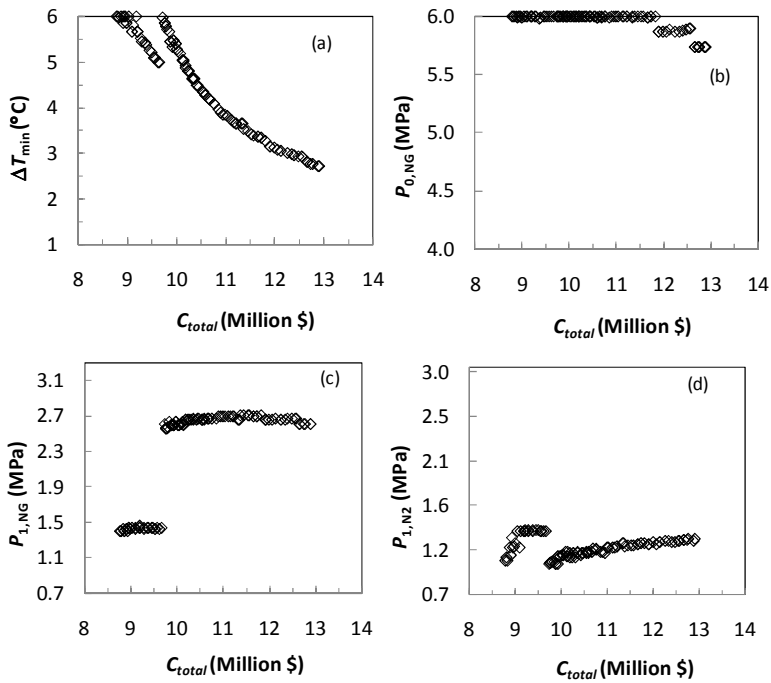


Fig. 8.21 Values of decision variables corresponding to the Pareto-optimal front in Fig. 8.17: (a) heat exchanger ΔT_{min}; (b) initial refrigerant pressure in the natural gas loop, $P_{0,NG}$; (c) final natural gas refrigerant pressure, $P_{1,NG}$; and (d) final N_2 refrigerant pressure, $P_{1,N2}$.

The values of two constraints corresponding to the Pareto-optimal solutions in Figs. 8.18 and 8.21 are shown in Fig. 8.22. It can be seen that the maximum possible yield of LNG is 98%; from this maximum, it decreases and remains almost constant at 90%, close to its lower bound (Fig. 8.22a), which is as expected. To achieve 98% yield, the LNG stream needs to be cooled to -156.5 °C (Fig. 8.22b) which is the lower limit on T_n. Since the yield depends on the degree of sub-cooling, there are no optimal solutions with yield higher than 98%. Similarly, to achieve 90% yield, a minimum of T_n = -143.8 °C is required; hence, even

though the upper limit on T_n is -139.5 °C, there are no optimal solutions that correspond to T_n greater than -143.8 °C. As expected, the solution corresponding to 98% yield (or T_n = -156.5 °C) requires the highest amount of shaftwork (Figs. 8.18 and 8.22). In addition, as the yield decreases or T_n increases, the shaftwork requirement decreases; however, the work required by the fuel gas compressor increases (Fig. 8.23), because the decrease in yield means an increase in the amount of LNG vaporized due to the expansion across the multi-phase turbine. Therefore, it can be concluded that the shaftwork required by refrigeration dominates the Pareto-optimal solutions, and not the shaftwork required by the fuel gas compressor.

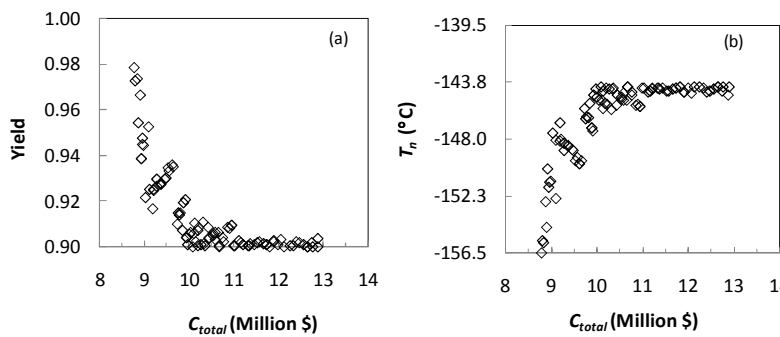


Fig. 8.22 Values of constraints corresponding to the Pareto-optimal solutions shown in Figs. 8.18 and 8.21: (a) yield of LNG; and (b) final refrigeration temperature, T_n.

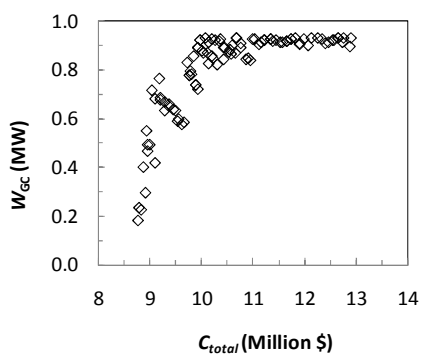


Fig. 8.23 Amount of shaftwork required in the fuel gas compressor corresponding to the Pareto-optimal solutions shown in Fig. 8.18.

8.4.3 *Discussion*

The MOO of the nitrogen cooling and the dual independent expander process for LNG provides significant insight into the optimal range of the objectives and the design and operating variables associated with these processes. Although the number of variables in the MOO is small, they are critical in the overall process economics. Targeting of any process provides an initial screening of the available design options. The MOO study can help with the screening of the design options and with selecting the best option for the detailed design. The designer remains in control of the decision making process and can choose the lowest capital cost, or the lowest operating cost, or the one which gives a perfect match of the composite curves (high thermal efficiency), in conjunction with other considerations and/or experience.

The computational time taken in the nitrogen case study for a population size of 100 and 400 generations was nearly three hours on a Pentium 4, 3.2 GHz personal computer with 1 GB RAM. In the case of the dual independent expander cycle for LNG, the computational time for a population size of 100 and 800 generations was close to twelve hours. The time taken for the latter problem is considerably longer due to the large number of generations. The LNG case consisted of twelve different flowsheets (compared to one flowsheet in the nitrogen case), and these flowsheets are opened and optimized simultaneously, which requires a lot of memory and also slows down the process simulation and optimization. The computational time could be improved by implementing all the twelve flowsheets as sub-flowsheets in one HYSYS simulation file.

8.5 Conclusions

In this chapter, two multi-stage gas-phase refrigeration processes: nitrogen cooling and the dual independent expander refrigeration process for LNG were successfully optimized for two objectives using MPMLE, HYSYS and NSGA-II. In the first of these, the design problem of

nitrogen cooling using nitrogen refrigerant was optimized by minimizing the total shaftwork requirement and the total heat exchanger area. Interestingly, only 4 or 7 refrigeration stages give the optimal solutions; four stages require higher energy input, but at the same time require lower heat exchanger area; seven refrigeration stages are required for reducing the total shaftwork. In the second case study, the dual independent expander refrigeration process was optimized for minimizing the capital cost of the major equipment items and the shaftwork requirement. It was found that two and three natural gas stages are necessary for a better match of composite curves. Two natural gas stages require higher shaftwork (but lower capital cost) than three natural gas stages. There were no optimal solutions with one natural gas stage because of either high shaftwork requirements or the problem of internal pinching. Interestingly, only one nitrogen refrigeration stage gives optimal solutions. The maximum possible yield of LNG was 98%, which required the highest amount of shaftwork but, at the same time, required the lowest capital investment.

Although to an experienced LNG process engineer some of the conclusions from this work may be obvious, the results are generated over a wider range of conditions than would normally be tested in a simple sensitivity analysis around a specific reference case, and therefore there is more confidence that the true Pareto-optimal curve has been obtained. Furthermore, this curve is generated automatically, rather than using an adhoc trial-and-error approach. In both the case studies, the MPMLE was used successfully as an interface to optimize the process simulated in HYSYS using NSGA-II. In the second case, 12 different HYSYS flowsheets were created and the MPMLE interface allowed the selection of the correct flowsheet based on the decision variables generated from NSGA-II. This case demonstrates that the MPMLE interface in conjunction with HYSYS and NSGA-II promises to be a powerful tool for optimizing process flowsheets for multiple objectives.

Acknowledgements

Authors thank Mr. Armando Guerrero at Mafi-Trench Corporation for providing the cost data for turbo-expanders. We would also like to thank Dr. Naveen Bhutani for his assistance with the use of the MPMLE interface.

Nomenclature

Roman Symbols

A	Heat exchanger area [m^2]
A_{total}	Total heat exchanger area [m^2]
C	Cooler
C_{comp}	Cost of compressor [US$]
C_{EXP}	Cost of turbo-expander [US$]
C_{GC}	Cost of fuel gas compressor [US$]
C_{HE}	Cost of heat exchanger [US$]
C_{MPT}	Cost of multi-phase turbine [US$]
C_{total}	Total capital cost [US$]
E	Expander or turbo-expander
E_{MPT}	Multi-phase turbine
H	Auto refrigeration heat exchanger
K	Compressor
K_{GC}	Fuel gas compressor
n	Number of stages
P_0	Refrigerant pressure at the expander inlet [MPa]
P_1	Refrigerant pressure at the expander outlet [MPa]
P_{MPT}	Pressure at the outlet of the multi-phase turbine [MPa]
T	Temperature [°C]
$T_{desired}$	Temperature which is ΔT_{min} colder than T_n [°C]
T_i	Intermediate temperature [°C]
T_{in}	Temperature at the inlet of the expander [°C]
T_n	Final process stream temperature [°C]
T_0	Initial process stream temperature [°C]
T_{out}	Temperature at the outlet of expander [°C]
U	Overall heat transfer coefficient [$kW/m^2°C$]
W_{in}	Work supplied to the compressor [kW]

W_{MPT}	Work recovered from the multi-phase turbine [kW]
W_{out}	Work recovered from expander [kW]
W_{total}	Net Work required from the process [MW]

Greek Symbol

$\Delta T_{\min}$	Minimum temperature difference between hot and cold streams [°C]

Subscripts

i, j	Indices for refrigeration stages
N2	Refers to nitrogen refrigeration stages
NG	Refers to natural gas refrigeration stages

References

Bhutani, N., Tarafder, A., Rangaiah, G. P. and Ray, A. K. (2007), *A Multi-Platform, Multi-Language Environment for Process Modeling, Simulation and Optimization*, Int. J. of Comput. Appl. in Technology, In press.

Calise, F., Dentice d' Accadia, F., Vanoli, L. and Michael R. von Spakovsky (2007), *Full Load Synthesis/design Optimization of a Hybrid SOFC-GT Power Plant*, Energy, 32, pp. 446-458.

Deb, K., Pratap, A., Agarwal, A. and Meyarivan, T. (2002), *A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II*, IEEE Trans. on Evolutionary Computation, 6 (2), pp. 182-197.

Diaz, M. S., Serrani, A., Bandoni, J. A. and Brignole, E. A. (1997), *Automatic Design and Optimization of Natural Gas Plants*, Ind. Eng. Chem. Res., 36, pp. 2715-2724.

Duvedi, A. and Achenie, L. E. K. (1997), On the Design of Environmentally Benign Refrigerant Mixtures: A Mathematical Programming Approach. Comput. Chem. Eng., 21(8), pp. 915-923.

Foglietta, J. H. (2002), LNG Production Using Dual Independent Expander Refrigeration Cycles, US Patent 6,412,302 B1, July 2, 2002.

Guerrero, A (2007), Personal communications with Armando Guerrero, Mafi-Trench Corporation, USA, June 2007.

Houser, C. and Krusen, L. (1996), *The Phillips Optimized Cascade Process*, 17th International LNG/LPG Conference, Vienna, Dec. 3-6.

Jang, W-H., Hahn, J. and Hall, K. R. (2005), Genetic/quadratic Search Algorithm for Plant Economic Optimizations Using a Process Simulator, Comput. Chem. Eng., 30, pp. 285-294.

Konukman, A. E. S. and Akman, U. (2005), Flexibility and Operability Analysis of a HEN-integrated Natural Gas Expander Plant, Chem. Eng. Sci., 60, pp. 7057-7074.

Lee, G. C., Smith, R. and Zhu, X. X. (2002), *Optimal Synthesis of Mixed-refrigerant Systems for Low-Temperature Processes,* Ind. Eng. Chem. Res, 41, pp. 5016-5028.

Polley, G. T. (1993), *Heat Exchanger Design and Process Integration*, Chem. Eng., 8, pp.16.

Shah, N. M. and Hoadley, A. F. A. (2007), *A Targeting Methodology for Multistage Gas-Phase Auto Refrigeration Processes*, Ind. Eng. Chem. Res., 46(13), pp. 4497-4505.

Shelton, M. R. and Grossmann, I. E. (1986), Optimal Synthesis of Integrated Refrigeration Systems. I: Mixed-integer Programming Model, Comput. Chem. Eng., 10, pp. 445-459.

Shukri, T. (2004), *LNG Technology Selection*, Hydrocarbon Engineering, 9(2), pp. 71-76.

Swenson, L. K. (1977), Single Mixed Refrigerant Closed Loop Process for liquefying Natural Gas, US Patent, 4,033,73.

Vaidyaraman, S. and Maranas, C. D. (1999), Optimal Synthesis of Refrigeration Cycles and Selection of Refrigerants, AIChE J, 45, pp. 997-1017.

Wu, G. and Zhu, X. X. (2001), *Retrofit of Integrated Refrigeration Systems*, Trans. IChemE, 79, Part A., pp. 163-181.

Wu, G. and Zhu, X. X. (2002), *Design of Integrated Refrigeration Systems*, Ind. Eng. Chem. Res., 41, pp. 553-571.

Exercises

In this section, four exercise problems are given based on the gas-phase refrigeration system discussed in this chapter. The degree of difficulty increases from problem 1 to 4. Readers can tackle the first three problems with any one of the following three assumptions: (i) a constant C_P value – a very simplified approach; (ii) C_P is a function of temperature only (i.e., neglect the pressure dependence of C_P) – a reasonable approximation; and (iii) use the enthalpy and C_P correlations given below – very close to the real case. The solutions to problems 1 and 2 are available in the file: GasPhaseRefrigeration.xls in the folder: Chapter 8 on the CD.

Nitrogen is used as the refrigerant in all the following exercise problems. The enthalpy and the specific heat data for nitrogen at five values of pressure within the temperature range of -140 °C to 200 °C have been fitted with the following correlations:

$$H = a_1\, e^{b_1 T} + c_1\, e^{d_1 T} \qquad\qquad C_P = a_2\, e^{b_2 T} + c_2\, e^{d_2 T}$$

The values of the coefficients in these correlations can be found in Tables 8.E1 and 8.E2. All these data are also available in the GasPhaseRefrigeration.xls on the CD. These correlations for enthalpy and specific heat are required for answering the following exercises. They can also be used for calculating a C_P value (say, at mean pressure and temperature in the refrigeration system) for use as a constant, and also for finding a correlation as a function temperature (say, at mean pressure in the refrigeration system).

Table 8.E1 Coefficients in the correlation for enthalpy of nitrogen at different pressures.

Enthalpy (kJ/kmol) at P =					
	1000 kPa	2000 kPa	3000 kPa	4000 kPa	5000 kPa
a1	8609	5362	3743	2548	1819
b1	0.001625	0.002456	0.003337	0.00456	0.00586
c1	-9422	-6253	-4702	-3547	-2833
d1	-0.00167	-0.00275	-0.00394	-0.0056	-0.00742

Table 8.E2 Coefficients in the correlation for specific heat of nitrogen at different pressures.

Specific Heat (kJ/kmol °C) at P =					
	1000 kPa	2000 kPa	3000 kPa	4000 kPa	5000 kPa
a2	0.1144	0.08921	0.01353	3.19E-05	0.000348
b2	-0.02388	-0.03335	-0.0539	-0.107	-0.0919
c2	29.54	30.27	31.3	32.78	33.59
d2	0.00011	-3.7E-06	-0.0002	-0.00054	-0.00067

The efficiency of the expanders and compressors can be assumed to be $\eta_{isen} = 0.8$ and $\eta_P = 0.83$ respectively.

1. It is required to cool a 1000 kmol/hr of nitrogen gas stream at 2000 kPa from 30 °C to -17 °C using a single stage gas-phase refrigeration process shown in Fig. 8.E1.

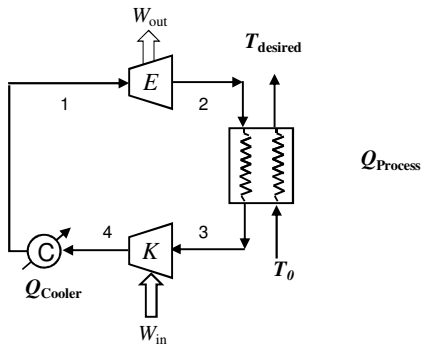


Fig. 8.E1 A single stage gas-phase refrigeration process.

(a) Assume that the temperature and pressure of the refrigerant at point 1 are 30 °C and 5000 kPa respectively. Using the following equation, calculate the temperature at point 2 if the refrigerant gas is expanded to 2000 kPa across the turbo-expander:

$$T_2 = T_1\left[\left(\phi^{\frac{\gamma-1}{\gamma}} - 1\right)\eta_{isen} + 1\right]$$

where $\phi = P_{out}/P_{in}$, $\gamma = C_P/(C_P - R)$ and R is the gas constant.

(b) Calculate the $\Delta T_{\min}$ required for achieving the desired product temperature. (*Hint:* $T_{desired} = T_2 + \Delta T_{\min}$).

(c) Using energy balance across the heat exchanger, calculate the refrigerant flow rate (m_{ref}) if a constant $\Delta T_{\min}$ is maintained throughout the exchanger.

(d) Calculate the work recovered from the gas expansion ($W_{out} = m_{\text{ref}} \Delta H$).

(e) Assume the pressure at point 3 is the same as that at point 2. Calculate the temperature at point 4 if the refrigerant gas is compressed back to the original pressure, and also the work required in the compressor using the following equations:

$$T_4 = T_3\left[\left(\phi^{\frac{\gamma-1}{\gamma}} - 1\right)\frac{1}{\eta_P} + 1\right] \quad \text{and} \quad W_{in} = \frac{m_{ref} C_{P,in} T_{in}\left(\phi^{\frac{\gamma-1}{\gamma}} - 1\right)}{\eta_P}$$

where $C_{P,in}$ is the specific heat of nitrogen calculated at the inlet conditions of the compressor (i.e., point 3).

(f) Calculate the duty of the cooler (Q_{Cooler}) if the refrigerant is cooled to its original temperature of 30 °C.

(g) Calculate heat exchanger area assuming U = 0.3 kW/m^{2}C and using

$$A = \frac{Q_{\text{Process}}}{U\Delta T_{\min}}$$

where Q_{Process} is the heat duty associated with the heat exchanger.

2. Specific heat, C_P of nitrogen does not vary significantly with pressure for temperatures above 5 °C, and can be taken as independent of pressure. So, use P_1 ($3000 \le P_1 \le 5000$) and $\Delta T_{\min}$ ($1 \le \Delta T_{\min} \le 6$ °C) as decision variables, and also the values of the coefficients at 5000 kPa given in Tables 8.E1 and 8.E2 to calculate the properties at P_1. Optimize the gas-phase refrigeration system for the same process conditions as in exercise 1 (i.e., cooling a 1000 kmol/hr of nitrogen gas stream at 2000 kPa from 30 °C to -17 °C, and temperature and pressure of the refrigerant at point 1 are 30 °C and 5000 kPa respectively), using the Excel® Solver, for each of following two objectives individually.

   (a) Minimization of total shaftwork requirement
   (b) Minimization of heat exchanger area.

   Hint: instead of fixing the value of $T_{desired}$, you can put a tolerance (e.g., 1 °C) on it. Repeat the above optimization for P_2 = 1000 kPa and 3000 kPa to understand the effect of pressure ratio on the optimization results.

3. Optimize the design problem in the previous exercise for two simultaneous objectives: minimization of total shaftwork and minimization of total capital cost of major equipment items (compressor, turbo-expander and heat exchanger). The correlations for the capital cost of these can be taken from Section 8.4.2 of this

chapter. As the Excel® Solver is only for single objective optimization, use either the ε-constraint method or a robust optimization code like NSGA-II for MOO.

4. It is required to cool a 1000 kmol/hr of nitrogen gas stream at 2000 kPa from 30 °C to -40 °C using the two-stage gas-phase refrigeration system shown in Fig. 8.E2. Assuming fixed values of P_0 = 4000 kPa and P_1 = 2000 kPa, optimize this two-stage system for minimization of total shaftwork and minimization of total capital cost of major equipment items (compressor, turbo-expander and heat exchanger). The correlations for the capital cost of these can be taken from Section 8.4.2. Firstly, optimize the problem for each objective individually, and then optimize the problem for the two objectives simultaneously using the ε-constraint method. Try using different sets of the values of P_1 and P_2, and analyze the effect of pressure ratio.

Suggested steps for simulating the process in Fig.8.E2:

(a) First step is to calculate temperatures at the exit of both the turbo-expanders (points 3a and 2b). Assume that a constant $\Delta T_{\min}$ is maintained throughout the heat exchanger.

(b) Set up the spreadsheet for the second stage calculations first, and then do the calculations for the first stage.

(c) The supplementary flow needed in the second stage (and taken from the first stage, at point 2a) can be calculated from the equation:

$$m_{ref} C_{P,warm} = (m_{ref} + m_{suppl}) C_{P,cold}$$

where $C_{P,warm}$ and $C_{P,cold}$ denote the average specific heat on the warm and the cold refrigerant side respectively. Note that the supplementary flow causes imbalance in the flow calculation, and so the loops in Fig. 8.E2 are not closed. However, this would not affect the optimization as the temperature and pressure at 6b or 4a are the same. For simplicity, two different compressors and coolers have been shown. Ideally, one compressor and one cooler would suffice for any number of stages – as long as the temperatures and pressures are the same; and we are calculating the compressor power individually and then adding them to calculate the total power.

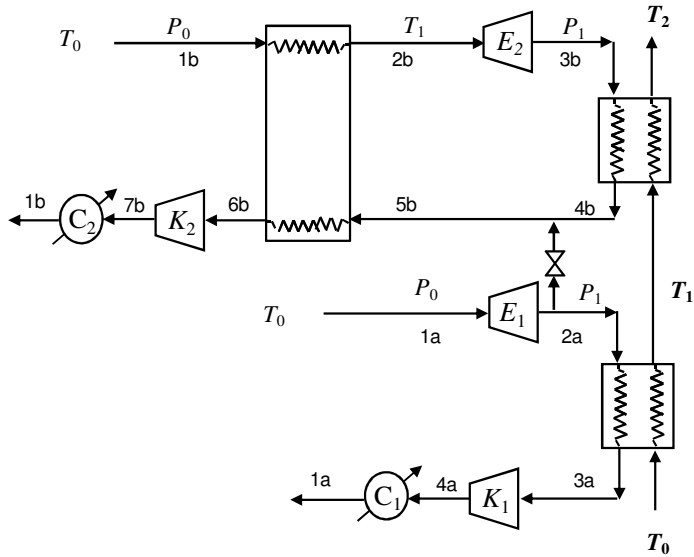


Fig. 8.E2 A two-stage gas-phase refrigeration process.

Chapter 9

Feed Optimization for Fluidized Catalytic Cracking using a Multi-Objective Evolutionary Algorithm

Kay Chen Tan*, Ko Poh Phang and Ying Jie Yang
Department of Electrical & Computer Engineering
National University of Singapore, Singapore

*Corresponding author; e-mail: eletankc@nus.edu.sg

Abstract

Feed optimization in the fluidized catalytic cracking (FCC) process is a prominent chemical engineering problem, where the objective is to maximize the production of high-quality gasoline stocks at a low energy consumption level. However, the various feeds, based on the density and volumetric flow rate of its constituent stream, are conflicting in nature and subjected to many practical constraints. As such, this chapter presents the application of a multi-objective evolutionary algorithm (MOEA) which will simultaneously optimize the various flow streams in a FCC feed surge drum of a local refinery. An interactive Graphical User Interface (GUI) based MOEA toolbox developed by the authors is used as the platform for optimization. The various trade-off surfaces between the different objectives evolved by the MOEA provide further insights to this problem and allow more optimal choices during the decision making process. Lastly, a performance comparison based on several key performance indexes shows that the overall economic gain offered by MOEA optimization against the conventional approach like linear programming is significantly higher.

Keywords: Multi-objective Evolutionary Algorithm, Feed Optimization, Fluidized Catalytic Cracking

9.1 Introduction

Most industry engineers believe that improving efficiency and increasing profitability of existing plants can be achieved by effective optimization. As a matter of fact, given today's emphasis on the conservation of natural resources and reduction of energy consumption, the motivation of using optimization is indeed compelling as it will provide the users a competitive advantage over their competitors. Typically, optimization involves the maximization or minimization of a cost function (sometimes unknown) that represents the performance of some systems.

In any crude oil refinery business, the main objective is to extract as many light-end products as possible from the crude oil, as this will translate to a higher revenue and hence profit margin. In some advanced refineries, the residue from the initial process of vacuum distillation will be further cracked to generate more products. This process is known as fluidized catalytic cracking (FCC) where heavy hydrocarbon components are cracked into lighter compounds under high temperature. Inevitably, this process requires relatively high energy input which will therefore result in higher operating cost. As such, to generate the maximum profits from FCC, the feeds should be optimized to produce large quantity of gasoline stocks at superior quality with low energy consumption. However, this optimization problem is non-trivial as it includes several conflicting objectives and is subjected to certain constraints.

This chapter considers the practical problem of feed optimization of the FCC process in a local refinery. A thorough study of the FCC process and the formulation of the feed optimization problem will be presented. Accounting for the conflicting nature between the various objectives, the FCC feed optimization is solved by a multi-objective evolutionary toolbox developed by the authors (Tan *et al.,* 2001a). Finally, for the actual implementation of the FCC process, three major key performance indexes are adopted to rate the various solutions evolved, so as to facilitate the decision-making process.

The rest of the chapter is organized as follows. Section 9.2 describes the FCC process in detail and the corresponding multi-objective mathematical model. The multi-objective evolutionary toolbox used as the optimization tool in this work is presented in Section 9.3, and the results obtained from the evolutionary design are discussed in Section 9.4. Section 9.5 involves the decision-making and economic evaluation of the solutions for practical implementation. Finally, conclusions are drawn in Section 9.6.

9.2 Feed Optimization for Fluidized Catalytic Cracking

9.2.1 *Process Description*

In typical petroleum refineries, crude oil is first refined through the distillation, which will separate gasoline from the other components by heating crude oil under pressure to vaporize gasoline. The products drawn from the top of the tower are the light-end products while the bottom products are the heavy products. The latter usually goes through another process called vacuum distillation, which will further separate heavy products in vacuum and at high temperature. Consequently, this process will generate other valuable light-end and heavy products. The resultant heavy products from this process will be residues of extremely high viscosity. These residues are regarded as unwanted remains and are usually sold as asphalt, tar or bitumen at a relatively low price.

In some advanced refineries, these residues will be cracked to light-end products. They will be fed through the FCC process to produce high-quality gasoline stock. This catalytic process utilizes high temperature to break (or crack) heavy hydrocarbon components into lighter compounds. Unfortunately, this process requires relatively higher energy input which will increase the overall operating cost. As such, optimization of the FCC process is vital for the improvement of the profit margin because well-optimized feeds can significantly reduce energy consumption while producing high quality and quantity gasoline stocks.

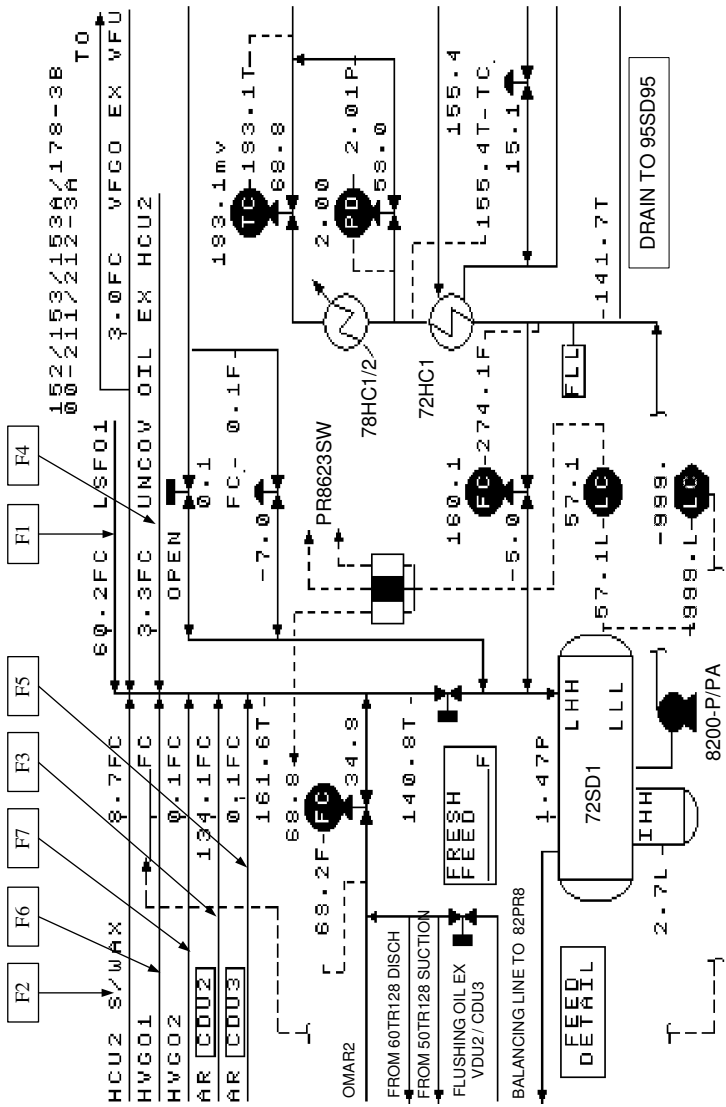


Fig. 9.1 The Schematic Layout of the FCC Feed Surge Drum (72SD1).

The schematic layout of the FCC feed surge drum (72SD1) considered in this work is illustrated in Fig. 9.1. Altogether, there are a total of 7 different feed streams:

- F1 - Low Sulfur Fuel Oil 1 (LSFO 1)
- F2 - Hydro Cracker Unit 2 Slop Wax (HCU 2 slop wax)
- F3 - Atmospheric Residue from Crude Distillation Unit 2 (AR-CDU 2)
- F4 - Unconverted Oil Ex Hydro Cracker Unit 2 (Uncov oil Ex HCU2)
- F5 - Atmospheric Residue from Crude Distillation Unit 3 (AR-CDU3)
- F6 - Heavy Vacuum Gas Oil 1 (HVGO1)
- F7 - Heavy Vacuum Gas Oil 2 (HVGO2)

Each of these feed streams has its own controllable flow range and specific density range as shown in Table 9.1. The flow rate of the seven flow streams is controlled individually based on certain pre-calculated ratio. At every 100-hours interval, the level in 72SD1 will build up to a physical level of around 85%. During this time, the mixture will settle down and any condensate of water will be collected at the water-boot of the drum. At the 85% physical level, a high level alarm will be triggered to activate the 8200-P/PA pump. A level controller is also incorporated in this drum for monitoring the level to prevent it from overfilling. The 8200-P/PA pump will direct the mixture to the 72CC FCC reactor for cracking via two heat exchangers, 72HC1 and 78HC1/2, until the level hits the low level setting that will automatically stop the pump and re-initiate the whole process. At the heat exchangers, the mixture will be heated up to the temperature of about 190°C. A temperature controller and a pressure controller at the downstream of the line are used to fine tune and maintain the corresponding settings.

Table 9.1 Details of Streams to the FCC Feed Surge Drum.

Tag no.	Description	Flow range (metre cubic/h)	Specific Density range (kilogram/litre)
FC-872C	LSFO 1	F1: 13 to 17	D1: 0.90 to 0.94
FC-811E	HCU2 slop wax	F2: 90 to 120	D2: 0.89 to 0.92
FC-223E	AR-CDU2	F3: 135 to 150	D3: 0.85 to 0.89
FC-511T	Uncov oil Ex HCU2	F4: 30 to 50	D4: 0.91 to 0.95
FC-P124	AR-CDU3	F5: 85 to 120	D5: 0.85 to 0.89
FC-Z281	HVGO 1	F6: 110 to 150	D6: 0.95 to 0.98
FC-8P2C	HVGO 2	F7: 150 to 270	D7: 0.94 to 0.97

9.2.2 *Challenges in the Feed Optimization*

In order to optimize the FCC process, the initial configuration of the various feeds based on their temperature, specific density, viscosity and composition in the total volumetric flow is extremely crucial. Neglecting any of these factors will result in substantial losses in terms of energy, raw material, resources and time. In extreme cases where the parameters are outside their operating range, this might even damage the equipment and pipe lines, which will consequently lead to an unsafe working environment.

Due to the complexity of the FCC process and the presence of different interacting components of the flow stream, the optimization task to determine the amount of feed based on volumetric flow rate of each stream and their individual specific density is an extremely challenging task. The ideal values of these parameters are affected by the following factors:

- Type of feeds coming in which greatly depends on the type of crude being processed. Low quality crude produces low end products.
- Individual feed processing cost and its market value. The energy intake, which has direct relationship with cost, varies significantly with the type of products being produced. Light-end products have high market value but costs more energy to produce, which is in contrary to the heavy product.
- The resultant specific density of the combined feeds. This is a vital parameter the engineer needs to look into. The reaction process in the reactor is very sensitive to the resultant specific density of the final feed into it. There is a specific range of density it can process effectively and efficiently.
- Pressure and temperature of the individual feeds. These factors will affect the specific density of the mixture and thus directly affect the reaction process.
- Piping size and length for individual feeds and the final mixture.

With these factors in consideration, the feed optimization will be conducted and the solutions obtained will be used as the set points for the individual controllers to control the respective parameters. This

optimization process to determine the initial flow configuration is necessary during the start-up phases or whenever there is a change of crude oil, as the actual characteristics and components of the residues are uncertain.

Presently, the Honeywell DCS System adopts linear programming (LP) as the optimization tool for the initial setup of the controllers (Dantzig and Thapa, 1997; Foulds, 1981) before the main Online Distributed Control System takes over subsequently. Due to the non-linearity of some of the flow characteristics and the constraints on the combination ratio, assumptions have to be made on certain parameters and factors for the LP model adopted. As a result, the accuracy of the results will be limited to the realism and practicality of the assumptions and models.

9.2.3 *The Mathematical Model of FCC Feed Optimization*

The mathematical formulation of the FCC feed optimization is as follows (BP Amoco Group, 1991). Altogether, it consists of three different objectives to be maximized and is subjected to four constraints. The details of the derivation for these equations are complex and will not be revealed as it is considered a trade secret by the organization.

9.2.3.1 *Objective Functions*

1. Maximize *Fmx*, which consists of certain ratio of the waxy feed and feeds with low *specific density*. This combination is to reduce the viscosity and thus reduce the *flash-point* of the waxy feed so that the reaction process will consume less energy.

$$Fmx = (8\sqrt{F_2} + 5F_3 + 7\sqrt{F_5}) / F_t ,$$
$$F_t = \sum_{i=1}^{7} F_i . \tag{9.1}$$

2. Maximize *Dif*, the differentiated total flows of F6 and F7. The complexity here is due to the very heavy viscosity of these feeds. Reaction process will be slow when viscosity is high. However with the right temperature it will overcome this constraint. Thus

temperature is one of the main factors taken into consideration here.

$$Dif = \frac{3 \cdot [(F_6 + F_7) + 0.0001]^2 - 3 \cdot (F_6 + F_7)^2}{0.0001} \tag{9.2}$$

3. Maximize *Fmi***,** which is related to a ratio of F1, F4 and *Dt*. F1 feed is high in sulfur that will form acid gas, which causes serious corrosion of pipes and extremely hazardous to people when inhaled. By optimizing this formula the unconverted oil of F4 and *Dt* will reduce the acidity in some way and make it feasible for the reaction to take place at the same time.

$$\begin{aligned} Fmi &= \frac{6F_1 + 12F_4}{0.2 \sin Dt} \\ Dt &= \sum_{i=1}^{7} F_i \cdot d_i \Big/ \sum_{i=1}^{7} F_i \end{aligned} \tag{9.3}$$

9.2.3.2 *Constraints*

1. The range of the individual feed volumetric flow rate as shown in Table 9.1.
2. The range of the specific density of the individual feed as shown in Table 9.1.
3. The resultant specific density, *Dt*, of the final mixture should be less than 0.92 kg/cm^2.

$$Dt = \sum_{i=1}^{7} F_i \cdot d_i \Big/ \sum_{i=1}^{7} F_i \le 0.92 \tag{9.4}$$

4. The level, *Lv*, of tank 72SD1 must be controlled at 85% level.

$$Lv = \frac{100F_t}{\pi \cdot 20^2} + 30 \le 85 \tag{9.5}$$

9.3 Evolutionary Multi-Objective Optimization

Many practical problems require the simultaneous optimization of several non-commensurable and often competing objectives. The

contradictory nature of the objective functions makes it impossible to find a single solution that is optimal for all the objectives simultaneously. Often, instead of a single optimum, there will be a set of alternative trade-offs, known as Pareto-optimal set. These solutions are optimal in the sense that no improvement is possible for any objective function without sacrificing at least one of the other objectives (Goldberg and Richardson, 1987; Horn and Nafpliotis, 1993; Srinivas and Deb, 1994).

Emulating the Darwinian-Wallace principle of "survival-of-the-fittest" in natural selection and adaptation, an evolutionary algorithm represents a class of stochastic optimization methods, widely proven as a general, robust and powerful search mechanism. In an evolutionary algorithm, natural selection is simulated by a stochastic selection process, where better solutions are assigned a higher chance to survive and reproduce. Reproduction is then performed via recombination and mutation where new solutions are generated, imitating natural capability of creating "new" living beings. Repeated iterations of selection and reproduction will result in a pool of solutions that are well suited for the optimization problem.

Moreover, an evolutionary algorithm seems to be especially suited to multi-objective optimization (MOO) due to their ability to capture multiple Pareto-optimal solutions in a single run and may exploit similarities of solutions by recombination. As such, multi-objective evolutionary algorithm (MOEA) has been gaining significant attention from researchers in various fields as more and more researchers validate their efficiency and effectiveness in solving sophisticated multi-objective problems where conventional optimization tools fail to work well. Corne *et al.* (2003) argued that "single-objective approaches are almost invariably unwise simplifications of the real-problem", "fast and effective techniques are now available, capable of finding a well-distributed set of diverse trade-off solutions, with little or no more effort than sophisticated single-objective optimizers would have taken to find a single one", and "the resulting diversity of ideas available via a multi-objective approach gives the problem solver a better view of the space of possible solutions, and consequently a better final solution to the problem at hand".

There have been many surveys on evolutionary techniques for MOO (Fonseca and Fleming, 1995; Coello Coello, 1998; Van Veldhuizen and Lamont, 2000; Tan *et al.*, 2002; Chapter 3 in this book). While conventional methods combined multiple criteria to form a composite scalar objective function, modern approach incorporates the concept of Pareto optimality or modified selection schemes to evolve a family of solutions at multiple points along the tradeoffs simultaneously (Tan *et al.,* 2002).

Although MOEA is powerful for MOO, users often require certain programming expertise with considerable time and effort in order to write a computer program for implementing the often sophisticated algorithm to meet their need. As a result, implementation could be tedious and needs to be done before users can start their design task for which they should really be engaged in. Tan *et al.* (2001a) presented a global optimization toolbox that is built upon the MOEA (Tan *et al.,* 1999, 2003) to address the need of a more user-friendly and comprehensive MOEA toolbox for MOO. It is ready for immediate use with minimal knowledge in MATLAB or evolutionary computing.

The MOEA toolbox is fully equipped with interactive GUIs and powerful graphical displays for ease-of-use and efficient visualization of different simulation results, hence providing excellent supports for decision-making and optimization in complex real-world optimization applications. It is also designed with many useful features such as the goal and priority settings to provide better support for decision-making in MOO (Tan *et al.*, 2003), dynamic population size that is computed adaptively according to the online discovered Pareto-front (Tan *et al.,* 2001b), soft/hard goal settings for constraint handlings, multiple goals specification for logical "AND"/"OR" operation, adaptive niching scheme for uniform population distribution, and a useful convergence representation for MOO. Furthermore, the toolbox contains various analysis tools for users to compare, examine or analyze the different results or trade-offs anytime during the optimization. The overall architecture for the MOEA toolbox is shown in Fig. 9.2.

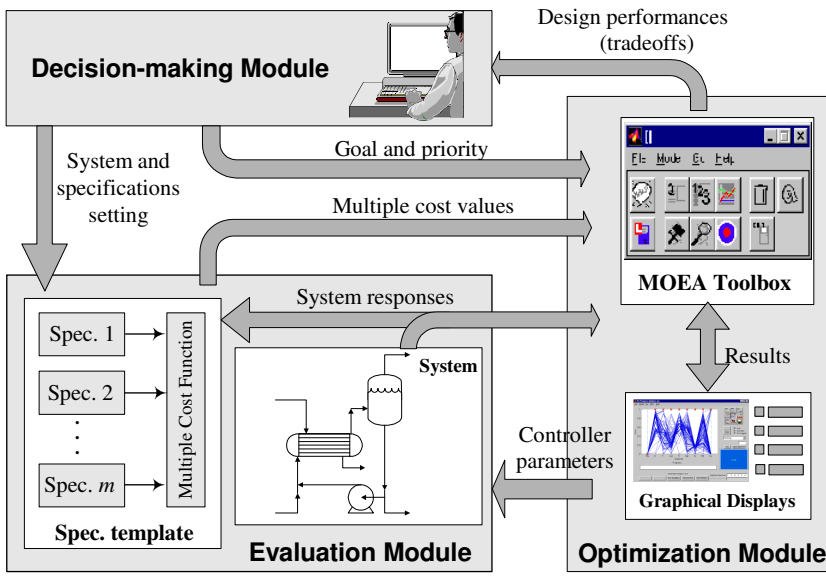


Fig. 9.2 The evolutionary optimization architecture.

Constraints often exist in practical optimization problems. These constraints can be incorporated in the MOO as objective components to be optimized. A constraint could be "hard" where the optimization is directed toward attaining a threshold or goal, and further optimization is meaningless or not desirable if the constraint has been violated. In contrast, a "soft" constraint requires that the value of the objective component corresponding to the constraint is optimized as much as possible. An easy approach to deal with both hard and soft constraints concurrently in evolutionary MOO was proposed by Tan *et al.* (1999, 2003). At each generation, an updated objective function $\boldsymbol{F}_x^{\#}$ concerning both hard and soft constraints for an individual x with its objective function $\boldsymbol{F}_x$ can be computed before the goal-sequence domination scheme as given by

$$\boldsymbol{F}_x^{\#}(i) = \begin{cases} \boldsymbol{G}(i) & \text{if } [\boldsymbol{G}(i) \text{ is hard}] \wedge [\boldsymbol{F}_x(i) < \boldsymbol{G}(i)], \\ \boldsymbol{F}_x(i) & \text{otherwise,} \end{cases} \quad \forall i = \{1, \ldots, m\}. \tag{9.6}$$

The *i*th objective component that corresponds to a hard constraint is assigned to the value of $\boldsymbol{G}(i)$ whenever the hard constraint has been satisfied. The underlying reason is that there is no ranking preference for any particular objective component that has the same value in an evolutionary optimization process, and thus the evolution will only be directed toward optimizing soft constraints and any unattained hard constraints.

9.4 Experimental Results

The mathematical model of FCC feed optimization described in section 9.2.3 was programmed in MATLAB model-files where the various constraints were formulated as objective components in the "hard" form. Applying the duality principle, the various maximization objectives were converted to minimization functions by simply negating them (Dantzig and Thapa, 1997; Deb, 2001). These model files were linked to the MOEA Toolbox to obtain the optimized values for the various control variables. LP was also applied and its solutions were used as a basis for comparison.

A population size of 100 was considered and the MOEA was run for 80 generations. Ten optimization runs were carried out to account for the stochastic nature of the evolutionary optimizers. Fig. 9.3 shows the objectives versus the costs for one of the best solutions attained by the MOEA toolbox. The decision to select the optimal one will be discussed later. Fig. 9.4 shows the evolutionary trace of the progress ratio (Tan *et al.*, 2001b) during one run. The progress ratio $Pr^{(n)}$ at generation n is defined as the ratio of the number of non-dominated solutions at generation n dominating the non-dominated solutions at generation (n-1) over the total number of non-dominated strings at generation n, accounting for the evolutionary progress towards the direction that is normal to the tradeoff surface formed by the current non-dominated solutions. It can be observed in Fig. 9.4 that the progress ratio of the optimization is relatively high and erratic at the initial stage and decreases asymptotically towards zero as the evolution proceeds, signifying closer proximity to the global tradeoff surface.

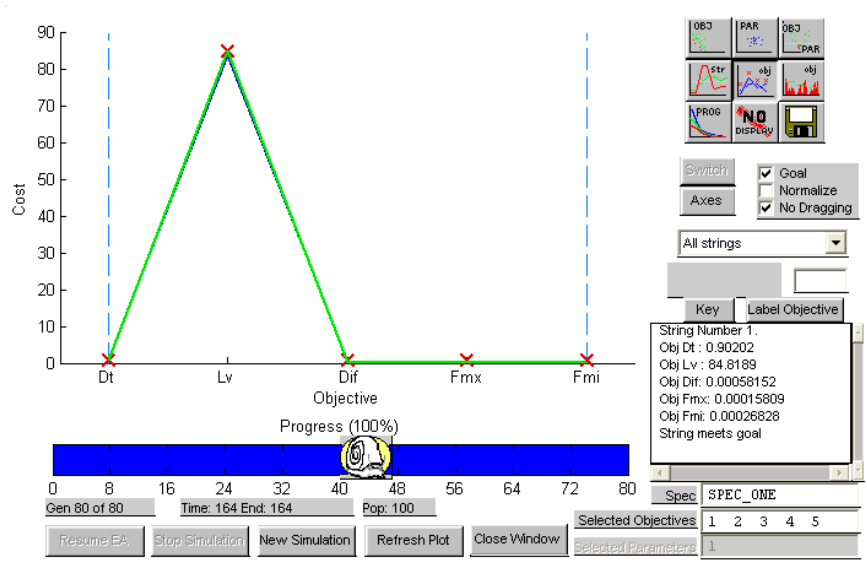


Fig. 9.3 The objective values of a solution.

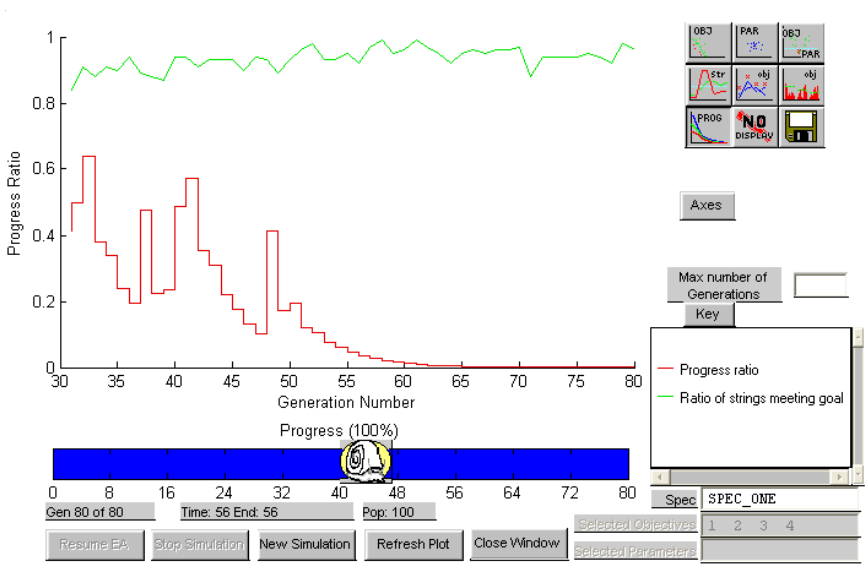


Fig. 9.4 Progressive ratio along the evolution.

To provide the decision maker insights into the characteristics of the problem before a final solution is chosen, eleven different solutions from Pareto optimal set were considered; Tables 9.2 and 9.3 list their objective and constraints values and their corresponding flows of the individual feeds respectively. For comparison, Tables 9.4 and 9.5 give the results obtained from linear programming in the Honeywell DCS system.

Table 9.2 Results of Selected Solutions from MOEA Toolbox.

String No.	Objectives to be maximized			Constraints	
	Fmx	*Dif*	*Fmi*	$Dt < 0.92$	$Lv < 85$
1-28	5049	1806	2912	0.905	84.61
2-2	5402	1810	3005	0.904	84.56
3-61	5666	1791	3112	0.904	84.72
4-18	5776	1780	3201	0.904	84.86
5-10	5817	1769	3222	0.903	84.38
6-13	6150	1727	3450	0.903	84.38
7-98	6354	1695	3768	0.903	84.59
8-99	6517	1662	4008	0.901	84.76
9-24	6652	1651	4120	0.902	84.95
10-44	6832	1638	4200	0.9	84.97
11-55	6890	1601	4252	0.904	84.51

Table 9.3 Individual Parameters of the Selected Solutions from MOEA Toolbox.

String No.	F1: 13~17 kl/h	F2: 90~120 kl/h	F3: 135~150 kl/h	F4: 30~50 kl/h	F5: 85~120 kl/h	F6: 110~150 kl/h	F7: 150~270 kl/h
1-28	16.765	99.375	135.200	47.311	86.677	119.712	181.264
2-2	14.413	99.596	135.200	41.053	93.671	119.633	182.105
3-61	16.490	107.908	135.209	41.043	88.425	116.308	182.225
4-18	16.684	405.794	135.468	41.073	93.678	136.831	159.858
5-10	16.284	100.700	135.394	34.011	102.030	113.623	181.266
6-13	15.485	99.385	135.023	41.383	104.268	115.904	171.906
7-98	16.413	10.900	135.800	46.918	111.174	116.316	160.762
8-99	16.485	97.800	135.394	41.053	112.751	113.620	168.907
9-24	16.288	107.698	135.403	46.617	107.783	115.904	160.799
10-44	16.413	100.700	135.200	47.291	118.177	112.050	160.891
11-55	16.698	107.575	135.118	47.344	111.527	113.621	153.151

Table 9.4 Result from Linear Programming.

Fmx	*Dif*	*Fmi*	*Dt* <0.92	*Lv* <85
5261	1803.5	3254	0.918902	84.938

Table 9.5 Individual parameter from Linear Programming.

F1: 13~17 kl/h	F2: 90~120 kl/h	F3: 135~150 kl/h	F4: 30~50 Kl/h	F5: 85~120 kl/h	F6: 110~150 kl/h	F7: 150~270 kl/h
15.142	112.102	140.144	39.189	92.489	125.781	195.483

Figures 9.5 to 9.7 plot the different trade-off surfaces among the various objectives. It can be observed from Figs. 9.5 and 9.6 that the objective *Dif* is conflicting to both the objectives *Fmx* and *Fmi* i.e. increasing *Fmx* and *Fmi* will cause *Dif* to decrease and vice versa. In contrast, the objectives of *Fmx* and *Fmi* are non-conflicting as shown in Fig. 9.7.

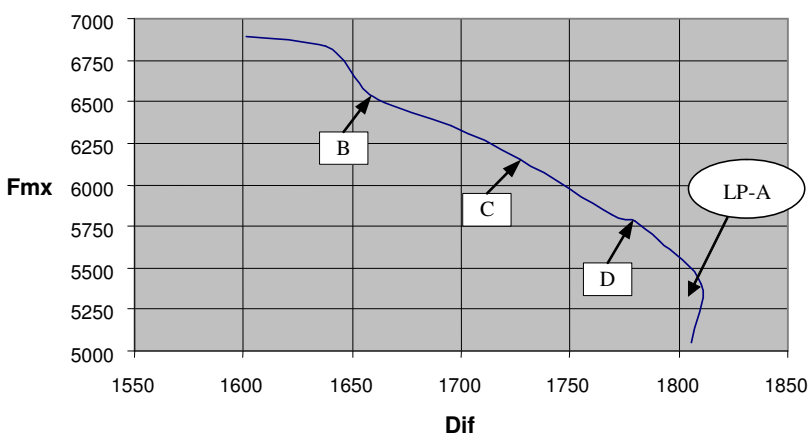


Fig. 9.5 Conflicting Objectives: *Fmx* Versus *Dif*.

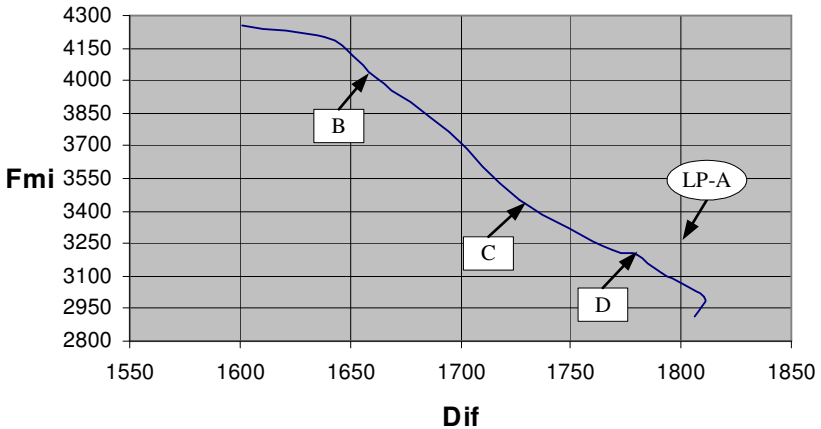


Fig. 9.6 Conflicting objectives: *Fmi* Versus *Dif*.

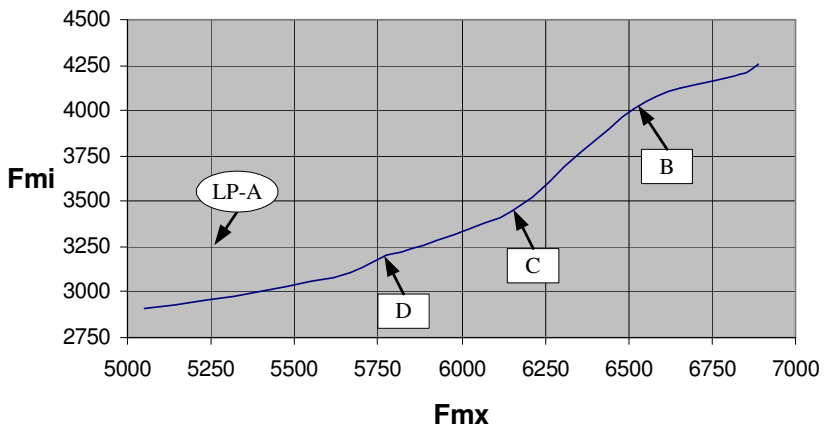


Fig. 9.7 Objectives: *Fmi* Versus *Fmx*.

9.5 Decision Making and Economic Evaluation

Although the multi-objective approach for optimizing the feeds allows us to appreciate the underlying trade-off among the various objectives, a solution has to be chosen ultimately for actual implementation. The

decision of selecting optimized point along the evolved trade-off surface will be based on three key performance indexes, set forth by the recommendations from Solomon Associates**,** a group of experienced petroleum industry experts that examines performance of the petrol chemical industries throughout the world and sets key performance indexes as the benchmark for performance assessments (Goodsell, 2002). The indexes are as follows:

1. Reactor Fuel Gas Consumption.
2. Reactor Steam Consumption.
3. Rate of Exothermic Reaction.

In the ideally optimized situation, the FCC process will utilize the minimal amount of steam and fuel gas for cracking, and yet at a high exothermic reaction rate. The high exothermic reaction rate will improve the overall efficiency of the process and provides the necessary energy gain. However, deciding the final solutions to be implemented from the set of optimal solutions is not a straightforward exercise. Various details need to be re-examined and the tradeoffs of each and every objective against the various factors are to be studied.

9.5.1 *Fuel Gas Consumption of Reactor 72CC*

Figure 9.8 shows the response of the fuel gas flow rate on "F.G. Line 1 and 2**"**. Four sets of parameters from the optimization curve were used to test the response of the FCC process at different time. The economic gain arising from the implementation of the different parameter sets are calculated and listed in Table 9.6. Clearly, the optimal parameter is solution C, where *Fmx* is 6150, *Dif* is 1727, and *Fmi* is 3450. The parameters derived from LP have the worst performance. Furthermore, LP only generates one solution, leaving no alternatives for the process engineer to work on. This is a major drawback especially in this MOO problem.

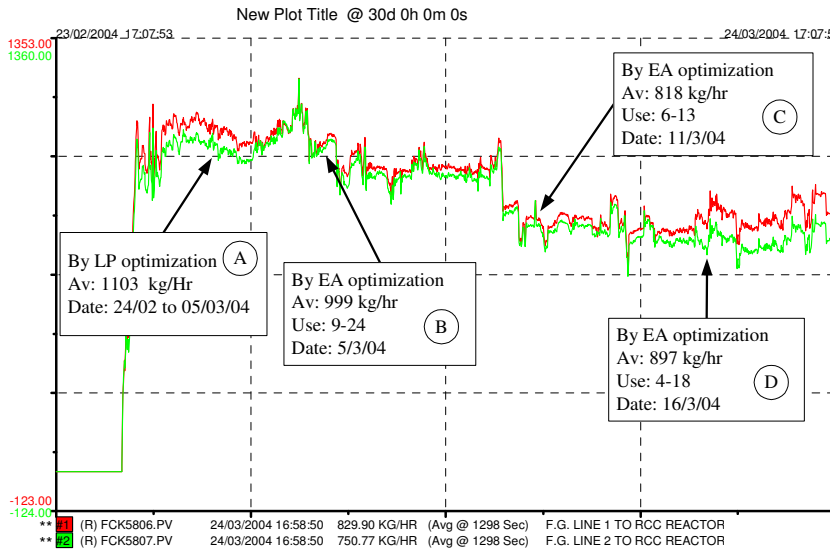


Fig. 9.8 The Fuel Gas Consumption of Reactor 72CC.

Table 9.6 Comparison of the cost of fuel gas consumption.

	A	B	C	D
Average Flowrate: X_1 in kg per hour	1103	999	818	897
Mass equivalent: $X_2 = X_1 * 2.2046$ lb	2432	2202	1803	1978
BTU equivalent: $X_3 = X_2 * 20500$	4984931	45149106	36968937	40539287
TFOE per hour: $X_4 = X_3 \div 38130000$	1.30735	1.18408	0.96955	1.06319
TFOE per month: $X_5 = X_4 * 24 * 30$	941	853	698	765
Total cost of fuel gas per month = X_5 * US$120 (1 TFOE = US$ 120)	US$112955	US$102305	US$83769	US$91859

Note: BTU- British Thermal Unit
TFOE - Ton-Fuel-Oil Equivalent.

9.5.2 *High Pressure (HP) Steam Consumption of Reactor 72CC*

Figure 9.9 shows HP steam consumption under different parameter settings of A, B, and C. Solution D was omitted as the differences between D and C are relatively small. For this performance index, solution A consumes the least steam while solution C the most. Here, the steam consumption of solution C was sacrificed to achieve more gains in the fuel gas and energy gain. The cost of steam consumption is calculated in Table 9.7.

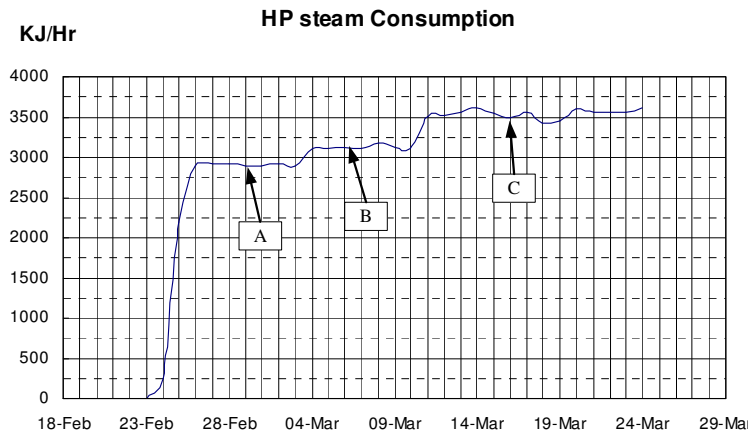


Fig. 9.9 HP Steam Consumption of Reactor 72CC.

Table 9.7 Comparison of the cost of HP steam consumption.

	A	B	C
Flow rate: X_1 = kJ per hour	2901	3107	3511
MBtu/hr equivalent: $X_2 = X_1 * (947.81 \times 10^{-6})$	2.745	2.945	3.328
TFOE equivalent: (1 TFOE = 6.05 MBtu/hr) $X_3 = X_2 \div 6.05$	0.454	0.487	0.550
(1 TFOE = US$ 21.86) $X_4 = X_3$ * US$ 21.86 per hr	9.93	10.64	12.02
Demineralized water cost: (US$1.3 per TFOE) $X_5 = X_3$ * US$1.3	0.590	0.633	0.715
Total cost of HP steam per month: $X_6 = (X_4 + X_5) * 24 * 30$	US$7574	US$8117	US$9169

9.5.3 *Rate of Exothermic Reaction or Energy Gain*

Figure 9.10 shows the change of the exothermic reaction rate under different settings of A, B and C. Clearly, solution C was able to consistently generate a high exothermic reaction rate.

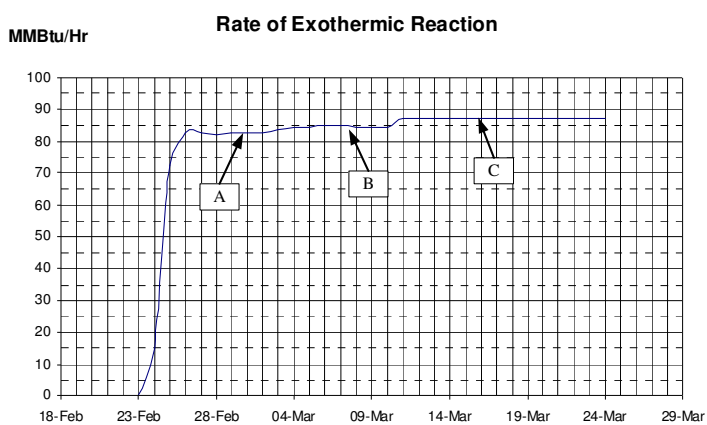


Fig. 9.10 The Exothermic Reaction Rate of Reactor 72CC.

The rate of exothermic reaction determines the amount of energy gain in the cracking process and is calculated as follows (Chevron Corporation, 1989).

$$\text{Energy Gain} = \text{Exothermic Reaction Rate} * \text{Reaction Factor}$$
$$\text{Reaction Factor (RFactor)} = SG * Tx * O_2 * Crc * Ac \quad (9.7)$$

where

SG- specific gravity of the combined feeds
Tx - Temperature of the combined feeds after preheat
O_2 - Percentage of oxygen in the reactor during reaction
Crc - Alumina exothermic reaction coefficient
Ac - Alumina absorption constant

The calculation of the energy gain at the different optimization levels was based on the following fixed parameter values given in Table 9.8, and the corresponding economic gain from the different solutions are listed in Table 9.9. Solution C with the highest exothermic reaction rate contributes the most energy gain in this key performance index. This is

one of the most complex indexes that consumes the highest energy and will give significant cost saving if optimized correctly.

Table 9.8 Parameter values for the calculation of Rfactor.

Parameter	Values
SG	0.9
Tx	0.374
O_2	0.0666
Ac	117.5
Crc	0.0077245
Ac	117.5
RFactor	0.020347

Table 9.9 Comparison of the Energy Gain from Exothermic Reaction Rate.

	A	**B**	**C**
Exothermic Reaction Rate (MMBtu/hr):	82.85	85.00	87.25
Energy Gain (MMBTU/hr)	1.686	1.729	1.775
Energy Gain to TFOE equivalent (1 MMBtu/hr = 4.3052 TFOE)	7.259	7.446	7.643
Total Energy Gain per month: (1 TFOE = US$ 120)	US$627140	US$643319	US$660348

9.5.4 *Summary of the Cost Analysis*

To better illustrate the effect of the optimization process, Table 9.10 summarizes the various key performance indexes considered in terms of dollars. The best solution is point C optimized by MOEA, where the total consumption of fuel and steam is the lowest while the energy gain is the highest. However it has the highest steam consumption.

Table 9.10 Cost Comparison Among Different Settings.

	A by LP	**B by MOEA**	**C by MOEA**
Fuel gas consumption (US$/month)	112955	102305	83769
Steam consumption (US$/month)	7574	8117	9169
Energy Gain (US$/month)	627140	643319	660348

The overall economic gain of MOEA optimization against LP in this particular process is computed as follows:

1. Savings in fuel gas consumption:
   112955 - 83769 = US$29186 per month.
2. Loss in steam consumption:
   9169 - 7574 = - US$1595 per month.
3. Savings in energy gain:
   660348 - 627140 = US$33208 per month

As can be seen, by having the parameters setting from point C of the optimization curve, the obvious tradeoff is the increase of steam consumption for the reaction process. However, the gains in fuel gas and energy can adequately compensate the loss in steam.

9.6 Conclusions

In the feed optimization problem of FCC which comprises of several conflicting objectives, the application of MOEA allows multiple tradeoff solutions to be found in a single run. Physical and market constraints can be introduced to the optimization model easily, allowing us to explore the "what-if question", hence providing a lot of valuable information for the process engineer to work on. Consequently, this provides the process engineer more freedom in the decision making and deeper insights to the optimization problem, and alternatives on how to run the process in a safe, efficient and economical manner. On the other hand, conventional methodologies like LP, based on single objective approach, are not able to provide enough information on the trade-off between the different objectives. The performance comparison between MOEA and LP clearly demonstrates the potential benefit of MOO in real-world problems.

References

Amoco Group (1991). RCC Reactor Feed Optimization Formulation, Standard Operation Manual.

Chevron Corporation (1989). Fired RCC Reactor Energy Efficiency Analysis, Engineering Manual.

Coello Coello, C. A. (1998). An Updated Survey of GA-Based Multi-objective Optimization Techniques, Laboratorio Nacional de Informatica Avanzada (LANIA), Mexico. Rep. No. Lania-RD-98-08.

Corne, D. W., Deb K., Fleming, P. J. and Knowles, J. D. (2003). The Good of the Many Outweighs the Good of the One: Evolutionary Multi-objective Optimization. *The Newsletter of the IEEE Neural Networks Society 1(1),* pp. 9-13.

Dantzig, G. B. and Thapa M. N. (1997). *Linear Programming 1: Introduction*, Springer Series, pp. 115-117.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, New York, pp. 14..

Fonseca, C. M. and Fleming, P. J. (1995). An Overview of Evolutionary Algorithms in Multi-objective Optimization, *Journal of Evolutionary Computation* 3(1), pp. 1-16.

Foulds, L. R. (1981). *Optimization Techniques an Introduction*, Springer-Verlag, New York, pp. 15-16.

Goodsell, P. (2002). 1999 BP Oil Refining Energy Performance Review, Refinery Technology Group.

Goldberg, D. E. and Richardson J. (1987). Genetic Algorithms with Sharing for Multi-Modal Function Optimization, *Proc. 2nd International Conference on Genetic Algorithms*, pp. 41-49.

Horn, J. and Nafpliotis, N. (1993). Multi-objective Optimization Using the Niched Pareto Genetic Algorithm, University of Illinois at Urbana-Champaign, USA, Tech. Rep. 93005.

Sathikannan, R. and Tan, K. C. (2001). Evolutionary Computation for Chemical Engineering Applications, *Conf. Computational Intelligence, Robotics and Autonomous Systems*, Singapore, pp. 253-259.

Srinivas, N. and Deb, K. (1994) Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithms, *J. Evolutionary Computation*, 2(3), pp. 221-248.

Tan, K. C., Lee, T. H. and Khor, E. F. (1999). Evolutionary Algorithms with Goal and Priority Information for Multi-Objective Optimization, *IEEE Conf. Evolutionary Computation*, 1, pp. 106-113.

Tan, K. C., Lee, T. H., Khoo, D. and Khor, E. F. (2001a). A Multi-objective Evolutionary Algorithm Toolbox for Computer-Aided Multi-objective Optimization, *IEEE Transactions on System, Man and Cybernetics Part B*, 31(4): 537-556.

Tan, K. C., Lee, T. H. and Khor, E. F. (2001b). Evolutionary Algorithm with Dynamic Population Size and Local Exploration for Multi-objective Optimization, *IEEE Transactions on Evolutionary Computation*, 5(6), pp. 565-588.

Tan, K. C., Lee, T. H. and Khor, E. F. (2002). Evolutionary Algorithms for Multi-objective Optimization: Performance Assessments and Comparisons. *Artificial Intelligence Review*, 17(4), pp. 251-290.

Tan, K. C., Khor, E. F., Lee, T. H. and Sathikannan, R. (2003). An Evolutionary Algorithm with Advanced Goal and Priority Specification for Multi-objective Optimization. *J. Artificial Intelligence Research,* 18, pp.183-215.

Van Veldhuizen, D. A. and Lamont, G. B. (2000). Multi-objective Evolutionary Algorithms: Analyzing the State-of-the-art, *J. Evolutionary Computation* 8(2), pp. 125-147.

This page intentionally left blank

Chapter 10

Optimal Design of Chemical Processes for Multiple Economic and Environmental Objectives

Elaine Su-Qin Lee, Gade Pandu Rangaiah* and Naveen Agrawal
Department of Chemical & Biomolecular Engineering
National University of Singapore,
Engineering Drive 4, Singapore 117576

*Corresponding Author; e-mail: chegpr@nus.edu.sg

Abstract

For optimal design of chemical processes, selectivity, productivity or simple profit is often used as the sole objective for optimization. Several studies in the past decade have investigated multi-objective optimization (MOO) of chemical processes with selectivity, yield, productivity and/or energy as the objectives; economic criterion such as profit was not an objective in many of these studies. Further, environmental objectives are gaining importance due to the increasing emphasis on environmental protection and sustainability. In this chapter, the optimal design of several processes for multiple economic and environmental objectives is studied. Simultaneous optimization of the selected objectives was carried out using the jumping gene version of the elitist Non-dominated Sorting Genetic Algorithm (NSGA-II-aJG).

For process optimization with respect to several economic criteria such as net present worth, payback period and operating cost, the classical Williams and Otto (WO) process and an industrial low-density polyethylene (LDPE) plant are considered. Results show that either single optimal solution or Pareto-optimal solutions are possible for process design problems depending on the objectives and model equations. Subsequently, industrial ecosystems are studied for optimization with respect to both economic and environmental objectives. Economic objective is important as companies are inherently profit-driven, and there is often a tradeoff between profit and environmental impact. Pareto-optimal fronts were successfully obtained for the 6-plant industrial ecosystem optimized for multiple objectives by NSGA-II-aJG. The study and results reported in this chapter show the need and potential for optimization of processes for multiple economic and environmental objectives.

Keywords: Economic criteria, Environmental criteria, MOO, Williams and Otto Process, Low-density polyethylene plant, Industrial ecosystems.

10.1 Introduction

Multi-objective optimization (MOO) has attracted considerable attention from researchers in chemical engineering, particularly in the past decade. Reported MOO studies have mainly used criteria such as selectivity, yield, productivity and/or energy consumed; see Chapter 2 for the chemical engineering applications studied since 2000 and the objectives used in them. However, profit, an important criterion in any commercial operation, was not used in many of these studies. Apart from the simple profit, several economic criteria such as payback period (PBP), net present worth or value (NPW or NPV) and internal rate of return (IRR) are popular for evaluating projects in industrial practice. Edgar *et al.* (2001) compared the pros and cons of these three profitability criteria. Studies by Buskies (1997) and Pintarič and Kravanja (2006), show that optimal solutions of chemical processes are dependent on the economic objective selected. This indicates the conflicting nature of some, if not all, economic objectives, which means MOO is probably required even if one is interested in only the profitability criteria.

Besides economic objectives, other factors influencing the selection from amongst competing projects include safety, health and

environmental concerns, skills and experience of employees, and the need to diversify. Justifiably, emphasis on environmental impact and sustainability is increasing. Recently, Singh and Lou (2006) looked into the aspect of sustainable development - defined as the development to meet the needs and aspirations of the present generation without compromising the ability of future generations to meet their requirements. This in turn implies reducing consumption of fresh resources and waste generated. Lou *et al.* (2004) and Singh and Lou (2006) presented a model for an industrial ecosystem and solved it using the hierarchical Pareto optimization methodology. The objectives considered are economic and environmental performance indices based on emergies of all feed, product and waste streams.

The above background provides the motivation for the study and applications described in this chapter. Here, two types of process optimization problems are described. The first type has only economic objectives; the two examples considered for this are the classical Williams and Otto (WO) process used recently by Pintarič and Kravanja (2006), and an industrial low-density polyethylene (LDPE) plant based on our recent studies (Agrawal *et al.*, 2006 and 2007). The economic objectives tried are PBP, NPW, IRR, profit before taxes, and/or operating cost. The second type has both economic and environmental indices; for this, the industrial ecosystem with four plants employed by Singh and Lou (2006) is expanded to an ecosystem with six plants and then optimized for multiple objectives.

The technique used for solving the MOO problems in this chapter is the NSGA-II-aJG (Kasat and Gupta, 2003; Agrawal *et al.*, 2006); see Chapter 4 in this book for a description of jumping gene and its inclusion in the Non-dominated Sorting Genetic Algorithm (NSGA) and other algorithms. It is a modification of NSGA-II (Deb *et al.*, 2002) where a section of the chromosome (of pre-determined length) is replaced by a random gene. In the computer programme employed for solving the MOO problems, chromosomes were binary coded and the penalty function approach is used for handling constraints. The NSGA-II-aJG parameter values used for all MOO problems in this chapter are: population size = 200, number of bits for representing a decision variable = 20, length of substring for jumping gene = 20 or 30 for the WO and LDPE example respectively, probability of crossover = 0.8, probability of mutation = 0.01, probability of jumping gene = 0.8 and seed for random number generation = 0.6. These are based on the values used by

Agrawal *et al.* (2006). Number of generations is varied from one problem to another, and is chosen to obtain a converged optimal Pareto with a good spread of points.

10.2 Williams-Otto Process Optimization for Multiple Economic Objectives

Williams-Otto process, representative of numerous chemical processes (Williams and Otto, 1960), has been used by many researchers (e.g., Luus and Jaakola, 1973; Rangaiah, 1985; Biegler *et al.*, 1997; Edgar *et al.*, 2001; and Pintarič and Kravanja, 2006) as an example. In this classical process (Figure 10.1), reactants A and B, together with the recycle stream, are introduced into the continuous-stirred-tank reactor, where the desired product P is produced together with the byproduct E and the waste product G, with C as the intermediate. The reactions occurring in the reactor are:

$$A + B \rightarrow C$$
$$C + B \rightarrow P + E$$
$$P + C \rightarrow G$$

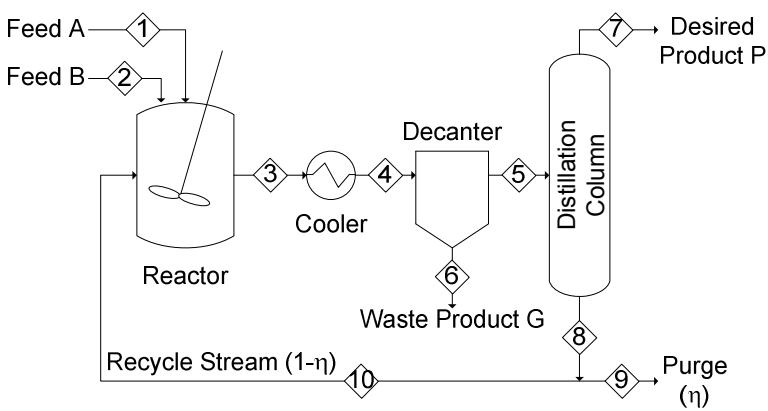


Fig. 10.1 Schematic of the WO Process with stream numbers used in the model.

After cooling the reactor outlet stream in the cooler, component G is separated from the other components in the decanter. In the column, product P is collected in the distillate, while some of the product is

retained in the bottom due to an azeotrope formation. To prevent accumulation of the byproduct (E), a part of the bottom is purged, while most of it is recycled to the reactor. As the purge stream has a substantial fuel value, it is sold to the market.

10.2.1 *Process Model*

The complete model of the WO process, excluding the economic and objective functions, consists of 82 variables and 78 equality constraints (see 'Information and Figures.doc' on the CD). The 82 variables are the rate constants of the 3 reactions (k_i where i = 1, 2, 3), the mass flow rates of reactant j in stream i ($q_{\mathrm{m},i}^{j}$ where i = 1, 2, …, 10, and j = A, B, C, E, G, P), mass flow rates of stream i ($q_{\mathrm{m},i}$ where i = 1, 2, …, 10), reactor temperature (T), reactor volume (V), mass fraction of component j in stream 3 ($w_j = q_{\mathrm{m},3}^{j}/q_{\mathrm{m},3}$ where j = A, B, C, E, G, P) and purge fraction (η). 77 of the equality constraints arise from mass balances on the reactor, cooler, decanter and column. In addition, flow rate of the desired product P (stream 7 in Figure 10.1) is set at 2160 kg/h, providing the last equality constraint. So, there are only 4 (= 82 – 78) degrees of freedom for optimizing the desired objectives; after scrutinizing the equality constraints, decision variables chosen are: V, T, $q_{\mathrm{m},2}^{\mathrm{B}}$ and η. Of the 78 equality constraints, 72 can be used easily to eliminate 72 variables leaving the following 6 nonlinear equations having the mass flow rates of A, B, C, E and G in stream 3 ($q_{\mathrm{m},3}^{\mathrm{A}}$, $q_{\mathrm{m},3}^{\mathrm{B}}$, $q_{\mathrm{m},3}^{\mathrm{C}}$, $q_{\mathrm{m},3}^{\mathrm{E}}$, $q_{\mathrm{m},3}^{\mathrm{G}}$) and of component A in stream 1 ($q_{\mathrm{m},1}^{\mathrm{A}}$) besides the 4 decision variables.

$$q_{\mathrm{m},1}^{\mathrm{A}} + q_{\mathrm{m},3}^{\mathrm{A}}(1-\eta) - \frac{k_1 q_{\mathrm{m},3}^{\mathrm{A}} q_{\mathrm{m},3}^{\mathrm{B}} V\rho}{\left(q_{\mathrm{m},3}\right)^2} - q_{\mathrm{m},3}^{\mathrm{A}} = 0 \qquad (10.1)$$

$$q_{\mathrm{m},2}^{\mathrm{B}} + q_{\mathrm{m},3}^{\mathrm{B}}(1-\eta) - \frac{\left(k_1 q_{\mathrm{m},3}^{\mathrm{A}} + k_2 q_{\mathrm{m},3}^{\mathrm{C}}\right) q_{\mathrm{m},3}^{\mathrm{B}} V\rho}{\left(q_{\mathrm{m},3}\right)^2} - q_{\mathrm{m},3}^{\mathrm{B}} = 0 \qquad (10.2)$$

$$q_{\mathrm{m},3}^{\mathrm{C}}(1-\eta) + \frac{\left[2k_1 q_{\mathrm{m},3}^{\mathrm{A}} q_{\mathrm{m},3}^{\mathrm{B}} - 2k_2 q_{\mathrm{m},3}^{\mathrm{B}} q_{\mathrm{m},3}^{\mathrm{C}} - k_3 q_{\mathrm{m},3}^{\mathrm{C}}\left(2160 + 0.1 q_{\mathrm{m},3}^{\mathrm{E}}\right)\right] V\rho}{\left(q_{\mathrm{m},3}\right)^2} - q_{\mathrm{m},3}^{\mathrm{C}} = 0 \qquad (10.3)$$

$$q_{m,3}^{E}(1-\eta)+\frac{2k_2 q_{m,3}^{B} q_{m,3}^{C} V\rho}{\left(q_{m,3}\right)^2}-q_{m,3}^{E}=0 \tag{10.4}$$

$$0.1q_{m,3}^{E}(1-\eta)+\frac{\left[k_2 q_{m,3}^{B} q_{m,3}^{C}-0.5k_3 q_{m,3}^{C}\left(2160+0.1q_{m,3}^{E}\right)\right]V\rho}{\left(q_{m,3}\right)^2}-$$

$$\left(2160+0.1q_{m,3}^{E}\right)=0 \tag{10.5}$$

$$\frac{\left[1.5k_3 q_{m,3}^{C}\left(2160+0.1q_{m,3}^{E}\right)\right]V\rho}{\left(q_{m,3}\right)^2}-q_{m,3}^{G}=0 \tag{10.6}$$

where $k_1=\left(5.9755\times10^{9}\right)\exp\left(\frac{-12000}{1.8T}\right)$,

$k_2=\left(2.5962\times10^{12}\right)\exp\left(\frac{-15000}{1.8T}\right)$,

$k_3=\left(9.6283\times10^{15}\right)\exp\left(\frac{-20000}{1.8T}\right)$, $\rho=801$ kg/m^3 and

$q_{m,3}=q_{m,3}^{A}+q_{m,3}^{B}+q_{m,3}^{C}+1.1q_{m,3}^{E}+q_{m,3}^{G}+2160$.

The 78 equality constraints in the complete model were thus reduced to 6 nonlinear equations as the genetic algorithm, NSGA-II-aJG is not effective in handling multiple equality constraints. Its inadequateness was also observed even when the equations had been reduced to 6 equations. Hence, the Broyden's update and finite-difference Jacobian function (DNEQBF) of the IMSL Library was embedded in the objective evaluation to solve the nonlinear equations 10.1 to 10.6.

The above model equations are validated by reproducing the results in Pintarič and Kravanja (2006) for single objective optimization, using NSGA-II-aJG. For this, the 4 decision variables are: *V, T,* $q_{m,2}^{B}$ and η. For each set of values of these decision variables, the above 6 nonlinear equations are solved using the DNEQBF program of the IMSL software, to find $q_{m,3}^{A}$, $q_{m,3}^{B}$, $q_{m,3}^{C}$, $q_{m,3}^{E}$, $q_{m,3}^{G}$ and $q_{m,1}^{A}$. The remaining variables and objective functions can be calculated using the values of the 4 decision variables and the 6 variables found by solving the nonlinear equations 10.1 to 10.6. As can be seen in Table 10.1, the optimal results obtained

using NSGA-II-aJG are similar to those of Pintarič and Kravanja (2006), with minimum, mean and maximum difference of 0.002%, 0.2% and 1.1% respectively. The single objective solutions are also validated using the Solver tool in Excel(R); see 'Excel Solver.xls' on the attached CD. In this model, there are 10 variables: V, T, $q^{B}_{m,2}$, η, $q^{A}_{m,3}$, $q^{B}_{m,3}$, $q^{C}_{m,3}$, $q^{E}_{m,3}$, $q^{G}_{m,3}$ and $q^{A}_{m,1}$, and equations 10.1 to 10.6 are the constraints that the Solver has to satisfy while the selected economic objective was optimized. If the initial guesses of the 10 variables are random, Solver experienced difficulty in converging to the optimal solution; thus, it is essential that the initial guesses are close to the optimal values for Solver to produce the correct optimal solution. When the solutions of the Solver are compared with the results given by Pintarič and Kravanja (2006), the mean and maximum difference is 0.094% and 0.36% (in IRR for max. PBT) respectively. Note that the programme used by Pintarič and Kravanja (2006) is GAMS/CONOPT.

Table 10.1 Optimal values for single objective optimization obtained using NSGA-II-aJG in the present study and reported values by Pintarič and Kravanja (2006) in brackets.

	Profitability measures used as the objective function			
	Min. PBP or Max. IRR	**Max. Profit Before Tax (PBT)**	**Min. Total Annual Cost (TAC)**	**Max. NPW**
V (m^3)	0.871 (0.873)	6.80 (6.82)	7.89 (7.90)	3.71 (3.75)
T (K)	375 (374)	343 (342)	342 (342)	351 (351)
η	0.100 (0.100)	0.113 (0.113)	0.102 (0.102)	0.110 (0.109)
$q^{A}_{m,1}$ (kg/h)	6,121 (6,123)	4,956 (4,957)	4,809 (4,808)	5,247 (5,239)
$q^{B}_{m,2}$ (kg/h)	13,965 (13,956)	11,118 (11,113)	10,882 (10,880)	11,809 (11,792)
FCI (Million US$)	0.924 (0.925)	7.21 (7.22)	8.37 (8.37)	3.94 (3.97)
CF (Million US$/ yr)	0.875 (0.876)	2.42 (2.42)	2.52 (2.52)	1.99 (2.00)
NPW (Million US$)	4.02 (4.02)	6.46 (6.44)	5.87 (5.86)	7.31 (7.30)
IRR (yr^{-1})	0.944 (0.945)	0.312 (0.313)	0.272 (0.274)	0.489 (0.493)

10.2.2 *Objectives for Optimization*

Objectives used for optimizing the WO process, as used by Pintarič and Kravanja (2006), are presented here. Fixed capital investment (FCI) is the capital necessary for the installed process equipment with all components needed for complete process operation. For the WO process, FCI is dependent on the volume of the reactor (V) and the density of the material flow (ρ):

$$\text{FCI} = \frac{600V\rho}{0.453} \ \$ \tag{10.7}$$

Operating cost (c_{op}) is the cost incurred when the plant is in operation, excluding the depreciation cost. The total annual cost (TAC) is the sum of c_{op} and the depreciation cost, which thus includes contribution from FCI. Depreciation cost is calculated by dividing FCI over the lifetime of the project assumed to be 10 years. Profit before taxes (PBT) is the difference between the annual revenue and TAC, without accounting for taxes. Annual cash flow (CF) is the sum of profit after taxes and depreciation where the tax rate is taken to be 30% per year. For the WO process, TAC, PBT and CF (all in \$) are given by

$$\text{TAC} = \frac{1}{0.453}\left(168q_{m,1} + 252q_{m,2} + 2.22\left(q_{m,10} + q_{m,1} + q_{m,2}\right) + 84q_{m,6} + 60V\rho\right) + 1041.6 \tag{10.8}$$

$$\text{PBT} = \frac{1}{0.453}\left(2207q_{m,7} + 50q_{m,9}\right) - \text{TAC} \tag{10.9}$$

$$\text{CF} = 0.7\left[\frac{1}{0.453}\left(2207q_{m,7} + 50q_{m,9} - 168q_{m,1} - 252q_{m,2} - 2.22\left(q_{m,10} + q_{m,1} + q_{m,2}\right) - 84q_{m,6}\right) - 1041.6\right] + 0.3\left[\frac{60V\rho}{0.453}\right] \tag{10.10}$$

Note that depreciation and CF are assumed to be the same every year during the project life of 10 years. Hence, PBP, the time required after start-up to recover FCI through the annual cash flows, is simply FCI/CF.

NPW is the present value of all investments and cash flows during the project lifetime. This takes into account time value of money at the expected rate of return (i year^{-1}). The expected rate of return, usually termed as the discount rate, is taken as 0.12 year^{-1}. NPW for the WO process is:

$$\text{NPW} = -\text{TCI} + f_{PA}(\text{i})\ \text{CF} \tag{10.11}$$

where TCI is the total capital investment and $f_{PA}(i)$ is the present worth annuity factor to bring future CFs to the present time, which is given by $[(1+i)^{10}-1]/[i(1+i)^{10}]$ for 10 years of identical cash flows. Internal rate of return (IRR) is the particular i at which NPW is zero. Hence, by re-arranging equation 10.11, IRR can be found by solving

$$f_{PA}(\text{IRR}) = \frac{(1+\text{IRR})^{10}-1}{\text{IRR}(1+\text{IRR})^{10}} = \frac{\text{TCI}}{\text{CF}} \tag{10.12}$$

Assuming no working capital (i.e., TCI = FCI), the right hand side of the above equation is the same as PBP, and hence IRR is related to PBP as shown in Figure 10.2. Owing to the monotonic relationship in Figure 10.2, minimizing PBP is equivalent to maximizing IRR. This conclusion and the above equations are for the WO process assuming constant CF and no working capital. For the LDPE plant (discussed later), TCI which is the sum of FCI and working capital, is 110% of FCI. As TCI is proportional to FCI, the above conclusion is valid. It is easier to minimize PBP than to maximize IRR as calculation of the latter requires iterations.

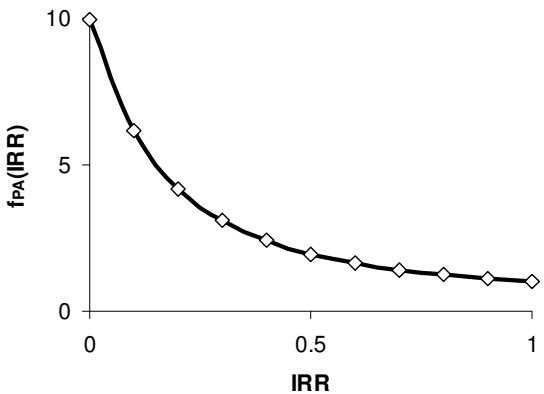


Fig. 10.2 Variation of $f_{PA}(\text{IRR})$ (= PBP) with IRR.

10.2.3 *Multi-Objective Optimization*

After validating the model and single objective optimization results, the WO process is optimized for multiple objectives using NSGA-II-aJG. Guided by the conflicting results of single objective optimization in Table 10.1, two combinations of the objectives are used for MOO; these two problems including bounds on decision variables are summarized in

Table 10.2. For many of the results reported in this chapter, a personal computer with Pentium(R) M (CPU 796 MHz, 1.25GB RAM) is used. The million floating point operations per second (MFlops) of this system for a 500×500 matrix using the LINPACK benchmark program (available at http://www.netlib.org, accessed on June 3, 2007), is 123 (mean of 10 trials). From Figures 10.3a, 10.3b, 10.4a and 10.4b, 1000 generations of NSGA-II-aJG is preferable for MOO of the WO process as it gave a smoother Pareto over a larger range of values, especially when compared to the Pareto after 500 generations. Hence, for the WO process, 1000 generations are used for the two cases. The CPU time required for optimizing the WO process on the above personal computer, for 1000 generations is about 8 minutes.

Table 10.2 Objectives, constraints and decision variables in the MOO of the WO Process.

Case	Objectives	Decision Variables	Constraints
A	Max. NPW Max. PBT	$0.85 \le V \le 10\ \mathrm{m}^3$ $322 \le T \le 378\ \mathrm{K}$ $0 \le \eta \le 0.99$ $10{,}000 \le q_{\mathrm{m},2}^{\mathrm{B}} \le 15{,}000\ \mathrm{kg/h}$	Equations 10.1 to 10.6
B	Max. NPW Min. PBP		

10.2.3.1 *Case A - Maximize NPW and PBT*

The Pareto-optimal front in Figure 10.3a confirms that NPW and PBT are conflicting; NPW increases rapidly and then slowly from 6.48 to 7.31 million US$ as the PBT decreases from 2.43 to 2.29 million US$/year. Three decision variables (T, V and $q_{\mathrm{m},2}^{\mathrm{B}}$) contribute to the optimal Pareto (Figures 10.3c, 10.3d and 10.3e) whereas η is practically constant at 0.11. These results can be explained as follows. From Figure 10.3c, it is evident that NPW increases as the reactor temperature (T) increases. As a result, the reactor volume (V) decreases as shorter residence time is required for reactions to take place, and so FCI decreases as illustrated in Figures 10.3d and 10.3g. The decrease in residence time is also due to the increase in $q_{\mathrm{m},1}^{\mathrm{A}}$ and $q_{\mathrm{m},2}^{\mathrm{B}}$ (Figures 10.3f and 10.3e). Since the production rate of P is fixed, revenue from it is constant but the revenue from the sale of purge could vary. As $q_{\mathrm{m},1}^{\mathrm{A}}$ and $q_{\mathrm{m},2}^{\mathrm{B}}$ are increased, by mass balance, the purge flow rate increases. Thus, although revenue is

increasing as T and $q^{B}_{m,2}$ increase, operating cost increases at a faster rate due to higher flow rates of the recycle, waste stream G and feed flows. This means TAC increases and CF decreases (Figure 10.3h) contributing to the decrease in PBT. As FCI decreases at a faster rate than CF multiplied by $f_{PA}(0.12)$, NPW increases (Figures 10.3a, 10.3g and 10.3h). Consequently, PBP decreases as NPW increases (Figure 10.3i). Hence, the set of optimal solutions is different when PBP is minimized while maximizing NPW in Case B (as illustrated in the next section).

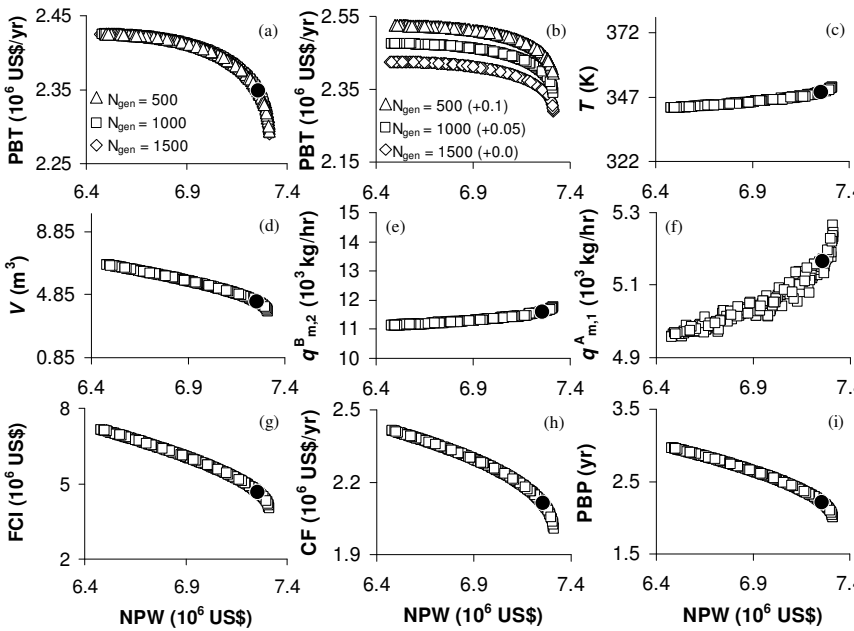


Fig. 10.3 Optimal results of the WO process for multiple objectives – Case A; for clarity, results in (a) are re-plotted in (b) with suitable vertical shifts in the ordinate; NPW is shown on the x-axis in all plots.

Pareto optimal solutions provide many equally good solutions to the decision maker. The methods discussed in Chapter 7 can be used for ranking the Pareto optimal solutions. In general, the relative importance of the two objectives should be considered for selecting one of the Pareto optimal solutions, for implementation. When evaluating the feasibility of a design project, NPW is more important than PBT because the former takes into account the cash flows over the project life and the time value

of money. Hence, an optimal solution with a larger NPW would be preferable. From Figure 10.3a, at high values of NPW, the opportunity cost of PBT is huge for a small increase in NPW. Thus, NPW could be traded off slightly for a larger improvement in PBT and this point is indicated by a black circle in Figure 10.3 (with NPW = 7.26 million US$ and PBT = 2.35 million US$/yr). Quantitatively, NPW worsens by 0.7% from its maximum value while PBT improves by 3.6% from its minimum value. Corresponding to this particular optimal solution, values of the decision variables (T, V, $q^{B}_{m,2}$ and η) are 348.9 K, 4.41 m^3, 11.6×10^3 kg/hr and 0.11 respectively.

10.2.3.2 *Case B - Maximize NPW and Minimize PBP*

The Pareto-optimal front in Figure 10.4a shows that NPW and PBP can vary significantly by a factor 2 for the WO process. NPW increases from 4.03 to 7.31 million US$ as the PBP increases from 1.06 to 1.96 years. Compared to Figure 10.3i, the PBP values obtained in Case B is better (i.e. lower) than that in Case A with a different set of optimal values for T, V and $q^{B}_{m,2}$ (Figures 10.4c, 10.4d and 10.4e) except that the optimal η is practically constant at 0.11 as before. Further, the variation of NPW with the decision variables (Figure 10.4) is opposite to that mentioned in Case A. To increase NPW in Case B, T decreases and so the reactions proceed at a slower rate. High residence time is then required and it is effected by the increase in V and the decrease in $q^{A}_{m,1}$ and $q^{B}_{m,2}$ (Figures 10.4d, 10.4f and 10.4e respectively). As V increases, FCI increases simultaneously (Figure 10.4g). Since the feed flow rates decrease, the revenue decreases as the purge to be sold as fuel decreases (recall that the product flow rate is constant at 2160 kg/h). Although the revenue is decreasing, the operating cost decreases at a faster rate from the reduction in flow rates of the recycle, waste stream G and feed flows. Hence, TAC decreases and CF increases (Figure 10.4h). The increase in CF multiplied by $f_{PA}(0.12)$ is much larger than the increase in FCI, and hence NPW increases. However, the increase in CF is smaller than the increase in FCI, thus PBP increases. As shown in Figure 10.4i, values of PBT for this set of optimal solutions are inferior (i.e. lower) to those in Case A. Hence, the set of optimal solutions in Case B do not produce a Pareto for the simultaneous maximization of PBT and NPW (Case A).

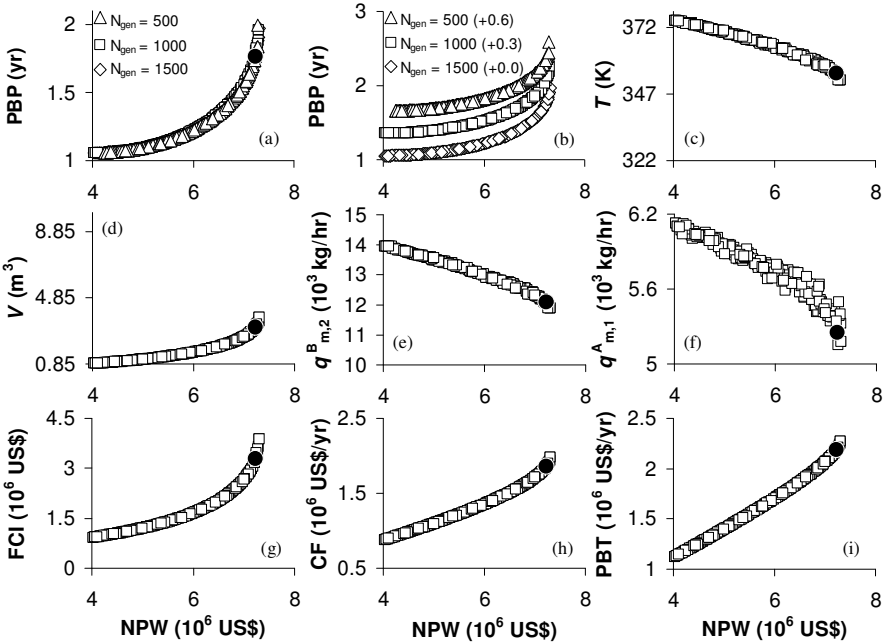


Fig. 10.4 Optimal Results of WO Process for Multiple Objectives – Case B; for clarity, results in (a) are re-plotted in (b) with suitable vertical shifts in the ordinate; NPW is shown on the x-axis in all plots.

As mentioned in Case A, the relative importance of the objectives should be known for selecting one of the Pareto optimal solutions. Of the two objectives, NPW is still of higher importance as it accounts for the time value of money and the expected lifespan of the project. On the other hand, PBP does not account for these and it is less than 2 years for this project (Figure 10.4a), which is generally acceptable. Hence, more importance will be given to NPW in selecting one of the Pareto optimal solutions. From Figure 10.4a, it can be seen that, at high values of NPW, the tradeoff for PBP is high for a small improvement in NPW. Hence, NPW could be sacrificed marginally for a larger improvement in PBP, and a suggested optimal design is indicated by a black circle in Figure 10.4 (with NPW = 7.23 million US$ and PBP = 1.76 yr). Quantitatively, NPW worsens by 1.1% from its maximum value while PBP improves by 10.2% from its minimum value. Corresponding to this particular optimal

solution, values of the decision variables (T, V, $q_{m,2}^{B}$ and η) are 354.7 K, 3.09 m^3, 12.1×10^3 kg/hr and 0.11 respectively.

10.3 LDPE Plant Optimization for Multiple Economic Objectives

Low-density polyethylene (LDPE), a commodity polymer used in a wide variety of applications such as plastic bags and wrappings, cable insulation and coatings, can be produced by either tubular or autoclave processes (Read *et al.*, 1997). It is unique in its polymerization process; free-radical initiated polymerization is used to make it compared to transition-metal catalysis for high density polyethylene and linear LDPE. The free-radical process leads to the unique molecular structure of LDPE: large amounts of long-chain branching (Maraschin, 2006). A simplified process flow diagram for an LDPE plant is shown in Figure 10.5.

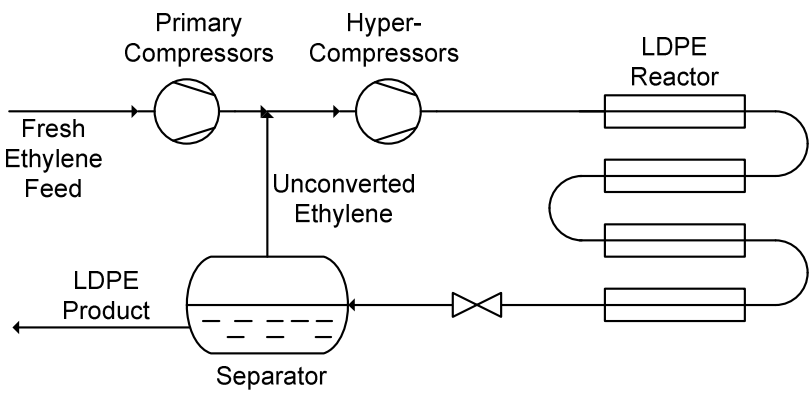


Fig. 10.5 Simplified schematic of an LDPE plant.

10.3.1 *Process Model and Objectives*

Recently, Agrawal *et al.* (2006) had modeled the LDPE reactor as an ideal plug flow reactor and presented all the model equations and parameters for use by researchers. The model equations include ordinary differential equations for overall and component mass balances, energy balance and momentum balance. The reactor model of Agrawal *et al.* (2006) is adopted, and cost expressions and economic objectives are

formulated for the LDPE plant optimization. Owing to the extremely high pressure (about 2000 atm) in the LDPE reactor, cost data for major units in the plant: reactor and compressors are not available in the open literature. Confidential industrial data available to us for an LDPE plant gave the breakdown of the investment and operating cost. From these data, it was observed that compressors contributed a large proportion to the investment cost. Further, investment cost is often correlated with the plant capacity. Hence, FCI with two terms – one involving compression power (CP) and another involving the plant capacity, is developed. Using the Chemical Engineering Plant Cost Index (CEPCI) and a discount factor of 10% for technological advancements, FCI of the past was brought to CEPCI of 500. The FCI formulated thus is:

$$\text{FCI} = 5.41\left(F_{\text{M}} X_{\text{M,f}}\right)^{0.6} + 0.589(\text{CP})^{0.6} \text{ Million US\$} \tag{10.13}$$

$$\text{where} \quad \text{CP} = F_{\text{M}}\left(649.79 X_{\text{M,f}} + 233.94 P_{\text{in}}^{0.23} - 987.21\right) \tag{10.14}$$

Here, F_{M} is the flow rate of the monomer through the reactor, $X_{\text{M,f}}$ is the monomer conversion and P_{in} is the pressure at the reactor inlet. In equation 10.13, the first term on the right hand side takes care of plant capacity while the second term is for compressors involving CP.

Cost data such as the price of feed (ethylene), solvent (n-butane), initiators, LDPE and electricity are provided in Table 10.3. The remaining cost equations involve the multiplication of the price data and the respective flow rate, and then bringing the values to yearly basis. These are also included in Table 10.3. Assuming working capital is 10% of FCI, TCI is

$$\text{TCI} = 1.1\text{FCI} = 5.95\left(F_{\text{M}} X_{\text{M,f}}\right)^{0.6} + 0.648(\text{CP})^{0.6} \text{ Million US\$}$$

The operating cost (Million US\$/yr) is the sum of all the costs (except revenue) in Table 10.3.

$$c_{\text{op}} = F_{\text{M}}\left(15.5 X_{\text{M,f}} + 0.138 P_{\text{in}}^{0.23} - 0.583\right) + 8.51 F_{\text{S}} + 851\left(F_{\text{I,1}} + F_{\text{I,2}}\right) + 0.590\left(F_{\text{M}} X_{\text{M,f}}\right)^{0.6} + 0.0642(\text{CP})^{0.6}$$

Using the straight-line method to calculate depreciation over a period of 10 years, PBT (Million US\$/yr), $\text{PBT} = \text{Rev} - c_{\text{op}} - D$ or

$$\text{PBT} = F_{\text{M}}\left(15.7 X_{\text{M,f}} - 0.138 P_{\text{in}}^{0.23} + 0.583\right) - 8.51 F_{\text{S}} - 851\left(F_{\text{I,1}} + F_{\text{I,2}}\right) - 1.13\left(F_{\text{M}} X_{\text{M,f}}\right)^{0.6} - 0.123(\text{CP})^{0.6}$$

Cash Flow (Million US\$/yr), $\text{CF} = (1 - r_{\text{t}})(\text{Rev} - c_{\text{op}}) + r_{\text{t}} D$. Assuming the tax rate, r_{t} is 0.3 yr^{-1},

$$\mathrm{CF} = F_{\mathrm{M}}\left(11.0X_{\mathrm{M,f}} - 0.0966P_{\mathrm{in}}^{0.23} + 0.408\right) - 5.96F_{\mathrm{S}} - 596\left(F_{\mathrm{I,1}} + F_{\mathrm{I,2}}\right) - 0.251\left(F_{\mathrm{M}}X_{\mathrm{M,f}}\right)^{0.6} - 0.0272(\mathrm{CP})^{0.6}$$

Payback Period (yr), PBP = FCI/CF

Assuming expected rate of return, i of 0.12 and project life of 10 years, NPW (Million US$) = - TCI + f_{PA}(i) CF where f_{PA}(i) =$[(1+0.12)^{10} - 1]/[0.12(1+0.12)^{10}]$ = 5.65. Hence,

$$\mathrm{NPW} = F_{\mathrm{M}}\left(62.3X_{\mathrm{M,f}} - 0.546P_{\mathrm{in}}^{0.23} + 2.31\right) - 33.7F_{\mathrm{S}} - 3370\left(F_{\mathrm{I,1}} + F_{\mathrm{I,2}}\right) - 7.37\left(F_{\mathrm{M}}X_{\mathrm{M,f}}\right)^{0.6} - 0.802(\mathrm{CP})^{0.6}$$

In the above equations, flow rates of solvent (*n*-butane), initiator I_1 (*tert*-butyl peroxypivalate) and initiator I_2 (*tert*-butyl 3,5,5 trimethyl-peroxyhexaonate) are represented by F_{S}, $F_{\mathrm{I,1}}$ and $F_{\mathrm{I,2}}$ respectively.

Table 10.3 Cost data for the LDPE plant.

Description	**Unit Price (Reference)**	**Revenue/Cost (Million US$/year) assuming a stream factor of 0.9 (i.e. 328.5 days of production per year)**
Product Price:		
LDPE	1.1 US$/kg (Maraschin, 2006)	$\mathrm{Rev} = 31.2F_{\mathrm{M}}X_{\mathrm{M,f}}$
Cost of Raw Materials:		
Ethylene	0.45 US$/kg (Fishhaut, 2003)	$c_{\mathrm{Eth}} = 12.8F_{\mathrm{M}}X_{\mathrm{M,f}}$
Solvent, n-butane	0.3 US$/kg (LPG World, 2004)	$c_{\mathrm{S}} = 8.51F_{\mathrm{S}}$
Initiators	30 US$/kg (Maraschin, 2006)	$c_{\mathrm{Init}} = 851\left(F_{\mathrm{I,1}} + F_{\mathrm{I,2}}\right)$
Utility Cost		
Electricity	0.075 US$/kWh*	$c_{\mathrm{Elec}} = 0.000591\mathrm{CP}$
Other Utilities	1% of Cost of Ethylene*	$c_{\mathrm{Util}} = 0.01c_{\mathrm{Eth}} = 0.128F_{\mathrm{M}}X_{\mathrm{M,f}}$
Labor Cost	7.5% of Cost of Ethylene*	$c_{\mathrm{Lab}} = 0.075c_{\mathrm{Eth}} = 0.96F_{\mathrm{M}}X_{\mathrm{M,f}}$
Maintenance Cost	10.9% of FCI*	$c_{\mathrm{Main}} = 0.109\mathrm{FCI} = 0.590\left(F_{\mathrm{M}}X_{\mathrm{M,f}}\right)^{0.6} + 0.0642(\mathrm{CP})^{0.6}$

*These data and percentages are based on the industrial data.

10.3.2 *Multi-Objective Optimization*

Table 10.4 gives the brief details of the MOO problems studied for the LDPE plant. Objectives in Cases A and B are the same as those used in the MOO of the WO process. Optimization of the LDPE plant for these objectives gave single optimal solution for both cases (Table 10.5). The objectives have almost the same value in both Cases A and B with only 0.9% difference. Despite this, flow rates of solvent, first and second initiators exhibit 14% to 38% difference. These results imply that it would take a large change in the flow rates of solvent and initiators to have any noticeable effect on the economic functions.

Table 10.4 Objectives, constraints and decision variables in the MOO – LDPE plant.

Case	Objectives	Decision Variables	Constraints
A	Max. NPW Max. PBT	$5\times10^{-5} \le F_o \le 10\times10^{-5}$ kg/s $2\times10^{-2} \le F_S \le 0.5$ kg/s $5\times10^{-5} \le F_{I,1} \le 5\times10^{-3}$ kg/s $5\times10^{-5} \le F_{I,2} \le 5\times10^{-3}$ kg/s $413.15 \le T_{J,m} \le 543.15$ K, $m = 1, 4, 5$ $473.15 \le T_{J,n} \le 543.15$ K, $n = 2, 3$ $182.39 \le P_{in} \le 248.25$ MPa $50 \le L_{z1} \le 70$ m $80 \le L_{z2} \le 120$ m $140 \le L_{z3} \le 220$ m $400 \le L_{z4} \le 600$ m $430 \le L_{z5} \le 650$ m $0.04 \le D_{int} \le 0.06$ m $0.1778 \le D_{Jacket} \le 0.2286$ m $0.5\times10^{-3} \le V_{J,m} \le 25\times10^{-3}$ m^3/s, $m = 2, 3, 4$ $0.1\times10^{-3} \le V_{J,5} \le 25\times10^{-3}$ m^3/s	Model equations in Tables 1 and 2 of Agrawal *et al.* (2006)
B	Max. NPW Min. PBP		
C	Max. PBT Min. c_{op}		

Table 10.5 Optimization of Cases A and B for the LDPE plant using NSGA-II-aJG.

	Case A: Max. NPW and PBT	**Case B: Max. NPW and Min. PBP**
$X_{M,f}$	0.458	0.456
P_{in} (atm)	1800	1800
F_S (kg/s)	1.05E-01	1.20E-01
$F_{I,1}$ (kg/s)	1.57E-03	9.70E-04
$F_{I,2}$ (kg/s)	1.29E-03	9.78E-04
PBT (M US$)	52.2	52.5
NPW (M US$)	136	137
PBP (yr)	2.66	2.64

The reason for single optimal solution for the LDPE plant in Cases A and B compared to the Pareto-optimal solutions for the WO process is the difference in the FCI expression, which, for the LDPE plant, is based on the production capacity and CP rather than equipment size (e.g., volume of reactor in the WO process). For the WO process, it is observed earlier that while the volume of the reactor increases, flow rate through it decreases (Figures 10.3d, 10.3e and 10.3f); this would increase FCI while c_{op} and revenue (due to lower purge flow) decrease. However, in the case of the LDPE plant, decrease in the production would decrease the revenue, c_{op} and FCI simultaneously. Hence, Pareto optimality is not observed for Cases A and B in the LDPE plant. Pareto-optimal fronts are obtained for Case C, and these results are presented and discussed below. PBT and c_{op} are chosen as Case C for this example as it is obvious that these two objectives are conflicting. Since industries are inherently profit-driven, PBT is the natural choice as an objective function. In addition, industries are also trying to increase their profit margin by reducing their c_{op}. Thus, with the Pareto optimal solutions obtained in Case C, the engineer/manager would be aware of the PBT traded off while minimizing c_{op}.

10.3.2.1 *Case C - Maximize PBT and Minimize c_{op}*

From Figures 10.6a and 10.6b, it is noted that 2000 generations were sufficient to obtain the widest spread of optimal solutions (denoted by squares). Only for the LDPE plant, the computer used is Intel$^{(R)}$ Xeon$^{(TM)}$ (CPU 3.60 GHz, 3.25GB RAM). The MFlops of this system for a 500×500 matrix using the LINPACK benchmark program (available at http://www.netlib.org, accessed on June 3, 2007), is 319 (mean of 10 trials). CPU time required for the MOO of the LDPE plant is about 17 hours. Results of maximizing PBT and minimizing c_{op} (Figure 10.6) show that PBT increases from 3.3 to 51.8 million US$/year as c_{op} increases from 36.3 to 92.5 million US$/year. The relationship between these two objectives is almost linear except for slight non-linearity when c_{op} is at about 92 million US$/year. For the linear portion, P_{in} remains constant as its increase would increase c_{op} (via electric power required and its expense) and reduce PBT which is not desirable (Figure 10.6d). On the other hand, $X_{M,f}$ increases as it would increase the PBT due to the increased sales but c_{op} also increases due to the increase in the raw materials required.

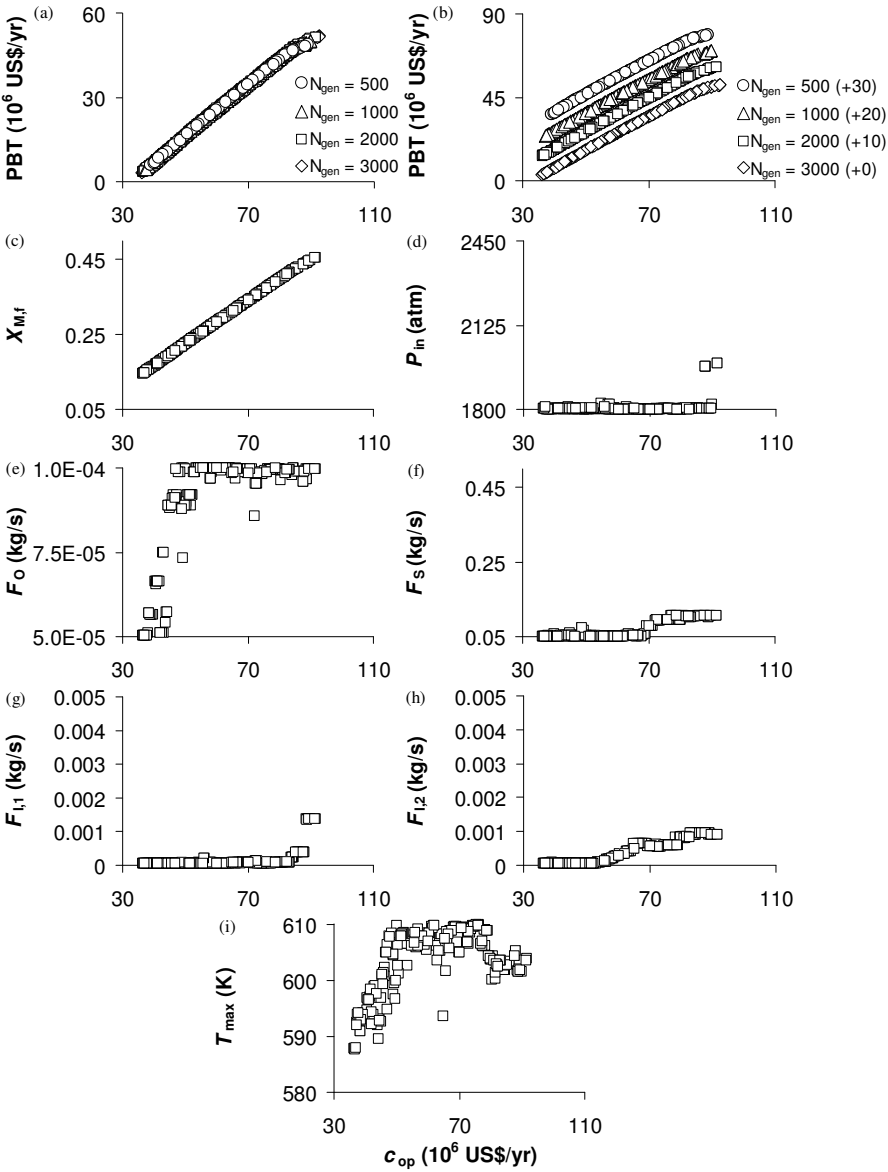


Fig. 10.6 Optimization of LDPE plant by NSGA-II-aJG – Case C; for clarity, results in (a) are re-plotted in (b) with suitable vertical shifts in the ordinate; c_{op} is shown on the x-axis in all plots.

There were mainly 5 decision variables that affect $X_{M,f}$ – namely, flow rate of oxygen (F_O), solvent (F_S), initiators I_1 and I_2 ($F_{I,1}$ and $F_{I,2}$) and inlet pressure to the reactor (P_{in}). The initial increase of $X_{M,f}$ is determined by F_O, which promotes free-radical reactions, thus increasing $X_{M,f}$ (Figure 10.6e). Thereafter, when F_O has reached its upper bound of 0.0001 kg/s, the increase in $X_{M,f}$ is due to the increase in $F_{I,2}$ and later $F_{I,1}$, which promote the initiation reaction and consequently other reactions (Figures 10.6g and 10.6h). The variation of F_S after c_{op} = 67 million US$/yr is due to the increase in $F_{I,2}$ and thereafter the increase in $F_{I,1}$, such that the molecular weight of the polymer produced is maintained. Optimal values of the remaining 16 decision variables are scattered and not shown in Figure 10.6 (but are available in 'Information and Figures.doc' on the CD); values of 12 of these variables are between their lower and upper bounds. The remaining 4 decision variables are the coolant/heating fluid flow rate though each of the 4 jackets; their optimal values are scattered in the upper two-thirds of their respective range, with none in the lower part of the range.

The nonlinear portion at the high PBT and c_{op} is contributed by the jump in values of P_{in} and $F_{I,1}$ (Figures 10.6d and 10.6g). The reason for the jump in the values is to maintain the increase in $X_{M,f}$ and thus revenue further. However, the increase in P_{in} and $F_{I,1}$ would lead to an increase in initiator and electricity costs as well as the increase in FCI through CP. Hence, PBT did not increase much. Generally, as $X_{M,f}$ increases, more ethylene is converted; with more reactions occurring, maximum temperature in the reactor (T_{max}) increases as the reactions are exothermic (Figure 10.6i).

10.4 Optimizing an Industrial Ecosystem for Economic and Environmental Objectives

In this section, we describe optimization of an industrial ecosystem (IE) having 6 plants for multiple economic and environmental objectives. The motivation and model of this IE follows that of Singh and Lou (2006) for an IE with 4 plants. The industrial symbiosis of Kalundborg, Denmark (Jacobsen, 2006), is the classic example of an IE. It began in 1961 when there was limited supply of ground water and the oil refinery, Statoil, initiated a project to utilize the surface water from Lake Tissø. Thereafter, the number of partners gradually increased as about 20 more collaborative projects were introduced. Today, there are a total of 8

partners which have built a network of material and energy exchanges amongst themselves. Their main products and services are heat and power production, motor fuels, ammonium thiosulphate, plasterboard products, remediation of contaminated soil, pharmaceutical products, enzymes and waste water treatment service.

The example of Kalundborg has illustrated many pointers worthy of study. Firstly, it has reduced its environmental impact significantly - reduction in the utilization of resources, decrease in the amount of waste emissions and valorization of wastes. Secondly, the collaborations had conserved money for the companies involved. Thirdly, as material and energy exchanges between partners evolved over time in which there were separate contracts between the partners, self-organization took place where projects were only carried out when they were financially viable. Lastly, the proximity of the companies had provided lower cost of infrastructure for material exchange. The other IE examples are Mississippi River Corridor Industrial Complex (Hertwig *et al.* 2002) and Dalian Economic and Technological Development Zone, China (http://www.uneptie.org/pc/ind-estates/casestudies/Dalian.htm, accessed on March 10, 2007). There is potential for developing more IEs in other locations such as Jurong Island of Singapore, where many petroleum refining and petrochemical industries are already operating in close proximity. In order to have an effective environmental protection, companies and societies have to look into sustainability, defined as the development to meet the needs of the present generation without compromising the ability of future generations to meet their own needs. There are three aspects to a sustainable development, namely, economic wealth, environmental cleanliness and societal welfare. All three should be achieved simultaneously.

Emergy analysis is one method of environmental accounting to assess the sustainability of IEs. Emergy of a material or service is defined as the total direct and/or indirect inputs of energy required to produce it. The inputs are in terms of the same type of energy, for which solar energy, which is the source of all the energy in this world, is usually used. Hence, transformity factors are required and used to bring all substances to a common unit through emergy. When these factors are unavailable, ratio of emergy to money (called emdollars) is used to calculate the emergy of the substances or services. An emdollar is the ratio of total emergy used to the gross domestic product of a country. This simplified approach implies emergy of a material or service is

proportional to its cost, which is generally not valid. However, it allows replacing emergy of a material with dollars in some objectives involving ratios.

10.4.1 *Model of an IE with Six Plants*

Figure 10.7 shows a typical IE with 6 plants; the main reasons for expanding the 4-plant IE of Singh and Lou (2006) are to increase the complexity of the optimization problem and to evaluate the capability of NSGA-II-aJG for a larger problem with 30 decision variables. Plant 1 would be used for describing the IE. The inputs to plant 1 are: total amount of non-renewable feed (N_1), treated waste from plant 1 recycled internally (W_{TRN1}) at a specific ratio to the fresh renewable feed (R_1), and untreated waste from plant 1 recycled internally (W_{URR1}) and from plant 6 (Y_{UW6}). A fraction of these inputs contributes to the product (Y_{P1}) sold to the market and the rest leaves as the waste. A portion of this waste is sent for treatment, exiting as treated waste (W_{T1}). The rest of the waste is the untreated waste (W_{U1}). W_{T1} is split into 3 streams – one to be sold to the market (Y_{TW1}), one to be recycled internally as a non-renewable stream (W_{TRN1}) and one to be disposed off (W_{TW1}). Similarly, W_{U1} is divided into 3 streams – one to be recycled internally as a renewable stream (W_{URR1}), one to be sold to another plant (Y_{UW1}) and one to be disposed off (W_{UW1}). The rest of the plants in the IE are modeled in a similar fashion.

The model equations for the IE are given below by considering Plant i. These are for calculating the product and waste streams from relevant mass balances.

$$M_{Y_{Pi}} = \alpha X_i \tag{10.15}$$

$$M_{W_{Ti}} = (1-\alpha) Z_{12} X_i \tag{10.16}$$

$$M_{W_{URRi}} = (1-\alpha) Z_{11} Z_{21} X_i \tag{10.17}$$

$$M_{Y_{UWi}} = (1-\alpha) Z_{11} Z_{22} X_i \tag{10.18}$$

$$M_{W_{UWi}} = (1-\alpha) Z_{11} Z_{23} X_i \tag{10.19}$$

$$M_{W_{TRNi}} = (1-\alpha) Z_{12} Z_{31} X_i \tag{10.20}$$

$$M_{Y_{TWi}} = (1-\alpha) Z_{12} Z_{32} X_i \tag{10.21}$$

$$M_{W_{TWi}} = (1-\alpha) Z_{12} Z_{33} X_i \tag{10.22}$$

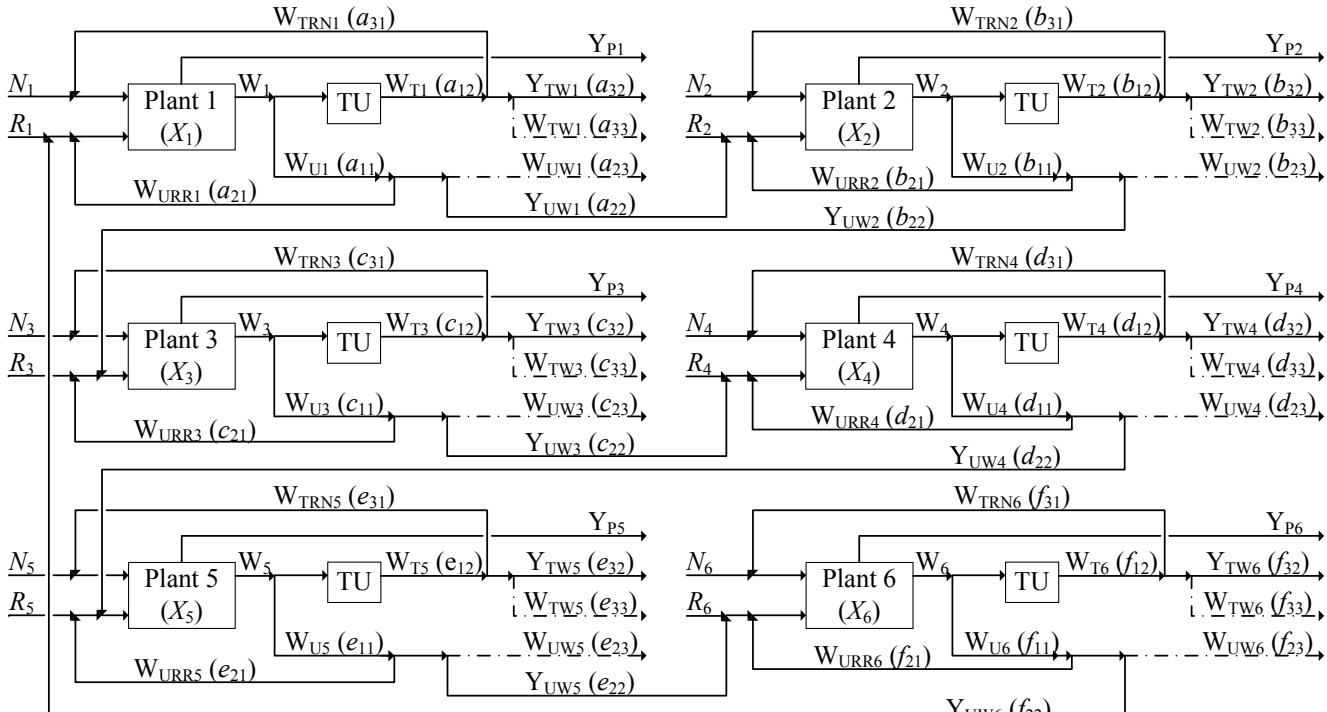


Fig. 10.7 Schematic diagram of an industrial ecosystem with 6 plants;
TU: treatment unit for waste streams; · – · –: waste streams for disposal

The renewable and non-renewable inputs to Plant *i* are:

$$R_i = \left(\frac{X_i}{1+\beta}\right) - M_{Y_{UWj}} - M_{W_{URRi}} \tag{10.23}$$

$$N_i = \beta\left(R_i + M_{Y_{UWj}} + M_{W_{URRi}}\right) - M_{W_{TRNi}} \tag{10.24}$$

Constraints for Plant *i* are:

$$Z_{21} + Z_{22} + Z_{23} = 1 \tag{10.25}$$

$$Z_{31} + Z_{32} + Z_{33} = 1 \tag{10.26}$$

$$0 \le Z_{21}, Z_{22}, Z_{23}, Z_{31}, Z_{32}, Z_{33} \le 1 \tag{10.27}$$

In the above equations, M is the mass flow rate of the stream indicated by its subscript, α is the ratio of the total plant capacity (X_i) that is sold as product while (1-α) is the ratio of X_i that is regarded as waste stream. Z_{11} is the proportion of the waste stream that is untreated while Z_{12} is the proportion that is treated. Z_{21}, Z_{22} and Z_{23} are the ratios in which the untreated waste is split to be recycled internally, to be sold to a neighboring plant and to be disposed off respectively. In a similar way, Z_{31}, Z_{32} and Z_{33} are the ratios in which the treated waste is split to be recycled internally, to be sold to the market and to be disposed off respectively. The renewable and non-renewable feed streams are represented by R_i and N_i respectively. In equation 10.24, β is the ratio of the amount of the nonrenewable feed (sum of N_i and $M_{W_{TRNi}}$) to the amount of renewable feed (sum of R_i, $M_{Y_{UWj}}$ and $M_{W_{URRi}}$), that is fed to the plant. Table 10.6 provides the symbols and values used in the model equations 10.15 to 10.27 and the objectives are given in the following section.

Table 10.6 Symbols used in the IE model equations and their values.

Quantity	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
α	0.75	0.8	0.85	0.7	0.65	0.9
β	0.7	0.75	0.65	0.72	0.78	0.67
δ	1100	1320	1210	1230	1375	1430
j	6	1	2	3	4	5
Z*	a	b	c	d	e	f
Z_{11}*	0.4	0.45	0.6	0.7	0.8	0.75
Z_{12}*	0.6	0.55	0.4	0.3	0.2	0.25

*For example, $a_{11} = 0.4$ and $a_{12} = 0.6$

10.4.2 *Objectives, Results and Discussion*

The index of economic performance (IEcP) is defined as the ratio of the total yield to the total investment and cost.

$$\text{IEcP} = \frac{\text{Y}_\text{P} + \text{Y}_\text{TW} + \text{Y}_\text{UW}}{\delta + X^{0.7} + \text{W}_\text{T}^{0.6} + \text{W}_\text{TW} + \text{W}_\text{UW} + \text{W}_\text{URR}^{0.5} + \text{W}_\text{TRN}^{0.5} + R + N + \text{Y}_\text{UW}} \tag{10.28}$$

The numerator is the sum of emergy of the product (Y_P) and the treated waste (Y_TW) sold to the market and the untreated waste sold to the neighboring plant (Y_UW). The denominator is the sum of the investment, waste disposal cost, operation cost and cost of feed streams, all in terms of emergy. Investment includes a base value (δ_i) and a term which varies with the capacity of the unit (Q_i) raised to a fraction (m) – that is, $\delta_i + Q_i^{\text{m}}$. There are two units which require investment: the production unit with a capacity of X and the waste treatment unit with a capacity of W_T. In equation 10.28, δ is the sum of δ_i for both the production unit and the waste treatment unit. Waste disposal cost includes the cost of disposing treated and untreated waste (W_TW and W_UW respectively). Operation cost is the cost of recycling both the treated and untreated waste internally to the plant ($\text{W}_\text{TRN}^{0.5}$ and $\text{W}_\text{URR}^{0.5}$ respectively). Lastly, cost of feed streams includes the cost of renewable feed, non-renewable feed and untreated waste from the neighboring plant (R, N and Y_UW respectively).

The index of the environmental performance (IEvP) gives a quantitative measure of the impact a process has on the environment.

$$\text{IEvP} = \frac{N + \text{W}_\text{TW} + \text{W}_\text{UW}}{R + \text{W}_\text{TRN}^{0.5} + \text{W}_\text{URR}^{0.5} + \text{Y}_\text{UW}} \tag{10.29}$$

The numerator is the sum of emergy of the non-renewable resources used (N) and the wastes (both treated and untreated) that is disposed off (W_TW and W_UW). The denominator is the sum of emergy of the renewable feed used (R), the treated and untreated waste recycled within the plant ($\text{W}_\text{TRN}^{0.5}$ and $\text{W}_\text{URR}^{0.5}$ respectively) and the untreated waste obtained from a neighboring plant (Y_UW).

Both IEcP and IEvP are ratio of emergies of relevant streams in the plant. If the emergy of a stream is calculated using the emdollars (i.e., emergy to money ratio) based on the total emergy used and gross domestic product of a country, then it can be replaced by the money value of a stream in IEcP and IEvP since the same emdollars occurs in both the numerator and denominator. This simplified approach, as in

Singh and Lou (2006), is followed in this study as well. In cases where the transformity factors of all material/service streams in a plant are available, they can be used for a more realistic MOO. In the following, the terms in the numerator and denominator in IEcP and IEvP are in terms of their money value (i.e., dollars) instead of emergy.

Profitability is the difference of the revenue earned and the cost incurred. Revenue is obtained from the sale of product to the market (Y_{Pi}), treated waste to the market (Y_{TWi}) and untreated waste to the neighboring plant (Y_{UWi}). Cost is incurred from the investment of the process and waste treatment unit as well as the cost of operation (i.e. cost of feed streams, cost of disposal of treated and untreated waste, cost of recycling treated and untreated waste). Profit, IEcP and IEvP for the IE with 6 plants are as follows:

$$\begin{aligned}\text{Profit}_i = &\sum_{i=1}^{6}\left(c_{\text{Y}_{\text{P}i}} M_{\text{Y}_{\text{P}i}} + c_{\text{Y}_{\text{TW}i}} M_{\text{Y}_{\text{TW}i}} + c_{\text{Y}_{\text{UW}i}} M_{\text{Y}_{\text{UW}i}}\right) - \\ &\sum_{i=1}^{6}\left(\delta + X_i^{\,0.7} + M_{\text{W}_{\text{T}i}}{}^{0.6} + c_{\text{FTD}i} M_{\text{W}_{\text{TW}i}} + c_{\text{FUD}i} M_{\text{W}_{\text{UW}i}} + \right. \\ &\left. c_{\text{W}_{\text{URR}i}} M_{\text{W}_{\text{URR}i}}{}^{0.5} + c_{\text{W}_{\text{TRN}i}} M_{\text{W}_{\text{TRN}i}}{}^{0.5} + c_{\text{R}i} R_i + c_{\text{N}i} N_i + c_{\text{Y}_{\text{UW}j}} M_{\text{Y}_{\text{UW}j}}\right)\end{aligned} \tag{10.30}$$

$$\begin{aligned}\text{IEcP}_i = &\sum_{i=1}^{6}\left(c_{\text{Y}_{\text{P}i}} M_{\text{Y}_{\text{P}i}} + c_{\text{Y}_{\text{TW}i}} M_{\text{Y}_{\text{TW}i}} + c_{\text{Y}_{\text{UW}i}} M_{\text{Y}_{\text{UW}i}}\right) \div \\ &\sum_{i=1}^{6}\left(\delta + X_i^{\,0.7} + M_{\text{W}_{\text{T}i}}{}^{0.6} + c_{\text{FTD}i} M_{\text{W}_{\text{TW}i}} + c_{\text{FUD}i} M_{\text{W}_{\text{UW}i}} + \right. \\ &\left. c_{\text{W}_{\text{URR}i}} M_{\text{W}_{\text{URR}i}}{}^{0.5} + c_{\text{W}_{\text{TRN}i}} M_{\text{W}_{\text{TRN}i}}{}^{0.5} + c_{\text{R}i} R_i + c_{\text{N}i} N_i + c_{\text{Y}_{\text{UW}j}} M_{\text{Y}_{\text{UW}j}}\right)\end{aligned} \tag{10.31}$$

$$\text{IEvP} = \frac{\sum_{i=1}^{6} c_{\text{N}i} N_i + c_{\text{FTD}i} M_{\text{W}_{\text{TW}i}} + c_{\text{FUD}i} M_{\text{W}_{\text{UW}i}}}{\sum_{i=1}^{6} c_{\text{R}i} R_i + c_{\text{W}_{\text{URR}i}} M_{\text{W}_{\text{URR}i}}{}^{0.5} + c_{\text{W}_{\text{TRN}i}} M_{\text{W}_{\text{TRN}i}}{}^{0.5} + c_{\text{Y}_{\text{UW}j}} M_{\text{Y}_{\text{UW}j}}} \tag{10.32}$$

Note that the ith term in each summation is for Plant i. Cost data in the above equations and bounds on the capacities for the 6 plants are summarized in Tables 10.7 and 10.9 respectively. These quantities for the first 4 plants are the same as those in Singh and Lou (2006); cost data

and bounds chosen for Plants 5 and 6 are based on these values with a ± 5 difference in the cost and a ± 50 difference in the bounds on the capacities. Further, the non-renewable feed has a higher cost than the renewable feed for Plants 1 to 4, and so the cost data for Plants 5 and 6 are chosen in a similar way. The transfer of the untreated waste from one plant to another is modified to accommodate the additional 2 plants.

Table 10.7 Cost data for IE containing 6 plants.

Cost/Price ($/t)	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
Cost of Renewable Feed, $c_{\mathrm{R}i}$	35	30	40	45	50	40
Cost of Non-renewable Feed, $c_{\mathrm{N}i}$	70	80	90	95	85	65
Price of Product, $c_{\mathrm{Y_{P}}i}$	105	150	130	170	135	160
Price of Untreated Waste, $c_{\mathrm{Y_{UW}}i}$	20	30	20	25	15	30
Price of Treated Waste, $c_{\mathrm{Y_{TW}}i}$	55	50	45	55	45	50
Disposable Cost of Untreated Waste, $c_{\mathrm{FUD}i}$	80	75	60	80	75	70
Disposable Cost of Treated Waste, $c_{\mathrm{FTD}i}$	65	60	50	65	70	55
Recycle Cost of Untreated Waste, $c_{\mathrm{W_{URR}}i}$	10	12	17	15	13	18
Recycle Cost of Treated Waste, $c_{\mathrm{W_{TRN}}i}$	25	25	22	26	24	25

Possible objectives for the IE with 6 plants are the profit, IEcP and IEvP. Comparing the two economic objectives, profit is the difference between the revenues and the costs of the system while IEcP is the ratio of the emergy of revenues to the emergy of costs. As mentioned above, the emdollars in both numerator and denominator of IEcP is the same, thus IEcP is simply reduced to the ratio of revenues to costs. In other words, profit gives the absolute monetary value that the system can earn

over costs while IEcP gives the amount of revenue that can be earned with every dollar invested as cost. Thus, it could be expected that maximizing either profit or IEcP would give different results. This is illustrated in the single objective optimization of each objective (Table 10.8). The optimal solutions for single objective were first obtained using NSGA-II-aJG and then used as the initial guess for the Solver. The resulting optimal solutions from the Solver are presented in Table 10.8; the objective for each case is further improved by about 1% compared to the optimal solutions obtained using NSGA-II-aJG.

Table 10.8 Single-objective optimization of IE with 6 plants for profit, IEcP and IEvP separately.

	Max Profit	**Max IEcP**	**Min IEvP**		**Max Profit**	**Max IEcP**	**Min IEvP**
Objective Functions							
Profit	0.282	0.187	0.165				
IEcP	2.330	2.510	2.380				
IEvP	1.299	1.349	1.180				
Plant 1				**Plant 2**			
X_1	500	100	500	X_2	700	700	200
a_{21}	0	0.5	0	b_{21}	0	0.5	0.01
a_{22}	1	0.5	1	b_{22}	1	0.5	0.99
a_{31}	1	1	1	b_{31}	1	1	1
a_{32}	0	0	0	b_{32}	0	0	0
Plant 3				**Plant 4**			
X_3	800	400	400	X_4	750	300	300
c_{21}	0	0.5	0.01	d_{21}	0	0.5	0
c_{22}	1	0.5	0.99	d_{22}	1	0.5	1
c_{31}	1	1	1	d_{31}	1	1	1
c_{32}	0	0	0	d_{32}	0	0	0
Plant 5				**Plant 6**			
X_5	650	150	150	X_6	850	850	850
e_{21}	0	0.5	0	f_{21}	0	0.5	0
e_{22}	1	0.5	1	f_{22}	1	0.5	1
e_{31}	1	1	1	f_{31}	1	1	1
e_{32}	0	0	0	f_{32}	0	0	0

For the maximization of profit, it is evident that the capacities of all plants hit the maximum. This is because, at the highest capacities, although total cost increases, total revenue increases to a greater extent; thus, maximum capacities translate to maximum profit. In addition, the ratios of a_{21}, a_{22}, e_{21} and e_{22} were found to take on multiple solutions for

the same value of Profit (Tarafder *et al*., 2007). The ranges a_{21} and e_{21} can take are from 0 to 0.5; consequentially, a_{22} and e_{22} will take on values corresponding to $(1 - a_{21})$ and $(1 - e_{21})$ respectively. On the other hand, maximization of IEcP led to the maximization of 2 plant capacities (X_2 and X_6) and the minimization of the remaining 4 plant capacities (i.e. X_1, X_3, X_4 and X_5). This is because the price of the product ($c_{\mathrm{Y}_{\mathrm{P}i}}$) for plants 2 and 6 are amongst the highest coupled with the lowest renewable feed cost ($c_{\mathrm{R}i}$) and the lowest non-renewable feed cost ($c_{\mathrm{N}i}$); see Table 10.7. Although plant 4 has the highest $c_{\mathrm{Y}_{\mathrm{P}i}}$ of 170 \$/t, it has amongst the highest feed costs as well: 45 \$/t for $c_{\mathrm{R}i}$ and 95 \$/t for $c_{\mathrm{N}i}$. Finally, it is observed that for low IEvP values, it would be better if the ratios of $c_{\mathrm{N}i}$ to $c_{\mathrm{R}i}$ be low (i.e. Plants 1, 5 and 6). Also, β should be low so that the renewable feed for the respective plant will be high, improving the IEvP (i.e. Plants 1, 3 and 6). Since plants 1 and 6 have both characteristics as mentioned above, their plant capacities (X_1 and X_6) reached the maximum while the remaining capacities were kept at their lower bounds, for minimizing IEvP.

Table 10.9 Objectives, decision variables and constraints for MOO of the IE with 6 plants.

Case	Objectives	Decision Variables	Constraints
A	Max. IEcP Min. IEvP	100 t/yr ≤ X_1 ≤ 500 t/yr 200 t/yr ≤ X_2 ≤ 700 t/yr 400 t/yr ≤ X_3 ≤ 800 t/yr 300 t/yr ≤ X_4 ≤ 750 t/yr 150 t/yr ≤ X_5 ≤ 650 t/yr 250 t/yr ≤ X_6 ≤ 850 t/yr $0 \le Z_{21} \le 0.5$ for $Z=a,b,\ldots,f$ $0 \le Z_{22} \le 1.0$ for $Z=a,b,\ldots,f$ $0 \le Z_{31} \le 1.0$ for $Z=a,b,\ldots,f$ $0 \le Z_{32} \le 0.5$ for $Z=a,b,\ldots,f$	$Z_{22} \ge Z_{21}$ for $Z=a,b,\ldots,f$ $Z_{31} \ge Z_{32}$ for $Z=a,b,\ldots,f$ Model equations (10.15) to (10.27)
B	Max. Profit Min. IEvP		

Since it is of interest to have both economic and environmental criteria in optimization, two MOO problems are considered as summarized in Table 10.9. This table includes decision variables and their bounds as well as constraints. The inequality constraints exist for the split ratios (decision variables) such that more treated waste is recycled back to the plant rather than sold to the market and more

untreated waste is sold to the neighboring plant than being recycled internally.

10.4.2.1 *Case A - Maximizing IEcP and Minimizing IEvP*

The results of optimizing the IE with 6 plants for maximizing IEcP and minimizing IEvP are shown in Figure 10.8; Figures 10.8a and 10.8b indicate that around 5000 generations are required for the convergence to the optimal Pareto. CPU time on the personal Computer stated above, for 5000 generations is nearly 10 minutes. Optimal IEvP increases from about 1.19 to 1.29 as the optimal IEcP increases from 2.35 to 2.47; the relationship between the optimal IEvP and IEcP is piece-wise linear with a change in slope at IEcP of ~ 2.43. Many decision variables corresponding to the optimal Pareto in Figure 10.8a are at their lower or upper bounds or constant; values of these variables are given in Table 10.10, and the remaining variables (X_1, X_2, e_{21} and e_{22}) are shown in Figures 10.8c to 10.8f. Capacity of Plant 1 (X_1) is at its upper bound and starts decreasing when IEcP is about 2.43 where X_2 reaches its upper bound (Figures 10.8c and 10.8d). The optimal split ratios are generally constant (Table 10.10) and satisfy the constraints in Table 10.9. This shows that the penalty function method could handle these constraints well. There are, however, two split ratios both for Plant 5 (e_{21} and e_{22}) whose optimal values were not constant as shown in Figures 10.8e and 10.8f. The two distinct, constant values which each of these variables have assumed are quite close to each other. This indicates multiple solutions giving the same values for objectives (Tarafder *et al.*, 2007).

Table 10.10 Optimal capacity and fractions for splitting treated and untreated waste in each of the six plants: maximize IEcP and minimize IEvP.

Plant 1		**Plant 2**		**Plant 3**		**Plant 4**		**Plant 5**		**Plant 6**	
X_1	*	X_2	*	X_3	400	X_4	300	X_5	150	X_6	850
a_{21}	0.02	b_{21}	0.01	c_{21}	0.02	d_{21}	0.04	e_{21}	*	f_{21}	0.06
a_{22}	0.98	b_{22}	0.99	c_{22}	0.98	d_{22}	0.96	e_{22}	*	f_{22}	0.94
a_{31}	1	b_{31}	0.97	c_{31}	0.99	d_{31}	1	e_{31}	0.98	f_{31}	0.75
a_{32}	0	b_{32}	0.03	c_{32}	0.01	d_{32}	0	e_{32}	0.01	f_{32}	0.25

*See Figure 10.8 for these variables.

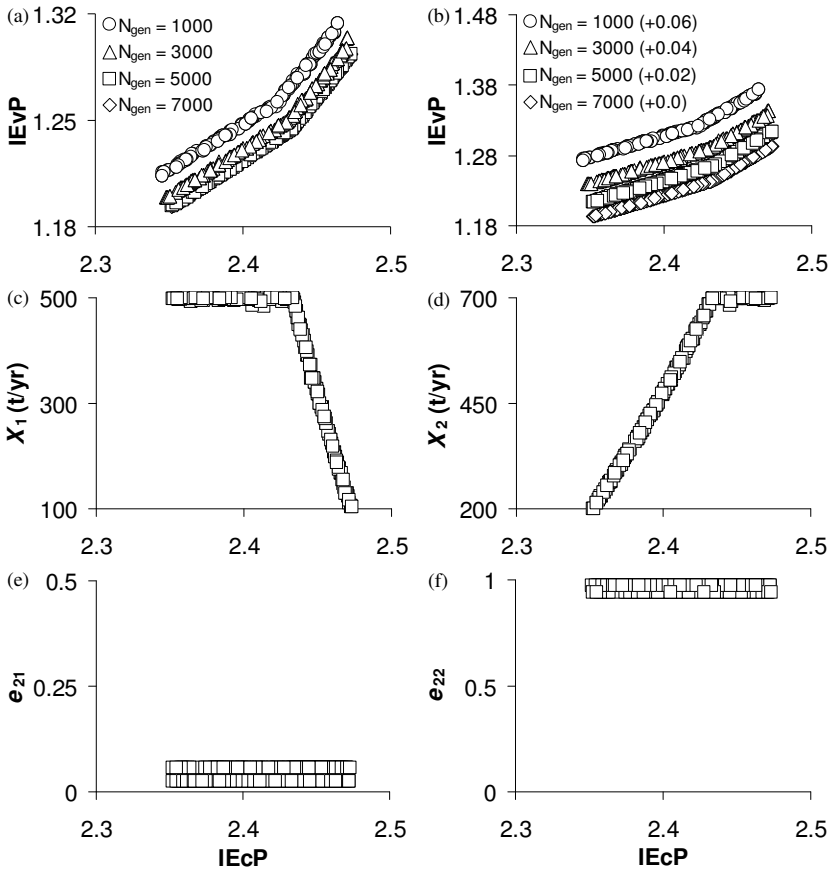


Fig. 10.8 MOO of the IE with 6 plants for maximizing IEcP and minimizing IEvP; for clarity, results in (a) are re-plotted in (b) with suitable vertical shifts; IEcP is shown on the x-axis in all plots.

10.4.2.2 *Case B - Maximizing Profit and Minimizing IEvP*

Companies are inherently profit-driven, and hence profit rather than IEcP is chosen as the objective function in this case. Almost 5000 generations are required for optimizing the IE with 6 plants for maximizing profit and minimizing IEvP (Figures 10.9a and 10.9b). Optimal IEvP increases from about 1.20 to 1.31, almost linearly, as the optimal profit increases from 0.16×10^6 to 0.28×10^6 \$/year. For these objectives, optimal capacities of Plants 1 and 6 are at their upper bound (500 and 850

tons/year respectively) whereas that of Plants 3, 2, 4 and 5 is initially at its lower limit and then begin to increase, one by one in sequence, to reach the respective upper bound (Figures 10.9c to 10.9f). Thus, profit is maximized when the plants reach their maximum capacities. The tradeoff is that IEvP worsens, which is expected since there is more throughput. Optimal split ratios are practically constant (Table 10.11) except 2 split ratios (Figures 10.9g and 10.9h) that displayed some variation, probably due to multiple solutions. Recall that, for maximizing IEcP, the optimal capacities of four out of six plants were at their lower bounds. Hence, with regards to the economic objective, the choice of IEcP or profit would lead to different optimal solutions.

Table 10.11 Optimal fractions for splitting treated and untreated waste in each of the six plants: maximize profits and minimize IEvP.

Plant 1		**Plant 2**		**Plant 3**		**Plant 4**		**Plant 5**		**Plant 6**	
X_1	500	X_2	*	X_3	*	X_4	*	X_5	*	X_6	850
a_{21}	0.25	b_{21}	0.01*	c_{21}	0.12	d_{21}	0	e_{21}	0.01	f_{21}	0.02
a_{22}	0.75	b_{22}	0.99*	c_{22}	0.87	d_{22}	1	e_{22}	0.99	f_{22}	0.98
a_{31}	1	b_{31}	1	c_{31}	0.98	d_{31}	0.97	e_{31}	0.97	f_{31}	0.87
a_{32}	0	b_{32}	0	c_{32}	0.02	d_{32}	0.03	e_{32}	0.03	f_{32}	0.12

*See Figure 10.9 for some variation in these variables.

10.4.2.3 *Comparison of Cases A and B*

To have a quantitative comparison between both Cases A and B, specific optimal solutions were chosen from Case A and Case B that have the same IEvP value, which is chosen to be the average of the maximum and minimum IEvP values obtained in both cases (= 1.253). The two solutions with IEvP value of 1.253 for both cases are summarized in Table 10.12. It is expected that Case A has a higher IEcP and lower Profit than Case B due to the objective functions in each case. As a result, values of decision variables are different. As discussed above for single objective optimization, maximization of profit is achieved by the maximization of the capacities of each plant. So, capacities of all plants are higher in Case B than in Case A except for the capacity of plant 2 (X_2). Note that maximum profit is not achieved as yet and X_2 will eventually increase to its maximum limit when higher profits are achieved at the expense of higher IEvP.

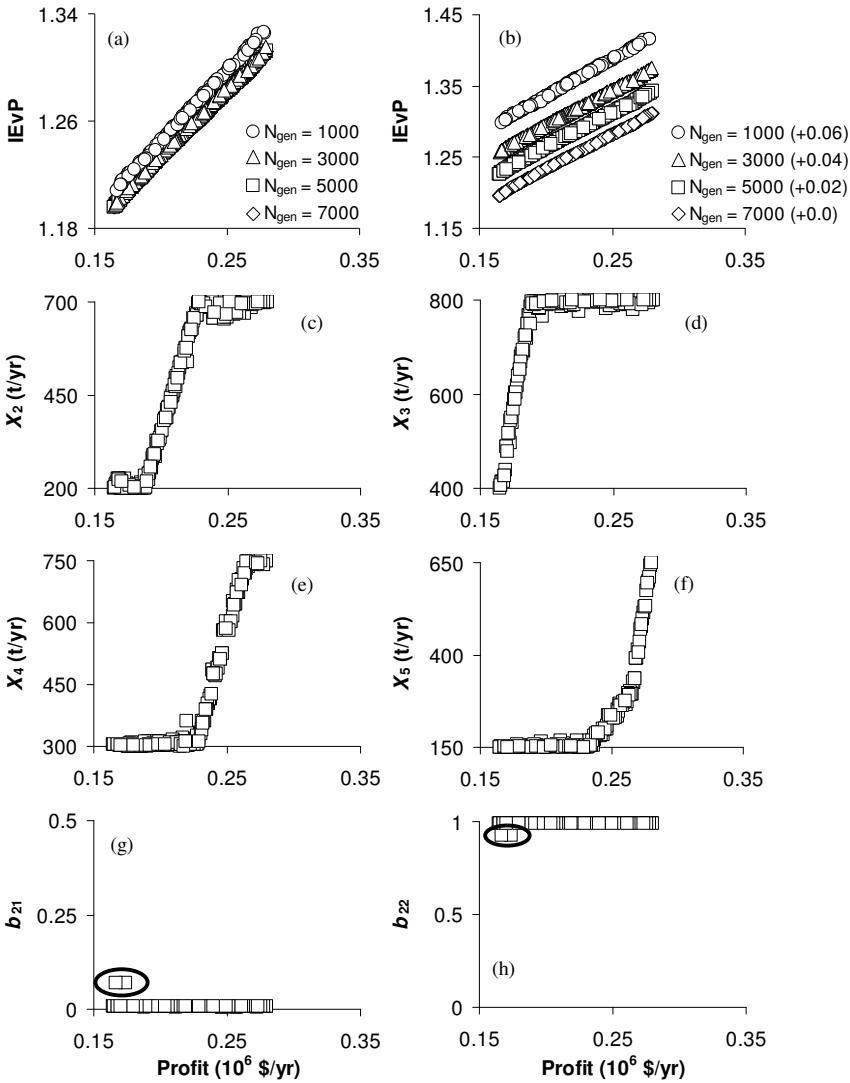


Fig. 10.9 MOO of the IE with 6 plants for maximizing profit and minimizing IEvP; for clarity, results in (a) are re-plotted in (b) with suitable vertical shifts; Profit is shown on the x-axis in all plots.

Table 10.12 Comparison of two optimal solutions in cases A and B with the same IEvP.

	Case A	Case B		Case A	Case B		Case A	Case B
Plant 1			**Plant 2**			**Plant 3**		
X_1	429	500	X_2	699	575	X_3	400	800
a_{21}	0.02	0.25	b_{21}	0.01	0.01	c_{21}	0.02	0.12
a_{22}	0.98	0.75	b_{22}	0.99	0.99	c_{22}	0.98	0.87
a_{31}	1	1	b_{31}	0.97	1	c_{31}	0.99	0.98
a_{32}	0	0	b_{32}	0.03	0	c_{32}	0.01	0.01
Plant 4			**Plant 5**			**Plant 6**		
X_4	301	312	X_5	150	152	X_6	850	850
d_{21}	0.04	0	e_{21}	0.06	0.01	f_{21}	0.06	0.02
d_{22}	0.96	1	e_{22}	0.94	0.99	f_{22}	0.94	0.98
d_{31}	1	0.97	e_{31}	0.98	0.97	f_{31}	0.75	0.87
d_{32}	0	0.03	e_{32}	0.02	0.03	f_{32}	0.25	0.12
Objective Functions								
Profit	0.201	0.219	IEcP	2.440	2.383	IEvP	1.253	1.253

10.5 Conclusions

In this chapter, MOO of three process design problems is described. The first two applications – classical WO process and LDPE plant are optimized for two economic objectives such as NPW, PBT, PBP and/or c_{op}. For maximizing NPW and PBT, and maximizing NPW and minimizing PBP, Pareto-optimal solutions are obtained in the case of the WO process while single optimum is obtained in the case of the LDPE plant. The reason for this is the difference in the cost expressions. Pareto-optimal solutions are obtained when the LDPE plant is optimized for PBT and c_{op}. Hence, Pareto-optimal solutions are possible for economic objectives alone depending on the problem, objectives and cost equations.

Next, an IE with six plants and a concise model for it are presented. Optimization of this IE for IEcP, IEvP and/or Profit is described. As expected, Pareto-optimal solutions are obtained since economic and environmental criteria are often conflicting. Obviously, these solutions depend on the objectives considered. Although conflict among the objectives of interest may be expected *a priori*, solving the MOO problems is useful as the resulting Pareto-optimal solutions provide not only the trend of objectives but also that of decision variables. In general, Pareto-optimal solutions provide greater insight into the optimal

solutions of the process design problem. Hence, an informed selection from among the many optimal solutions can then be made for the process design under investigation.

The design problems described in this chapter are successfully solved using NSGA-II-aJG along with penalty function for constraints. However, some difficulties were faced in solving the application problems with many equality constraints. In the WO process, these were reduced to a small number, which are then solved using a nonlinear equation solver for each set of decision variables selected by the optimization method in its search. Further research is required on better handling of equality constraints in NSGA-II and similar methods, for process design problems.

Nomenclature

The symbols used in this chapter have been defined at their first occurrence in the text. They are also compiled in the word document 'Information and Figures.doc' on the CD.

References

Agrawal N., Rangaiah G. P., Ray A. K. and Gupta S. K. (2006). Multi-Objective Optimization of the Operation of an Industrial Low-Density Polyethylene Tubular Reactor Using Genetic Algorithm and its Jumping Gene Adaptations, *Industrial and Engineering Chemistry Research*, 45, pp. 3182-3199.

Agrawal, N., Rangaiah, G. P., Ray, A. K. and Gupta, S. K. (2007). Design Stage Optimization of an Industrial Low Density Polyethylene Tubular Reactor for Multiple Objectives using NSGA-II and its Jumping Gene Adaptations, *Chemical Engineering Science*, 62, pp. 2346-2365.

Biegler, L. T., Grossman, I. E. and Westerberg, A. W. (1997). *Systematic Methods of Chemical Process Design*, Prentice Hall, New Jersey.

Buskies, U. (1997). Economic Process Optimization Strategies, *Chemical Engineering and Techonology*, 20, pp. 63-70.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6, pp. 182-197.

Edgar, T. F., Himmelblau, D. M. and Lasdon, L. S. (2001). *Optimization of Chemical Processes*, McGraw-Hill, New York.

Fishhuat, E. (2003). Petrochemical firms take a stand against oil price volatility, *Market Focus*, pp. 54-55.

Hertwig, T. A., Xu, A., Nagy, A. B., Pike, R. W., Hopper, J. R. and Yaws, C. L. (2002). A prototype system for economic, environmental and sustainable optimization of a chemical complex, *Clean Technologies and Environmental Policy*, 3, pp. 363-370.

Jacobsen, N. B. (2006). Industrial Symbiosis in Kalundborg, Denmark: A Quantitative Assessment of Economic and Environmental Aspects, *Journal of Industrial Ecology*, 10, pp. 239-255.

Kasat, R. B. and Gupta, S. K. (2003). Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator, *Computers and Chemical Engineering*, 27, pp. 1785-1800.

Lou, H. H., Kulkarni, M. A., Singh A. and Hopper, J. R. (2004). Sustainability Assessment of Industrial Systems, *Industrial and Engineering Chemistry Research*, 43, pp. 4233-4242.

LPG World (2004). LPG World – Prices Monthly, *Petroleum Argus*, X, 12, pp. 1-16.

Luus, R. and Jaakola, T. H. I. (1973). Optimization by Direct Search and Systematic Reduction of Size of Search Region, *AIChE Journal*, 19, pp. 760-766.

Maraschin, N. (2006). *Polyethylene, Low Density (LDPE)*, Kirk-Othmer Encyclopedia of Chemical Technology, Vol. 20, 5th Edition, John Wiley, New Jersey.

Pintarič, Z. N. and Kravanja, Z. (2006). Selection of the Economic Objective Function for the Optimization of Process Flow Sheets, *Industrial and Engineering Chemistry Research*, 45, pp. 4222-4232.

Rangaiah, G. P. (1985). Studies in Constrained Optimization of Chemical Process Problems, *Computers & Chemical Engineering*, 9, pp. 395-404.

Read, N. K., Zhang, S. X. and Ray, W. H. (1997). Simulations of a LDPE Reactor Using Computational Fluid Dynamics, *AIChE Journal*, 43, pp. 104-117.

Singh, A. and Lou, H. H. (2006). Hierarchical Pareto Optimization for the Sustainable Development of Industrial Ecosystems, *Industrial Engineering Chemistry Research*, 45, pp. 3265-3279.

Tarafder, A., Rangaiah, G. P. and Ray, A. K. (2007). A study of finding many desirable solutions in multiobjective optimization of chemical processes, *Computers and Chemical Engineering*, 31, pp. 1257-1271.

Williams, T. J. and Otto, R. E. (1960). *A Generalized Chemical Processing Model for the Investigation of Computer Control*, AIEE Trans, 79, pp. 458-473.

Exercises

10.1 In the Williams and Otto (WO) process solved in this chapter and also in Pintarič and Kravanja (2006), fixed capital investment, FCI is given as $600V\rho/0.453$ where ρ = 801 kg/m^3. However, the investment for equipment or plants is usually given by six-tenths rule (e.g., FCI α $V^{0.6}$). Hence, modify the FCI = $(600\beta\rho/0.453)$ $V^{0.6}$, and show that β = 1.967 so that FCI by the original equation and the modified equation are same at V = 5.425 m^3 (which is the mean of the lower and upper bounds for V). Optimize the WO process for multiple objectives as was done in this chapter but using the modified equation for FCI; a reliable MOO program such as NSGA is recommended for this. Compare the results obtained with those in this chapter, and explain the differences, if any.

10.2 In this chapter, PBP (= FCI/CF) did not consider the time value of money. In order to factor in the time value of money, future CFs have to be brought to the current value using the present worth annuity factor, $f_{PA}(r_d)$ given by:

$$f_{PA}(r_d) = \frac{(1+r_d)^{10}-1}{r_d(1+r_d)^{10}}$$

where r_d is the discount rate and can be taken as 0.12 yr^{-1}. The discounted payback period (DPBP) gives the time required to recover the FCI with all CF discounted back to time zero. It can be found by solving the non-linear equation:

$$FCI = CF\left[\frac{(1+r_d)^{DPBP}-1}{r_d(1+r_d)^{DPBP}}\right]$$

Derive the following iterative equation for finding DPBP and check its convergence.

$$DPBP_{i+1} = \frac{\ln\left\{\frac{FCI\left[r_d(1+r_d)^{DPBP_i}\right]}{CF}+1\right\}}{\ln(1+r_d)}$$

Optimize the WO process for multiple objectives with DPBP as one of them. Compare the results with those obtained in the chapter, and explain the effect of accounting the time value of money, if any.

10.3 Consider the 4-plant IE of Singh and Lou (2006) shown in Figure E.3. The model equations are the same as those given for the IE with 6 plants, i.e. equations 10.15 to 10.27. The symbols and data given in Tables 10.6 and 10.7 for Plants 1 to 4 are applicable for this model except that j should be 3, 1, 4 and 2 respectively for plants 1, 2, 3 and 4. The objectives are given in equations 10.30 to 10.32, where the terms should now be summed from i = 1 to 4 only. Cases A and B shall be considered for this exercise. Check that there are 20 decision variables for this exercise.

As the Excel(R) Solver is only for single objective optimization, use the ε-constrained method and the 'Excel Solver.xls' on the CD to optimize the 4-plant IE for the two objectives as in Cases A and B. For the Solver to work reliably, number of decision variables should be limited. Thus, it is recommended to set $Z_{21} = Z_{32} = 0$ and $Z_{22} = Z_{31} = 1$ for Z = a, b, c and d. This would leave the capacities of the 4 plants (X_i) as the decision variables. Treat IEvP as the constraint and vary it in the range 1.213-1.419 for Case A and 1.220-1.321 for Case B, and observe the trends of the decision variables and the objective. Do they follow similar trends as the IE for 6 plants?

10.4 Optimize the IE with 4 plants described in the previous exercise, with respect to all 20 decision variables using a robust program (e.g., NSGA-II, NSGA-II-aJG), and note the trends of the decision variables and the objectives. Discuss these results with those obtained using the Excel(R) Solver.

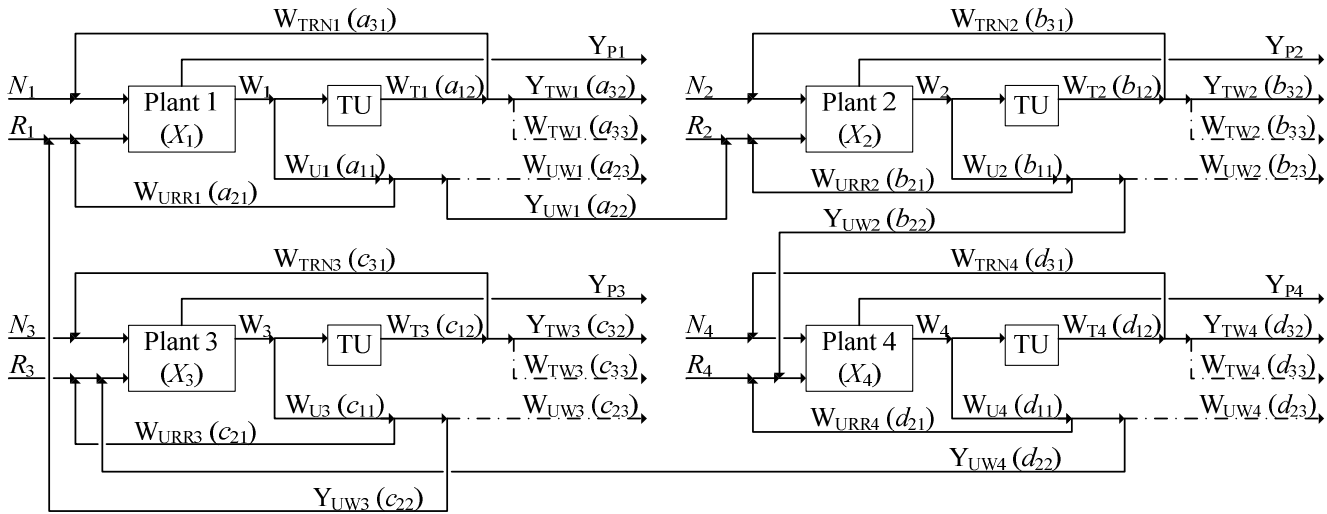


Fig. E.3 An IE system with 4 plants for exercise 10.3.

Chapter 11

Multi-Objective Emergency Response Optimization Around Chemical Plants

Paraskevi S. Georgiadou[a,c,*], Ioannis A. Papazoglou[b], Chris T. Kiranoudis[a] and Nikolaos C. Markatos[a]
[a]School of Chemical Engineering, National Technical University of Athens, Zografou Campus, Athens 157 80, Greece.
[b]Systems Reliability and Industrial Safety Laboratory, National Center of Scientific Research "Demokritos", Athens 153 10, Greece.
[c]Hellenic Institute for Occupational Health & Safety, Liossion 143 & Theirsiou 6, Athens 104 45, Greece.

*pgeor@central.ntua.gr

Abstract

The handling of certain quantities of hazardous materials (toxic, flammable and/or explosive) can potentially create major accidents endangering the public and worker's health, as well as the environment. Emergency response planning consists in assessing protective actions (evacuation, building protection of various degrees) for each and every area section around a hazardous facility. This chapter presents a methodology for the optimization of the response to an emergency situation around chemical plants processing hazardous substances (e.g. oil refineries, pesticide plants) by taking into account multiple criteria. A Multi-Objective Evolutionary Algorithm for the determination of the efficient set of solutions is presented.

Keywords: Emergency Planning, Decision Making, Major Accident, Evacuation, Multi-objective Optimization, Evolutionary Algorithm.

11.1 Introduction

In this work, we propose a methodology for multi-objective emergency response decision-making around chemical plants. In the past decades, a large number of severe accidents have been reported in the chemical industry and in the energy sector. Nuclear power plants as well as the handling or transportation of certain quantities of hazardous materials (toxic, flammable and/or explosive) can potentially create major accidents endangering the public and workers' health as well as the environment.

There are many safety and land-use planning regulations implemented to mitigate the risk. However, accidents may occur due to inadequate safety measures, human errors, etc. Therefore, apart from actions taken for the prevention of an accident, emergency response procedures as well as post accident actions (e.g., clean up of contaminated area) are of utmost importance for the protection of the population and the environment. Emergency planning and response decision-making consists of assessing protective actions (evacuation, shelter-in-pace, providing iodine tablets, use of personal protective equipment etc.) for the population adjacent to a hazardous facility.

Although there is no unique approach for emergency management in all countries, protective actions decision-making has been extensively recorded in the literature (see for example Sorensen *et al.,* 2004; Rogers *et al.,* 1990; Sinkko, 2004; Per Hedemann-Jensen, 2004). The factors influencing such decisions are numerous. These include the availability of communication, health care, transportation and shelter infrastructure, the level of training of the population and other organizational measures. One of the most important issues is the consequence assessment related to each protective action or to a combination of different protective actions implemented in the sub-areas around the plant. In the majority of emergency response approaches, the basic decisions are based on the intensity of the extreme phenomena following an accident. Calculations related to the specifics of the accidents (e.g., nuclear radiation, hazardous materials concentration, thermal radiation) lead to the specification of evacuation zones. In some of these approaches traffic factors related to the evacuees' movement are also taken into consideration. There are references concerning nuclear (Lindell, 2000;

Kirsteiger, 2006; Takada *et al.*, 2000; Gheorghe *et al.*, 1995; US Environmental Protection Agency, 2001) as well as hazardous materials emergencies (Brown *et al.*, 2007; Kiranoudis *et al.,* 2000; Keyworth *et al.*, 1992; Sorensen *et al.*, 1992, 2004; Geogriadou *et al.*, 2006).

Apart from the consequences assessment, other factors influencing emergency decision-making are the associated socioeconomic impacts of a protective action. The minimization of health consequences on the population and the minimization of the socioeconomic cost are generally conflicting objectives (Papazoglou and Kollas, 1997; Papazoglou and Christou, 1997; Pauwels *et al.*, 1996; Sinkko, 2004; Crick, 2004; Per Hedemann-Jensen, 2004; Govaerts, 2004; Georgiadou *et al.*, 2004). Therefore, an optimization methodology must be followed in order to support emergency decisions.

There have been references in the literature for multi-objective emergency response decision-making, mainly for nuclear power plants (Ishigami *et al.*, 2004; Schenker-Wicki, 1997; French, 1996; Hämäläinen *et al.*, 2000; Papamichail *et al.*, 2000, 2005; Sinkko, 2004; Mustajoki *et al.*, 2007). In most of them, the decision analysis format consists in the establishment of value trade offs among the various attributes measuring the consequences, so that a single index evaluator (through an appropriate value function) is obtained. In order to avoid the use of value trade-offs, which involves subjective judgments (sometimes called moral judgments), an alternative approach can be followed until the very late stage of the decision process through the use of the concept of *dominance*. A solution is called *non-dominated* (or *efficient*) if no other solutions in the search space are superior, when all objectives are considered (Miettinen, 1999). The set of all non-dominated solutions is called the *Pareto Optimal set* (or the *efficient frontier*). This approach is used for example by Papazoglou and Kollas (1997) and Papazoglou and Christou (1997). They use the multi-attribute utility theory in order to define the non-dominated set of all emergency plans in the event of a nuclear accident.

The methodology proposed in this chapter, is based on the determination of the Pareto Optimal set of solutions for multi-objective emergency response decision-making around chemical plants. Given the size of the decision space, the efficient set of solutions cannot be

determined through straightforward evaluation of the consequences of each alternative point in the decision space and inter comparison. An optimization algorithm is therefore necessary. Given the non-linear nature of the dependence of the consequences on the assumed alternative solution (i.e. emergency response plan), a Multi-Objective Evolutionary Algorithm (MOEA) approach has been chosen as described later in the paper. Evolutionary Algorithms (EA) belong to a class of stochastic optimization methods that simulate the process of natural evolution. Several EAs have been proposed for solving single or multi-objective optimization problems (Coello Coello, 2005; Jaimes and Coello Coello, 2008). They are used widely to solve combinatorial problems due to the fact that they deal simultaneously with a set of possible solutions allowing several members of the Pareto optimal set to be found in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional programming techniques.

The chapter consists of six sections. Section 11.2 describes the general structure of the methodology proposed. Section 11.3 discusses the approach of the consequence assessment of the problem. Section 11.4 presents the MOEA of emergency planning and response optimization around chemical plants. Section 11.5 includes some applications, and finally, section 11.6 offers a summary and the conclusion of this work.

11.2 Multi-Objective Emergency Response Optimization

11.2.1 *Decision Space*

An emergency plan in an area around a chemical plant is defined as follows (see Fig. 11.1): the area of interest is divided into a number of sub-areas of arbitrary shape and size. Several discrete actions are then defined such as *continuation of normal activities* or a protective action such as *evacuation, sheltering in large buildings, sheltering in houses etc*. An *Emergency Response Plan (ERP)* is defined when an action has been determined at each and every node of the area. This creates a discrete space of high dimensionality. The optimum or most preferred ERP is to be selected out of this decision space (Georgiadou *et al.*, 2004, 2006).

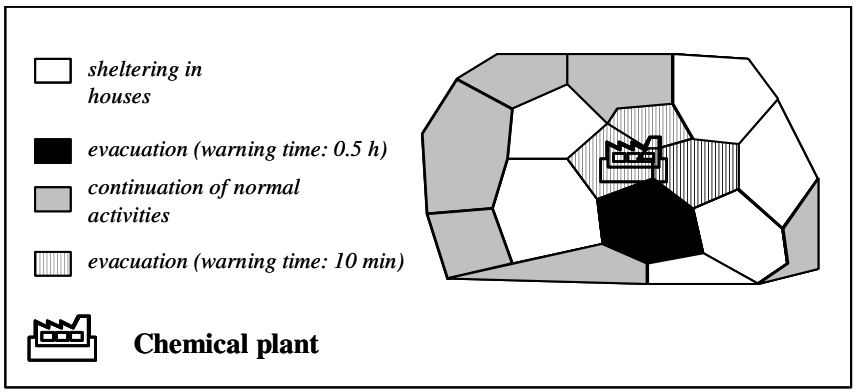


Fig. 11.1 Example of an emergency response plan in an area around a chemical plant.

11.2.2 *Consequence Space*

One major class of consequences is the adverse *health effects* including acute, latent fatalities and injuries. Another class of consequences addresses the so called *socioeconomic costs* of the ERP including psychological effects on the population subject to the ERP, social and economic effects of disruption of normal, everyday activities, etc. A set of distinct attributes, each measuring the degree to which each area of concern is affected as a result of the established ERP, is thus determined. As a result, each decision leads to a multidimensional consequence.

11.2.3 *Determination of the Pareto Optimal Set of Solutions*

A *MOEA* was developed for the determination of the Pareto Optimal set of solutions, i.e. the set of emergency plans that are non-dominated in all criteria. The EA approach was followed considering the nature of the problem, which means that consequences can be more accurately determined when the whole ERP has been determined.

EAs are stochastic optimization methods that simulate the process of natural evolution (Van Veldhuizen, 1999). The basic principles of these algorithms are the following:

- A possible solution of a problem is called the "*individual*" and a set of solution candidates is called the "*population*".
- An "individual" encodes the corresponding decision vector into a chromosome based on an appropriate structure (chromosome design and its parameters are specific to the problem to be solved.).
- Every "individual" is assigned a scalar value, the so-called "*fitness*", with respect to the optimization task, which represents the "individual"'s quality.
- The initial "population" undergoes a "*selection*" procedure and is manipulated by genetic operators, usually "*recombination*" ("*crossover*") and "*mutation*". The "individuals" that are going to be selected and reproduced, i.e. high-quality "individuals", are chosen according to their "fitness". Crossover operation leads to the generation of new chromosomes (children) by combining parts of parent chromosomes. The mutation operator involves the probability that one or more bits in a genetic sequence will be changed from its original state.
- Each loop consisting of the steps mentioned above is called a "*generation*".

In the last decades, and especially after 1990, several EAs have been proposed for solving multi-objective optimization problems. Surveys of MOEAs can be found in the literature (e.g., Coello Coello, 1999, 2005; Coello Coello *et al.*, 2002; Deb, 2001; Jaimes and Coello Coello, 2008). The main motivation for using MOEA to solve problems is the fact that they deal simultaneously with a set of possible solutions allowing to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional programming techniques (Coello Coello, 2005; Miettinen, 1999). In addition, they can easily deal with discontinuities and concave Pareto fronts (Coello Coello, 1999; Coello Coello *et al.*, 2002; Coello Coello, 2005; Deb, 2001).

11.2.4 *General Structure of the Model*

The main algorithm has been written in Visual C++ 6.0 and incorporates both the consequence assessment models as well as the MOEA. The general structure of the methodology is shown in Fig. 11.2. The following sections of this chapter describe the main parts of the model.

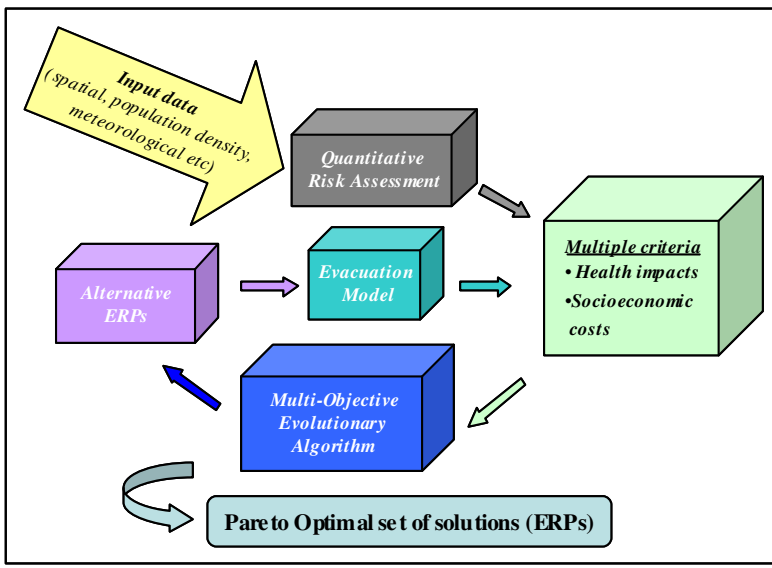


Fig. 11.2 Basic components of the algorithm for multi-objective emergency response optimization around chemical plants.

11.3 Consequence Assessment

11.3.1 *Assessment of the Health Consequences on the Population*

The use of dangerous substances in chemical plants such as oil refineries, pesticide plants etc., can cause major accidents (toxic releases, fires, explosions etc.). Such an accident might result in adverse effects on the health of the workers at the chemical plant as well as on the population in the area around of it.

One of the most important criteria used for the evaluation of an ERP is the *minimization of the consequences* resulting from the exposure to hazardous materials after an accident. In order to determine these consequences, we need to know the number of people in each sub-area and at each instant of time. The spatial population density changes with time during evacuation depending on several factors such as its initial distribution, the resulting level of congestion, the location of evacuees' final destinations etc. Therefore, in some cases the ERP considered by taking into account only the intensity of the extreme phenomena following an accident does not necessarily lead to the minimization of the accident consequences.

Given a specific accident (considering the amount and type of hazardous material, the type and characteristics of the accident, the environmental characteristics such as the wind speed and direction etc.), a spatial and temporal profile of the extreme condition, corresponding to each point in the area as a function of time, can be calculated by using appropriate models. According to the type of the accident (toxic release, fire and explosion), these profiles might correspond to the concentration of a toxic substance, the intensity of heat radiation or the size of overpressure (Papazoglou *et al.*, 1992; TNO, 1989, 1997).

These profiles allow the calculation of the dose to an individual if his/her time history in the area of interest is known. The dose received at location *(x, y)* is calculated by:

$$D(x,y) = \int_0^{Te} f\{c(x,y,t)\}dt \qquad (11.3)$$

where $c(x,y,t)$ is the intensity of the adverse effect (e.g. concentration of toxic material, heat radiation and overpressure) at point *(x, y)* and instant of time t, and T_e is the duration of the exposure.

Given that an individual is exposed to a *dose D*, the respective *conditional probability of consequence* P_D (e.g., death, several types of injuries etc.), can be calculated through the corresponding "*probit functions*" (Papazoglou *et al.*, 1992; TNO, 1989) as follows:

$$P_D = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Y-5} exp\left(-\frac{u^2}{2}\right) du \tag{11.4}$$

Here, Y is the probit variable given by:

$$Y = A + B \cdot ln(D) \tag{11.5}$$

where A *and* B are constants depending on the type of accident (exposure to specific toxic materials, heat, pressure, etc.).

Let N_o be the number of people exposed to a dose D. Then, the average expected number of people subject to the respective consequence (N_c) will be:

$$N_c = N_o \cdot P_D \tag{11.6}$$

From the dose calculation, the risk of specific adverse effect can be calculated and hence the consequences in the multidimensional space discussed above. Thus, given a specific ERP, the consequences in the form of specific values of the established set of attributes are determined and this creates the desired mapping between the decision and the consequence space.

The health consequences on the population must be estimated by taking into consideration the type of the accident and the population distribution in the area as a function of time. To calculate the variation of the spatial population density, we have developed a stochastic model that simulates the evacuation procedure. More precisely, we have adopted a *Markovian type* stochastic model to simulate the movement of the population (Georgiadou *et al.*, 2006).

For simulating the population movement, the area of interest is divided into N discrete nodes. A node represents a geographic area of variable (user selected) size and shape, i.e. it may represent an area including one or more building blocks. The nodes are connected between each other by links. A link might represent a single road between two building blocks. However, if the node corresponds to more than one building block, then the link represents the capability of connection between two adjacent nodes. In the latter case, the characteristics of a link used in the model (distance, capacity) are estimated by considering the corresponding values of the relevant existing roads in the area.

A discrete state stochastic Markov process simulates the movement of the evacuees. Transition from node-to-node is simulated as a random process where the probability of transition depends on the dynamically changed states of the destination and origin nodes and on the link between them. Solution of the Markov process provides the expected distribution of the evacuees in the nodes of the area as a function of time.

Let T be the total time steps considered for the evacuation procedure. The expected dose (D_T) of an individual which participates in the evacuation procedure moving from one node to another in order to reach his/her final destination, is calculated as follows: first, the average dose [$D(t_n)$] received by an individual at time t_n ($n=1,\ldots,T$) is calculated by averaging out over all nodes in the area under study:

$$D(t_n) = \sum_{i=1}^{N} x_i(t_n) \cdot D_i(t_n) \tag{11.7}$$

where $x_i(t_n)$ is the probability that an individual who takes part in the evacuation procedure was in node i ($i=1,\ldots,N$) at time t_n, estimated by the Markov evacuation model. Then the overall average dose received by an individual over the total duration of the evacuation is:

$$D_T = \sum_{n=1}^{T} D(t_n) \tag{11.8}$$

Next, P_D is obtained by virtue of Eqs. (11.4)-(11.5). Similar analysis is made for people not participating in the evacuation; for them, Eq. (11.7) is used only for the node which they occupy for the whole period T_e.

A Monte-Carlo solution of the evacuation model provides in addition a sample of actual trajectories of the evacuees. This information coupled with an accident analysis which provides the spatial and temporal distribution of the extreme phenomenon following an accident, determines a sample of the actual doses received by the evacuees. Both the average dose and the actual distribution of doses can be used as measures in evaluating alternative emergency response strategies.

11.3.2 *Socioeconomic Impacts*

The socio-economic impacts of an emergency response plan relate to social agitation (industry production, social network), anxiety of the workers and the population at large, monetary costs (direct and indirect costs of protective actions) etc. In the proposed methodology, the number of people participating in the evacuation procedure and the number of people to be sheltered measure the socio-economic impacts of an ERP. More precisely, the number of people taking part in a protective action is to be minimized (see also Papazoglou *et al.,* 1997).

11.4 A MOEA for the Emergency Response Optimization

11.4.1 *Representation of the Problem*

In the methodology proposed in this chapter, each alternative ERP corresponds to a *chromosome* having as many *genes* as nodes in the area of interest. A similar scheme is the "land-block" representation for multi-objective land-use planning (Matthews *et al.*, 2000; Datta *et al.*, 2006). The value of each gene is determined according to the action applied to the corresponding node (a protective action or continuation of normal activities). A different value of a gene might correspond to the same protective action (e.g. evacuation) but with different warning times, i.e. the time when the population movement begins after the accident. Figure 11.3 shows a part of an area corresponding to a part of a chromosome. The land blocks in which a different protective action is applied correspond to different values of the genes in a chromosome.

11.4.2 *General Structure of the MOEA*

The algorithm developed for the multi-objective emergency response optimization is based heavily on SPEA II, which is an improved algorithm of SPEA proposed by Zitzler *et al.* (2001). An initial "population" (mating pool) of ERPs is evolved through the generation of new chromosomes via recombination and mutation of genes of the parent "population". An *archive (external set)* is maintained, which

contains the non-dominated front among all solutions considered up to time (t), and it is renewed at every generation. This procedure continues until no further or minimal change is achieved in the non-dominated "population" from one generation to the other. The main steps of the algorithm are presented in the following subsections.

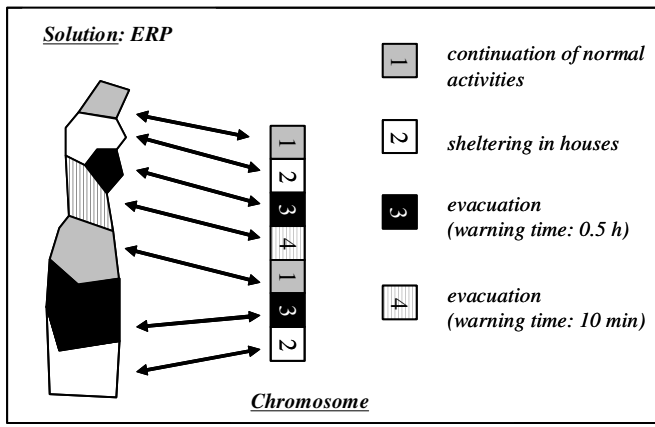


Fig. 11.3 Representation scheme of the emergency response plan (ERP).

11.4.3 *Initialization*

The input data concerning the "population" (mating pool) and the archive (external set) are provided: N - "*population*" size *(P)*; N' - archive size *(E)*; n - maximum number of generations; t - generation index *(t=0,…,n)*. In the first step ($t = 0$), the initial "population" $P(0)$ is created and also an empty archive $E(0)$. The former is created either randomly or by sampling solutions from an archive (e.g., solutions from the previous runs of the algorithm).

11.4.4 *"Fitness" Assignment*

At this step, "fitness" values of "individuals" in both $P(t)$ and $E(t)$ are calculated. For each "individual", the "*fitness*" *assignment scheme* takes into account how many "individuals" it dominates and it is dominated

by, and its distance (in objective space) to all "individuals", considering both the members of the parent "population" and the archive. More precisely, each "individual" i in the archive $E(t)$ and the "population" $P(t)$ is assigned a value called strength according to:

$$S(i) = \left| \left\{ j \middle| j \in P(t) + E(t) \wedge i \succ j \right\} \right| \tag{11.9}$$

where $i \succ j$ means that i dominates j. Then, the "*raw fitness*" of every "individual" is calculated according to:

$$R(i) = \sum_{j \in P(t)+E(t), j \succ i} S(j) \tag{11.10}$$

In order to discriminate between "individuals" having identical "raw fitness" values, additional density information is incorporated. Let $F(i)$ be the "fitness" of each "individual" i in both $P(t)$ and $E(t)$. It is given by:

$$F(i) = R(i) + D(i) \tag{11.11}$$

where $D(i)$ is the density of "individual" i.

The density estimation technique used in SPEA2 and in our algorithm is an adaptation of the kth nearest neighbor method (Silverman, 1986), where the density at any point is a (decreasing) function of the distance to the kth nearest data point. For each "individual" i, the distances (in objective space) to all "individuals" j in archive and "population" are calculated. A ranking algorithm is used in order to sort in increasing order the distances of "individuals". The distance-sought, σ_i^k corresponds to the kth element, where k is given by Eq. (11.12) (Silverman, 1986), and $D(i)$ is given by Eq. (11.13):

$$k = \sqrt{N + N'} \tag{11.12}$$

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{11.13}$$

The run-time of the "fitness" assignment procedure is dominated by the density estimator (Zitzler, 2001), while the calculation of the S and R values is of complexity $O(M^2)$ where $M = N + N'$.

11.4.5 *Environmental Selection*

Let $N''(t)$ be the number of non-dominated "individuals" determined up to time (t). If $N''(t) \leq N'$, then according to the "fitness" value of each "individual", solutions (both in the "population" and the archive) are sorted in an increasing order. Afterwards, the first N' "individuals" from $P(t) + E(t)$, i.e. all non-dominated and the best dominated "individuals", are copied to the archive of the next generation ($E(t+1)$). On the other hand, if $N''(t) > N'$, a *truncation method* is used to reduce the number of non-dominated "individuals". More precisely, the distances from every non-dominated "individual" to all others are calculated. The "individuals" with the smallest distance are removed (until the number of the remaining non-dominated "individuals" reaches N').

As mentioned by Zitzler *et al.* (2001), although the worst run-time complexity of the truncation operation is $O(M^3)$, on average the complexity will be lower as "individuals" usually differ with regard to the second or third nearest neighbour, and thus sorting of the distances governs overall complexity. In our program, a slightly different approach was used in order to reduce computational cost. The distances of all non-dominated "individuals" are calculated and sorted. Then, the first N' "individuals" corresponding to the highest distances are copied to $E(t+1)$.

11.4.6 *Termination*

The termination criterion used for a single run of the algorithm is the reaching of the maximum number of generations ($t \geq n$).

11.4.7 *Mating Selection*

In this step, a selection method is used in order to fill the mating pool, $P(t+1)$). A variety of selection methods can be used in EAs. Two classical selection schemes are the roulette wheel selection and the tournament selection. In the first scheme, chromosomes are chosen according to a given probability, which is a function of their "fitness". In the second scheme, a number of chromosomes are randomly chosen from the previous generation.

In our MOEA methodology, we use a combination of roulette wheel and tournament selection approaches according to:

$$index_i = (1 - \sqrt{Rand}\,) * (N - 1) \tag{11.14}$$

where $i = 1,...,N$, $Rand$ is a random number, $index_i$ is the chromosome to be chosen for reproduction. Due to the fact that the "individuals" have been ranked in the previous step according to their "fitness", solutions with better "fitness" values (i.e., lower in case of minimization) have a greater probability to be chosen for reproduction.

11.4.8 *Variation*

In this step, recombination (crossover) and mutation operators are applied to the "individuals" selected for reproduction in the previous step. The number of chromosomes, which will be reproduced by crossover operation, as well as, crossover points, can be determined either deterministically by the user or randomly. For example, in case of one-point crossover (see for example in Fig.11.4) the chromosomes of the parents are cut at some randomly chosen point and the resulting sub-chromosomes are swapped. As far as the mutation operator is concerned, the user gives the number of genes to be changed per generation. The points to be mutated are chosen either randomly or deterministically according to the accident scenario considered (see for example in Fig. 11.4).

11.5 Case Studies

Different ERPs (i.e., different areas of evacuation and, hence, different number of evacuees), result in different population distributions as a function of time, and thus in different doses and corresponding health consequences. In this section, some case studies are presented in order to demonstrate our methodology for multi-objective emergency response optimization around chemical plants. Two objectives are considered, i.e. minimize both the expected number of fatalities and the number of evacuees. The cases presented correspond to different types of accidents and different circumstances related to the emergency response issues

(i.e. population density in the area, capacity of the road network, warning time before the accident).

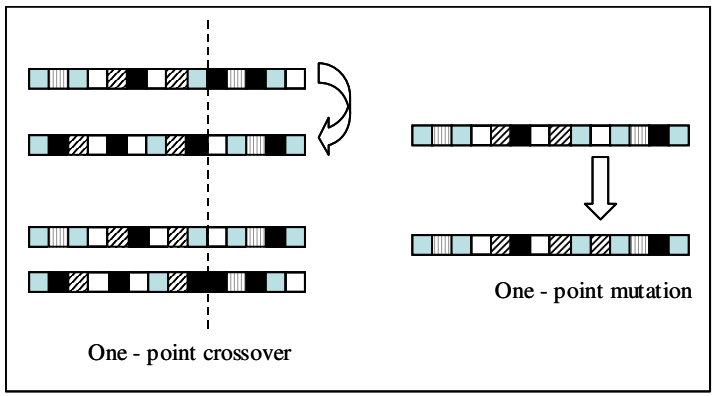


Fig. 11.4 Examples of one-point crossover and one-point mutation.

We consider one example of an accidental release from an ammonia storage plant in a densely populated area (2.916×10^7 m^2). Many serious accidents involving ammonia have been reported in the literature (Michaels, 1999). The area is divided into 361 discrete nodes of equal size. The resulting total number of links is 760, assuming one-direction movement of vehicles away from the hazardous facility. The scenario considered corresponds to a 264 kg/sec release rate of ammonia and release duration of 600 sec. With an appropriate model (Papazoglou *et al.*, 1996), we calculate the concentration of ammonia (gas heavier than air) as a function of time, as well as the arrival and the passage time of the plume at every point of the area. According to the procedure described in section 11.3, the consequence attributes for each solution are estimated.

In Fig. 11.5, the evolution of an initial "population", $P(0)$ until the 8th generation, $P(8)$ is represented. As shown in this figure, the algorithm creates better solutions according to the criteria taken into consideration between two successive generations. The non-dominated set of solutions (ERPs) of this specific accident scenario, after 100 generations, is also shown in Fig. 11.5. The EPR to be applied in the case of this accident

scenario can be chosen according to the preferences of the decision maker. The optimal solution subject to the health consequences is the evacuation of the whole area (100% of the population). However, even in this case, the number of fatalities is not equal to zero because some people who are very close in the source of the ammonia are going to receive immediately a lethal dose after the accidental release.

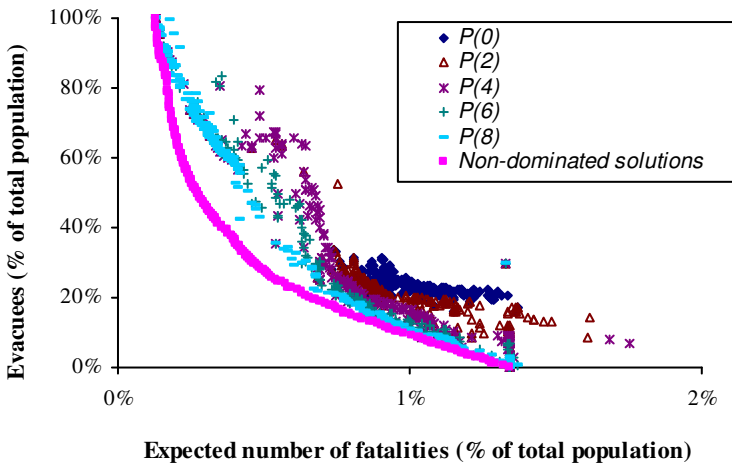


Fig. 11.5 Evolution of "population" of emergency response plans.

We have considered also an accident scenario in an oil refinery in Thriassion Pedion (West Attica, Greece). It is a densely populated area including many large, medium and small enterprises. One possible accident that can take place is a Boiling Liquid Expanding Vapor Explosion (BLEVE). Some of them (such as oil refineries, chemical plants etc.) use large amounts of hazardous substances. The area is divided into 992 discrete nodes with different shape and size. The resulting total number of links is 1384, assuming one-direction movement of vehicles away from the hazardous facility (oil refinery). A BLEVE in a tank (3945 m^3) containing propane and located at the oil refinery is assumed. The resulting heat flux at every node can be calculated (Papazoglou *et al.*, 1996). Consequently, the dose and the corresponding fatality probability at each node can be calculated

according to the methodology described in Section 11.3. The evacuation model provides the spatial distribution of the population at the time when the explosion occurs. The assumption is that the evacuation begins before the explosion, i.e. there is a "warning time" before the accident.

In Fig. 11.6, the non-dominated set of ERPs is presented in the case of a "warning time" of 1800 sec, i.e. the explosion occurs *1800 sec* after the beginning of the actual evacuation. The EPR to be applied in the case of this accident scenario can be chosen according to the preferences of the decision maker. It is important to mention that the methodology gives the opportunity to the decision maker to compare with other solutions which typically are chosen in case of emergencies. For example, according to a methodology for the application of the Seveso II Directive in Greece (Council Directive 96/82/EC, 1996; Kiranoudis *et al.*, 2000, 2002), a decision maker typically would evacuate the population residing in this zone, which corresponds to a distance from the hazardous source where the fatality probability is greater or equal to 10^{-5} (2525 meters from the hazardous source for this specific BLEVE scenario). As it is shown in the diagram, in this case the solution corresponding to a "typical evacuation zone" is dominated with respect to all criteria (see Fig. 11.6).

Other factors that could be evaluated by the decision maker by using the MOEA are the warning time and the capacity of the roads. In Fig. 11.7, the non-dominated set of ERPs is estimated in the case of greater "warning time", i.e., 3600 sec. It is shown that the increase of the "warning time" results in better solutions with respect to the number of fatalities (i.e. solutions with the same socioeconomic cost correspond to lower number of fatalities). In addition, the non-dominated set of ERPs for the same accident scenario is shown in case of a much greater "capacity" of the transportation network (Fig. 11.8). The shorter evacuation time related with the decrease of the level of congestion, results to non-dominated solutions which correspond to zero number of fatalities (i.e. in the cases presented in Figs. 11.6 and 11.7, there is not enough time for the population to evacuate the hazardous area, and thus, when the explosion occurs, some people are still in the vicinity of the hazardous source).

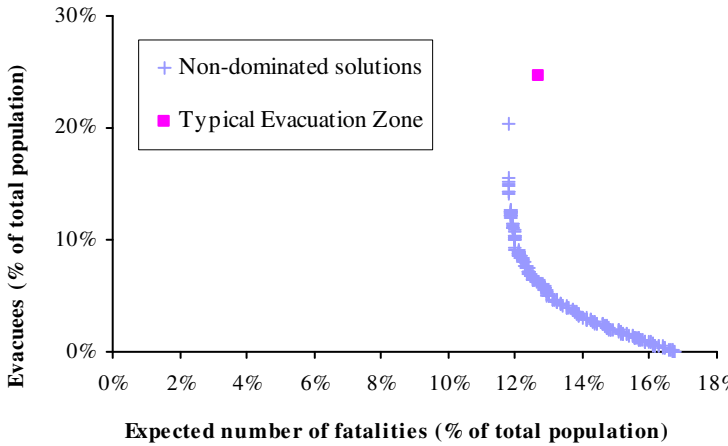


Fig. 11.6 Non-dominated solutions (ERPs) in case of a BLEVE phenomenon in an LPG storage tank ("capacity" of the transportation network: 1, "warning time": 1800 sec).

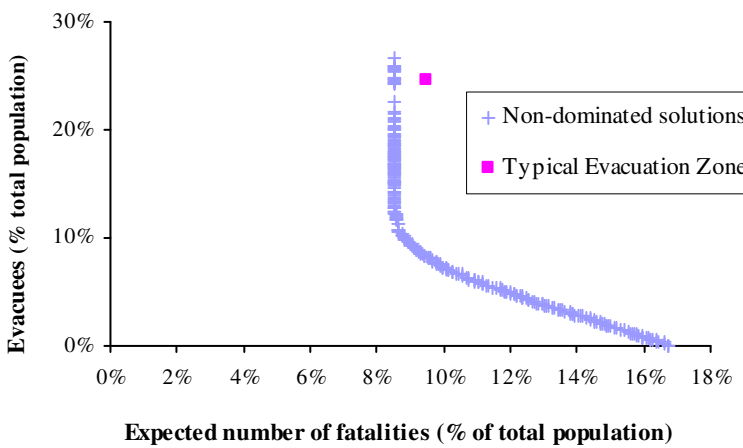


Fig. 11.7 Non-dominated solutions (ERPs) in case of a BLEVE phenomenon in an LPG storage tank ("capacity" of the transportation network: 1, "warning time": 3600 sec).

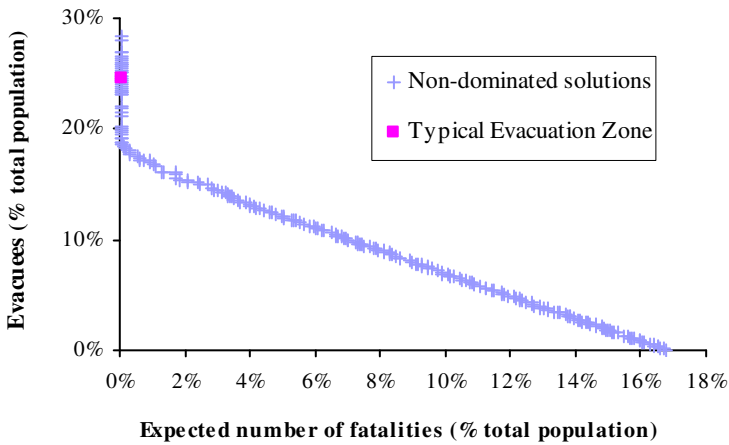


Fig. 11.8 Non-dominated solutions (ERPs) in case of a BLEVE phenomenon in an LPG storage tank ("capacity" of the transportation network: 5, "warning time": 1800 sec).

11.6 Conclusions

The handling of certain quantities of hazardous materials (toxic, flammable and/or explosive) can potentially create major accidents endangering the public and worker's health, as well as the environment. Emergency response planning consists in assessing protective actions (evacuation and building protection of various degrees) for each and every area section adjacent to a hazardous facility. In general, in evaluating the effectiveness of alternative protective actions, a decision maker is faced with several, usually conflicting objectives, including the minimization of the potential health effects of an accident, as well as the minimization of the associated socioeconomic impacts of an emergency plan. Even if only the minimization of various health consequences is considered as an evaluator of alternative emergency response plans, several factors influencing the extent of the consequences must be taken into consideration. An estimation of the dynamic movement of the population at risk, if the protective action includes evacuation, is necessary, coupled with information concerning the spatial and temporal distribution of the extreme phenomenon following an accident (e.g. toxic cloud and thermal radiation).

This chapter presents a methodology for the optimization of the response to an emergency situation around chemical plants processing hazardous substances (e.g. oil refineries and pesticide plants) by taking into account multiple criteria. A description of a MOEA for the determination of the efficient set of solutions to the problem of emergency response and some case studies are included in this chapter. The methodology presented can be used for supporting decisions for emergency response around installations using hazardous substances in a pre-planning phase, and in real-time emergency response procedures in cases there is enough warning time. It could also be useful for land-use planning issues such as providing information about the need for increasing the capacity of the transportation network or the need for safe shelters for the population.

Acknowledgements

The research reported in this paper is performed by P. S. Georgiadou as a partial fulfillment of the requirements for the Degree of Doctor of Philosophy at the National Technical University of Athens. The research was performed at the Systems Reliability and Industrial Safety Laboratory (National Center of Scientific Research "Demokritos", Greece) and at the Computational Fluid Dynamics Unit (School of Chemical Engineering, National Technical University of Athens).

References

Brown, D.F. and Dunn, W.E. (2007). Application of a quantitative risk assessment method to emergency response planning, *Computers & Operations Research*, 34 (5), pp. 1243-1265.

Coello Coello, C.A. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems,* 1 (3), pp. 269-308.

Coello Coello, C.A. (2005). Recent trends in evolutionary multiobjective optimization, in: Abraham, A., Jain, L. and Goldberg, R. (editors), *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*, Springer-Verlag, London.

Coello Coello, C.A., Van Veldhuizen, D.A. and Lamont, C.B. (2002). Evolutionary algorithms for solving multi-objective problems, Dordrecht, Kluwer Academic Publishers.

Council Directive 96/82/EC of 9 December 1996 on the control of major-accident hazards involving dangerous substances (1997). *Official J. Eur. Communities*, Luxembourg.

Crick, M., McKenna, T., Buglova, E., Winker, G. and Matrincic, R. (2004). Emergency management in the early phase. *Radiation Protection Dosimetry,* 109, pp. 7-17.

Datta, D., Deb, K., Fonseca, K.M., Lobo, F. and Condado, P. (2006). Multi-objective evolutionary algorithm for land-use management problem. KanGAL Report number 2006005.

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. New-York, Wiley.

French, S. (1996). Multi-attribute decision support in the event of a nuclear accident. *J. of Multi-Crit. Decision Anal.*, 5, pp. 39-57.

Georgiadou, P.S., Papazoglou, I.A., Kiranoudis, C.T. and Markatos, N.C. (2004). Emergency response optimization for major hazard industrial sites. In: Spitzer, C., Schmocker, U. and Dang, V.N. (editors). *Proc. of Int. Conf. Probabilistic Safety Assessment and Management (PSAM7 - ESREL '04),* Berlin, Germany, pp. 128-133.

Georgiadou, P.S., Papazoglou, I.A., Kiranoudis, C.T. and Markatos, N.C. (2007). Modeling emergency evacuation for major hazard industrial sites, *Reliability Engineering & System Safety*, 92(10), pp. 1388-1402.

Gheorghe, A.V. and Vamanu, D. (1995). A pilot decision support system for nuclear power emergency management, *Safety Science*, 20, pp. 13-26.

Govaerts, P. (2004). Using plant status data for off-site emergency management: technical and political aspects. *Radiation Protection Dosimetry,* 109 (1-2), pp. 31-33.

Hämäläinen, R.P., Lindstedt, M.R.K. and Sinkko, K. (2000). Multiattribute risk analysis in nuclear emergency management. *Risk Analysis,* 20, pp. 455-467.

Ikeda, Y., Beroggi, G.E.G. and Wallace, W.A. (1998). Supporting multi-group emergency management with multimedia. *Safety Science,* 30, pp. 223-234.

Ishigami, T., Kobayashi, K., Umemoto, M. and Matsunaga, T. (2004). A simplified simulation method for selecting the most effective off-site protection. *Reliability Engineering and Systems Safety*, 86, pp. 61–24.

Jaimes, A.L. and Coello Coello C.A. (2008). Multi-objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering, in: Rangaiah, G.P. (editor), *Multi-objective Optimization: Techniques and Applications in Chemical Engineering,* World Scientific.

Kenney, R.L. and Raiffa, H. (1976). Decisions with multiple objectives: preferences and value tradeoffs, John Wiley & Sons.

Keyworth, C.J. and Smith, D.G. (1992). Emergency notification under SARA Title III: impacts on facility emergency planning, *J. of Hazardous Materials*, 31, pp. 241-253.

Kiranoudis, C.T., Markatos, N.C., Zografos, K. and Ziomas, I. (2000). An operational centre for managing major chemical industrial accidents. *Proc. of Eur. Conf. Seveso 2000*, Athens, Greece.

Kiranoudis, C.T., Kourniotis, S.P., Christolis, M., Markatos, N.C., Zografos, K.G., Giannouli, I.M., Androutsopoulos, K.N., Ziomas, I., Kosmidis, E., Simeonidis, P. and Poupkou, N. (2002). An operational centre for managing major chemical industrial accidents, *J. of Hazardous Materials*, A89, pp.141–161.

Kirchsteiger, C. (2006). Current practices for risk zoning around nuclear power plants in comparison to other industry sectors, *J. of Hazardous Materials*, 136(3), pp. 392-397.

Lindell, M.K. (2000). An overview of protective action decision-making for a nuclear power plant emergency, *J. of Hazardous Materials,* 75, pp. 113-129.

Matthews, K., Craw, S., Elder, S., Sibbald, A. and MacKenzie, I. (2000). Applying genetic algorithms to multiobjective land use planning, *Proc. of the Genetic and Evolutionary Computation Conf. (GECCO 2000)*. Morgan Kaufmann, San Francisco, pp. 613-620.

Michaels, R.A. (1999). Emergency planning and the acute toxic potency of inhaled ammonia, *Environ. Health Perspectives*, 107, pp. 617-627.

Miettinen, K. (1999). Nonlinear multiobjective optimization, Kluwer Academic Publishers, Boston, Massachusetts.

Mustajoki, J. Hämäläinen, R.P. and Sinkko, K. (2007). Interactive computer support in decision conferencing: two case on off-site nuclear emergency management, *Decision Support Systems*, 42 (4), pp. 2247-2260.

Papamichail, K.N. and French, S. (2000). Decision support in nuclear emergencies, *J. of Hazardous Materials*, 71, pp. 321-342.

Papamichail, K.N. and French, S. (2005). Design and evaluation of an intelligent decision support system for nuclear emergencies, *Decision Support Systems*, 41, pp. 84-111.

Papazoglou, I.A. and Christou, M. (1997). A decision support system for the emergency response to major nuclear accidents, *Nuclear Technology*, 118, pp. 97–21.

Papazoglou, I.A. and Kollas, J.G. (1997). Establishing protective long measures after severe nuclear accidents using multiple criteria, *Health Physics Society,* 72 (5), pp. 676-692.

Papazoglou, I.A., Aneziris, O., Bonanos, G. and Christou, M. (1996). SOCRATES: a computerized toolkit for quantification of the risk from accidental releases of toxic and/or flammable substances, *International J. of Environmental Pollution*, 6 (4-6), pp. 500-533.

Papazoglou, I.A., Nivolianitou, Z., Aneziris, O. and Christou, M. (1992). Probabilistic safety analysis in chemical installations, *J. of Loss Prevention in the Process Industries*, 5, pp.181-191.

Pauwels, N., Hardeman, F. and Soudan, K. (1996). Assessing the economic impact of the decision to evacuate an industrial area. Do the existing models aplly? *Seminar of the Consequences of a Nuclear Accident: An Economic and Radiological Approach*, Belgian Association for Radiation Protection 9BVS/ABR: Brussels, Belgium.

Per Hedemann-Jensen. (2004). Protective actions in the late phase-intervention criteria and decision making, *Radiation Protection Dosimetry*, 109, pp. 45-51.

Rogers, G.O., Watson, A.P., Sorensen, J.H., Sharp, R.D. and Carnes, S.A. (1990). Evaluating protective actions for chemical agent emergencies, ORNL-6615, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Schenker-Wicki, A. (1997). The Swiss decision support system: state of the art and experiences, *Radiation Protection Dosimetry*, 73, pp. 119-122.

Silverman, B.W. (1986). Density estimation for statistics and data analysis, London: Chapman and Hall.

Sinkko, K. (2004). Nuclear emergency response planning based on participatory decision analytic approaches, PhD Thesis, Systems Analysis Laboratory, Helsinki University of Technology.

Sinkko, K., Hämäläinen, R.P. and Hänninen, R. (2004). Experiences in methods to involve key players in planning protective actions in the case of a nuclear accident, *Radiation Protection Dosimetry*, 109 (1-2), pp.127-132.

Sorensen, J. and Carnes, S. (1992). An approach for deriving emergency planning zones for chemical munitions emergencies, *J. of Hazardous Materials*, 30, pp. 223-242.

Sorensen, J., Shumpert, B. and Vogt. B. (2004). Planning for protective action decision making: evacuate or shelter-in-place, *J. of Hazardous Materials*, A109, pp. 1-11.

Sorensen, J.H., Vogt, B.M. and Mileti, D.S. (1987). Evacuation: an assessment of planning and research, Report prepared for U.S. Federal Emergency Management Agency, RR-9.

Takada, J. and Hoshi, M. (2000). External doses to 350m zone residents around the Tokai-mura criticality accident site, *J. of Environmental Radioactivity*, 50, pp. 43-48.

TNO, Committee for the Prevention of Disasters (1989). Methods for the determination of possible damage: to people and objects resulting from the releases of hazardous materials (Green Book), CPR-16E, The Netherlands.

TNO, Committee for the Prevention of Disasters (1997). Methods for the calculation of the physical effects resulting from the releases of hazardous materials (Yellow Book), CPR 14E, The Netherlands.

US Environmental Protection Agency (2001). Manual of protective action guides and protective actions for nuclear incidents, EPA 400-R-92-001.

Van Veldhuizen, D. (1999). Multi-objective evolutionary algorithms: classifications, analyses, and new innovations, PhD Thesis, Dayton, Ohio: Air Force Institute of Technology.

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: improving the strength pareto evolutionary algorithm, Technical Report 103, TIK, Swiss Federal Institute of Technology, Zurich.

Chapter 12

Array Informatics using Multi-Objective Genetic Algorithms: From Gene Expressions to Gene Networks

Sanjeev Garg
Department of Chemical Engineering
Indian Institute of Technology, Kanpur, UP, INDIA 208 016

E-mail: sgarg@iitk.ac.in

Abstract

cDNA microarray experiments produce expression ratios of thousands of genes across tens of experimental attributes. In this work, development of a robust clustering algorithm and a graph-theoretic model for reverse engineering the gene networks and their implementation using NSGA-II are reported. Clustering results on synthetic datasets as well as on a real life dataset are very encouraging. The clusters for synthetic as well as real life datasets obtained from the robust clustering are used for reverse engineering the gene regulatory networks using the graph-theoretic model inspired by 'small world phenomena'. A set of Pareto-optimal models have been proposed. The models generated for the real life dataset concur with the available biological information. Newer functionalities and interactions are also proposed that concur with the observed cDNA microarray data.

Keywords: cDNA Microarray, Gene Expression, Gene Network

12.1 Introduction

One of the most exciting challenges in computational biology is the task of deciphering underlying regulatory genetic networks mathematically from expression data resulting from large scale arraying techniques. cDNA microarray is a technique that provides a global analysis of gene expression at the level of transcription. Microarray studies in recent past have resulted in enormous amounts of gene expression data for several organisms under different conditions of interest. Moreover, these systems are characterized by a logical network of genes with mutual influence of expression levels. Reconstruction of the precise gene interaction network from the huge amount of expression data obtained from cDNA microarrays using mathematical and informatics tools is a challenging task and is an active area of research.

12.1.1 *Biological Background*

Cells are the structural and functional unit of all living organisms. All cells consist of a cell membrane, nucleus, cytoplasm and various organelles. The outer lining is called the cell membrane. The role of the cell membrane is to protect the cell from outside environment. The cell membrane also regulates the movement of water, nutrients and wastes into and out of the cell. Cytoplasm is a colloidal, semi-fluid matter contained within the cell membrane, in which all other cell organelles are suspended. Nucleus is a membrane enclosed organelle found in most eukaryotic cells. It contains most of the cell's genetic material in form of nucleic acids. RNA (ribonucleic acid, present in the nucleus as well as in the cytoplasm) and DNA (deoxyribonucleic acid, present in the nucleus) are two nucleic acids present in the cell and are responsible for transferring all the genetic information from a parent to the offspring. The amount of RNA produced by any cell is proportional to the status of that cell. Higher amount of RNA shows highly active cell, and the absence of RNA indicates totally inactive cell.

Genes are small segments of DNA. Gene expression is the process by which the set of instructions is read by the cell and translated into proteins, which perform specific functions within the cell. A DNA

molecule consists of two long strands wound tightly around each other in a form of a spiral structure, known as the double helix. Each single strand of the DNA molecule is made up of a linear sequence of four deoxyribo-nucleotides: adenine (A), guanine (G), cytosine (C) and thymine (T). The sequence of these bases is a linear code that leads to the synthesis of proteins through the cellular processes of *transcription* and *translation*. Gene expression begins with cells first transcribing the information from DNA into messenger RNA (mRNA), which is a sort of temporary copy of the gene when the DNA replicates. *Replication* involves unwinding of the double-stranded DNA helix, copying or replicating the master template itself so the gene expression can repeat in a new generation of cells or organisms. The mRNA so synthesized then separates from the DNA and transports out of the nucleus, across the nuclear membrane, and into the cellular cytoplasm. In the cellular cytoplasm, ribosomes act as sites for protein synthesis. Guided by ribosomal RNA (rRNA), a structural component of ribosome, mRNA binds to ribosome where the decoding of the coded information takes place. This coded information is actually formed by triplets of consecutive bases, called as codons, throughout the length of mRNA. Each triplet calls a particular amino acid, out of 20, that make up a protein molecule. This is the function of transfer RNA (tRNA) that carries amino acid element to the appropriate place as coded for by mRNA. This process is called translation. This series of gene expression events is commonly referred to as the *central dogma of molecular biology* (Crick, 1970).

The traditional methods of molecular biology were based on *one gene - one experiment* and the output was very limited. It was very difficult to understand about the various interactions. There was always a need for monitoring number of genes simultaneously. This has given birth to *Microarray Technology*. This technology, developed in past few decades, makes it possible for researchers to have a better picture of interactions among thousands of genes simultaneously, by monitoring the whole genome (genome of an organism is the whole hereditary information of an organism that is encoded in the DNA or, in RNA for some viruses) on a single chip. Currently, microarrays are widely used in

laboratories throughout the world to measure the expression levels of tens of thousands of genes simultaneously in a single chip.

Two single stranded DNA molecules whose sequences are complementary to each other will exhibit a tendency to bind together to form a single double stranded DNA molecule; this process is known as *hybridization* (Dhammika and Javier, 2004). Two DNA strands (or one DNA strand and other mRNA strand) will hybridize with each other, regardless of whether they are originated from a single source or two different sources, as long as their base pair sequence match according to the complementary base-pairing rules. A cDNA microarray (also known as gene chip, DNA chip, or biochip) is a collection of microscopic DNA spots attached to a solid surface, such as glass, plastic or silicon chip forming an array (Ingen, 2002; Dhammika and Javier, 2004). The affixed DNA segments are known as probes, thousands of which can be used in a single DNA microarray (Ingen, 2002; Dhammika and Javier, 2004). Microarray technology evolved from Southern blotting, where fragmented DNA is attached to a substrate and then probed with a known gene or fragment (Christian *et al.,* 2002). Measuring gene expression using microarrays is relevant to many areas of biology and medicine, such as studying treatments, diseases, and developmental stages (Lander, 1999; Dhammika and Javier, 2004). Scientists use DNA microarrays to measure the expression levels of large numbers of genes simultaneously or to genotype multiple regions of a genome (Brown and Botstein, 1999; Debouck and Goodfellow, 1999; Lander, 1999; Andrew *et al.,* 2002; Christian *et al.,* 2002). Applications of DNA microarrays to the study of gene expression include: to distinguish the functional differences between two or more genomic states (Brown and Botstein, 1999; Lander, 1999), to distinguish between cell phenotypes (Gerhold *et al.,* 2002; Speed, 2003) to perform diagnostics and prognostics (Speed, 2003), to profile complex diseases such as cancer (Speed, 2003), to provide a characteristic response of particular drug to a disease or treatment analysis (drugs discovery) (Speed, 2003), and to perform new gene discovery (Lander, 1999).

Microarrays are used to quantify mRNAs transcribed from different protein-encoding genes under different test (unknown) and reference (known) conditions. RNA is extracted from a cell or tissue sample, and

then converted to cDNA. Fluorescent dye tags, (usually Cy3 and Cy5) are enzymatically incorporated into the newly synthesized cDNA or can be chemically attached to the new strands of DNA (Cheung *et al.,* 1999; Dhammika and Javier, 2004). A cDNA molecule that contains a sequence complementary to one of the single-stranded probe sequences on the array hybridizes (complementary base pairing, A with T and C with G) to the spot at which the complementary reporters are affixed. The spot then fluoresces when examined using a microarray scanner. The fluorescence intensity of each spot is then evaluated in terms of the number of copies of a particular mRNA, which ideally indicates the level of expression of a particular gene (Duggan *et al.,* 1999; Churchill, 2002; Duyk, 2002).

DNA microarrays can be used to detect RNA's that may or may not be translated into active proteins (Duggan *et al.,* 1999). Scientists refer to this kind of analysis as *expression analysis* or *expression profiling*. Since there can be tens of thousands of distinct reporters on an array, each microarray experiment can accomplish the equivalent number of genetic tests in parallel. Arrays have therefore dramatically accelerated many types of investigations (Duggan *et al.,* 1999; Duyk, 2002). By giving information on the levels of gene expression of thousands of genes at the same time, DNA microarrays allow researchers for example, to relate the effect of a disease to particular genes and to find molecules that can be targeted for treatment with drugs among the various proteins encoded by disease-associated genes (Duggan *et al.,* 1999; Churchill 2002; Duyk, 2002). The applications are numerous, but the main challenge is to analyze the outcome and to use this massive databank to infer valuable biological information using mathematical and informatics tools.

12.1.2 *Interpreting the Scanned Image*

The end products of the microarray experiment are scanned array images. The intensities provided by array images can be quantified by measuring the average or integrated intensities of the spots. The ratio of fluorescent intensities for a spot is interpreted as the ratio of concentration for its corresponding mRNA in the two cell populations. For an array that has N_{array} distinct elements, the ratio (T) for i^{th} gene

(where i is an index running over all the arrayed genes from *1* to N_{array}) can be written as,

$$T_i = \frac{R_i}{G_i} \tag{12.1}$$

where G_i and R_i are the measured intensities of i-th array element at the excitation wavelengths of the two dyes. From this expression ratio it can be interpreted that if the R intensity is greater than G, then the spot will appear *red* and that gene is said to be *over-expressed* or *up-regulated*. If G intensity is greater than R, the spot will appear *green* and that gene is then *under-expressed* or *down regulated*. If both R and G are equal, then a *yellow* spot is obtained, which means that the gene is *equally expressed* in both samples. A *black* spot indicates that, at that position, *no hybridization* has occurred (Dhammika and Javier, 2004).

12.1.3 *Preprocessing of Microarray Data*

Interpreting the data from a microarray experiment is a challenging task. There are number of sources of systematic variations that affect the measured gene expression levels (Quackenbush, 2002). Such sources of systematic variation include:

- Differences in labeling efficiencies of the two dyes (cy3 and cy5).
- Differences in the power of the two lasers (for exiting the cy3 and cy5 dyes).
- Spatial biases in ratios across the surface of microarray.

Normalization is the term used to describe the process used for removing such variations. There exist a number of normalization methods (Quackenbush, 2001; 2002) but the simplest and commonly used method is total intensity based normalization. This assumes that equal quantities of RNA for the two samples are being used in the experiment. There are millions of individual RNA molecules in each sample, it is assumed that the average mass of each molecule is nearly the same and consequently the numbers of molecules are identical in both samples. Second assumption is that approximately same number of labeled molecules from each sample hybridized to the array and, therefore, the total hybridization intensities summed over all elements in

the arrays should be the same for each sample (Quackenbush, 2002). Using this approach, a normalization factor is calculated by summing the measured intensities in both channels (Quackenbush, 2002).

$$N_{total} = \frac{\sum_{i}^{N_{array}} R_i}{\sum_{i}^{N_{array}} G_i} \tag{12.2}$$

where G_i and R_i are the measured intensities for i^{th} array element and N_{array} is the total number of elements represented in the microarray. Now, one or both intensities can be appropriately scaled, for example, $G'_k = N_{total}G_k$ and $R'_k = R_k$. Thus, the normalized expression ratio becomes (Quackenbush, 2002),

$$T_i' = \frac{R_i'}{G_i'} = \frac{1}{N_{total}} \frac{R_i}{G_i} \tag{12.3}$$

This adjusts each ratio such that the mean ratio is equal to *1*. An equivalent process is to subtract a constant from the logarithm of the expression ratio (Quackenbush (2002)),

$$\log_2(T_i') = \log_2(T_i) - \log_2(N_{total}) \tag{12.4}$$

There are many variations on this particular type of normalization, including scaling the individual intensities so that the mean or median intensities are the same within a single array or across all arrays, or using a selected subset of the arrayed genes rather than the entire collection.

12.2 Gene Expression Profiling and Gene Network Analysis

The obtained preprocessed data is in the form of $n \times m$ matrix, where n represents number of genes (typically in thousands) and m represents experiments or time series (typically less than hundred). This data is analyzed efficiently so as to generate some useful biological information. The determined amount of up-regulation and under-regulation is sorted out using some computational algorithms. Genes are grouped according to their expression ratios, in form of clusters such that, within each

cluster, genes are co-regulated or similarly expressed but have different expression level when compared with the other clusters. This sort of information is used in understanding functional pathways and how genes and gene products (say proteins) interact with each other (Ingen, 2002; Speed, 2003; Dhammika and Javier, 2004).

12.2.1 *Gene Expression Profiling*

Gene expression profiling is a key initial step for detecting groups of genes that exhibit similar expression patterns. The data, gene expression matrix, to be clustered consists of elements (genes) and a characteristic vector (length of which depends on the number of observations considered) for each element. A measure of similarity is defined between pairs of such vectors. The goal is to partition the elements into subsets, which are called clusters, so that two criteria, first homogeneity, similarity of each gene in same cluster (i.e., small intra cluster distances), and second separation, low similarity between two clusters (i.e., large inter cluster distance) (Dhammika and Javier, 2004) are fulfilled. There exist a number of methods to sort the gene data into clusters. Clustering techniques can be classified into *divisive* or *agglomerative* (Speed, 2003). A divisive method begins with all genes in one cluster that is gradually broken down into smaller and smaller clusters (based on measure of dissimilarity). Agglomerative techniques usually start with single-gene clusters and gradually fuse them together (based on measure of similarity). Clustering can be either *supervised* or *unsupervised* (Quackenbush, 2001). Supervised methods use existing biological information about specific genes that are functionally related to guide the algorithm. On the other hand, unsupervised clustering algorithms do not need any covariate information directly in the analysis.

Unsupervised clustering algorithms (Speed, 2003; Dhammika and Javier, 2004) use some similarity measures based on Pearson's correlation coefficient or Euclidean distance between each gene to group them accordingly. One example is *hierarchical clustering* (Speed, 2003; Dhammika and Javier, 2004) in which the resultant classification has an increasing number of nested classes. *K-means clustering* (Dhammika and Javier, 2004) which is one of the widely used clustering techniques also

comes under this category. K-means clustering simply partition genes into different clusters without specifying the relation between the individual genes. The best advantage of unsupervised techniques is that they can be applied to get information of uncharacterized genes. But they have the disadvantage that final results are biased by the initial information (e.g., the number of clusters) provided and since no biological information is used, they can lead to serious misclassifications.

In supervised clustering, the clusters are predefined and the task is to understand the basis for the classification from a set of genes. One can use the profiles of known gene clusters by calculating a reference vector representative of the average log ratio data for each of the selected gene. In order to determine to which cluster individual gene can belong to, the vector calculated for each gene is scaled by a peak correlation value. The algorithm allows the integration of previous data compiled and analyzed by manual methods. But these have a disadvantage of miss-classification, as sometimes it forcefully clusters the new genes to defined clusters only. Some of the supervised clustering techniques are Self-Organizing Maps (SOMs) (Quackenbush, 2001), Support Vector Machines (SVMs) (Brown *et al.,* 2000).

12.2.2 *Gene Network Analysis*

Genetic interactions refer to the phenomena where the state of one gene is regulated by one or more genes. These interactions can have a positive or negative effect. There could be auto-regulation as well as feedback loops in these genetic networks. In feedback loops the output again interacts with system input and thus, changes the eventual output (Maine and Kimble, 1989; Packer *et al.,* 1998; Mathieu *et al.,* 2004). These interactions between genes are responsible for the observed expression of individual gene under given conditions (Mathieu *et al*., 2004). For example, analysis of the regulatory regions of the *Hox* genes has revealed that the genes as well as some gene products (metabolites, transcription factors, etc.) control the pattern of expression of other genes. Study confirms that normal expression of the genes requires both auto-regulatory and other interaction dependent modes of regulation (Packer

et al., 1998). The observed auto-regulation of the *Drosophila* deformed gene is conserved in a mouse homolog *in vivo*, and is reflected in a widespread requirement for positive feedback to maintain *Hox-a4* expression (Packer *et al.,* 1998). Interactions between Nodal/Activin and Fibroblast growth factor (Fgf) signaling pathways have long been thought to play an important role in mesoderm formation (Mathieu *et al.,* 2004). The *glp-1* gene functions in two inductive cellular interactions and in development of the embryonic hypodermis of *C. elegans* (Maine and Kimble, 1989). A variety of genome projects dealing with diverse prokaryotes and eukaryotes have been completed or are near to complete characterizations, such as *E. coli*, *M. tuberculosis*, and *C. elegans* (Blattner *et al.* 1997; Cole *et al.,* 1998; Collins *et al.,* 1998; Smolen *et al.,* 2000). Such data can help in obtaining a detailed understanding of how groups of genes control cellular responses to environmental stimuli (Edwards, 1994; Smolen *et al.,* 2000), and execute stored programs governing such biological processes as development (Rossant and Hopkins, 1992) or the cell cycle (Okayama *et al.,* 1996).

For achieving a proper understanding of the networks, only the collection of large experimental data sets is not sufficient. A proper framework is required to analyze this data. The various concepts and technicality of mathematical language makes mathematical modeling a useful framework for conceptualizing and understanding complex biological systems. Various existing mathematical models are used to analyze the behavior of genetic regulatory systems. Different qualitative properties and means of modeling generate these mathematical models. These mathematical models are often problem and domain dependent. The modeling is accomplished by taking the experimental knowledge about the natural system of interest and formulating a model to see where the model prediction matches experimental data and where the model prediction disagrees with the natural system. In recent past, many models have been introduced based on Boolean networks (Kauffman, 1969a; 1969b; Glass and Kauffman, 1973; Mestl *et al.,* 1996; Silvescu and Honavar, 2001; Thomas and Kaufman, 2001a; 2001b; Albert, 2004), differential equations (Jacob and Monod, 1961; Goodwin, 1965; Tyson and Othmer, 1978; Snoussi and Thomas, 1993; McAdams and Shapiro, 1995; Mestl *et al.,* 1996; Mestl *et al.,* 1997; Plahte *et al.,* 1998; Yuh

et al., 1998; Weaver *et al.,* 1999; Edwards, 2000; 2001; Edwards and Glass, 2000; Voit, 2000; Zill, 2000; Ando and Iba, 2001; Bower and Bolouri, 2001; Ben-Hur and Siegelman, 2004; Mason *et al.,* 2004), weights matrices (Weaver *et al.,* 1999; Ando and Iba, 2001), hybrid approaches (McAdams and Shapiro, 1995; Yuh *et al.,* 1998; Edwards, 2000; 2001; Edwards and Glass, 2000; Ben-Hur and Siegelman, 2004) and stochastic models (Keasling *et al.,* 1995; Gillespie, 1997; McAdams and Arkin, 1998; Kastner, 2002). Many of these formulations have been used for both forward and reverse–modeling problems (Smolen *et al.,* 2000). A forward modeling is a form of knowledge-driven model construction while reverse–modeling tries to use the behavior of the system itself to directly infer the interactions of the natural system.

12.2.2.1 Network Characteristics

One of the most basic feature of any complex network is its structure, thus, it is natural to investigate its connectivity. The structure of networks are often constrained and shaped by the growth processes i.e., evolution in the case of natural networks. The topology of biological networks might shed some light on the possible structures and dynamics which have been exploited by nature. Generally, these networks can be classified in two extreme kinds of networks, regular and random networks (Strogatz, 2001; Wang and Chen, 2003). Usually a network is characterized by a few parameters discussed next.

The most elementary characteristic of a node in a network is its *degree* (or connectivity), k, which tells us how many links the node has with other nodes. An undirected network with N nodes and L links is characterized by an average degree $\bar{k} = 2L/N$. The *degree distribution*, $P(k)$, gives the probability that a selected node has exactly k links. $P(k)$ is obtained by counting the number of nodes $N(k)$ with $k = 1, 2\ldots$ links and dividing by the total number of nodes N. The degree distribution allows us to distinguish between different classes of networks. A peaked degree distribution indicates that the system has a characteristic degree and that there are no highly connected nodes (which are also known as hubs). By contrast, a power-law degree distribution indicates that a few hubs hold together numerous small nodes.

The *characteristic path length L* is the measure of the typical separation between any two nodes in the network. The value of the characteristic average shortest path length L is equal to the number of edges in the shortest path connected by any two nodes, averaged over all pairs of nodes. Here, L determines the effective "size" of a network, the most typical separation of one pair of nodes therein.

The *clustering coefficient C* measures the cliquishness of a typical neighborhood. It is the average fraction of pairs of neighbors of a node which are also neighbors of each other. More precisely, one can define average clustering coefficient C as the average fraction of pairs of neighbors of a node that are also neighbors of each other. Suppose that a node i in the network has k_i edges and they connect this node to k_i other nodes. These nodes are all neighbors of node i. Clearly, at most $k_i (k_i - 1)/2$ edges can exist among the neighbors (considering no self edge), and this occurs when every neighbor of node i connected to every other neighbor of node i. The clustering coefficient C_i of node i is then defined as the ratio between the number E_i of edges that actually exist among these k_i nodes and the total possible number $k_i (k_i - 1)/2$, namely, $C_i = 2E_i/(k_i (k_i - 1))$. The clustering coefficient C of the whole network is the average of C_i over all i. Clearly, $C = 1$ if and only if the network is globally coupled, which means that every node in the network connects to every other node.

12.2.2.2 Regular Networks

The regular network consists of N nodes arranged regularly on a lattice such that each node has the same number, say M, of nearest neighbors. Regular networks have a highly clustered and large characteristic path length L, where L grows linearly with N in these networks. Most regular network possesses the property that if two nodes are connected to a common third node, then there is a high probability that the two nodes are connected between themselves, i.e., a high degree of clustering (Strogatz, 2001; Wang and Chen, 2003). However, in general it takes many steps to move between two arbitrary nodes in the network (de Moura *et al.,* 2003). A high degree of clustering and a large value for the average shortest path are thus the two defining properties of most locally

connected regular networks (Strogatz, 2001; de Moura *et al.,* 2003; Wang and Chen, 2003). On contrast, there exist random networks which exhibit small average path length with low clustering coefficient (Wang and Chen, 2003). The degree distribution of the regular network is constant, as each node has same number of neighbors or links.

12.2.2.3 Random Networks

The random networks consist of *N* nodes with *(NZ)/2* edges that connect *N* nodes randomly. Here, *Z* is the average number of neighbors to all nodes in the network. Random networks have small clustering coefficient as well as small average path length, where *L* grows with *Log (N)*. Large values of *C* are usually associated with large values of *L* and *vice versa.* Due to the sparse and random connections, such networks have extremely low degree of clustering and small average shortest path (Milgram, 1967; Newman *et al.,* 2000; de Moura *et al.,* 2003; Barabasi and Oltvai, 2004). The degree distribution follows a Poisson distribution, which indicates that most of the nodes have approximately the same number of links (close to average degree <*k*>). The tail (high *k* region) of the degree distribution *P(k)* decreases exponentially, which indicates that nodes that significantly deviate from the average are extremely rare.

12.2.2.4 Small World Networks

The majority of mathematical models to elucidate dynamical systems were described by completely regular or completely random networks. These are idealized mathematical extremes and are rarely observed in natural networks. The real networks are somewhere in-between these two kinds of networks. The degree distribution of small-world network is similar to random network in that it follows a Poisson distribution. Its peak value occurs at the average degree of the network and decays exponentially for large values of degrees. These real networks have characteristic high clustering coefficients and low average path lengths. The idea for such networks originated from the theory of "*small world phenomena*" (Milgram, 1967) and these in-between networks can be modeled mathematically, using the small world theory or phenomena

proposed by Watts and Strogatz (Watts and Strogatz, 1998). The small world network was found in many large real world networks in different field of science such as electrical power grid analysis, religion and economic growth networks, disease transmission, graph theory, corporate communication, gene network and computer circuitry (Milgram, 1967; Watts and Strogatz, 1998; Newman *et al.,* 2000; de Moura *et al.,* 2003).

12.2.2.5 Indirect Interactions and Time Delays

In gene networks, besides direct interactions of genes, there are indirect interactions which play a vital role in regulating the pathway. When considering a gene regulation, that is to know how one gene affects the activity of the other genes, the question often arises as to whether this is caused by direct or indirect interactions. For example, when over expressing a transcription factor *X*, the gene expression levels of genes *A* and *B* changes. *X* binds the upstream regulatory region of *A* and regulates its expression. This is called as direct effect of *X* on *A*. However, in case of *B*, *X* includes the regulation of gene *A* expression which indeed inactivates/activates (regulates) a transcriptional repressor of *B*. This is generally called as indirect effect of *X* on *B*. There are very few models which are considering these indirect effects in analyzing gene regulatory networks. Wagner (Wagner, 2001) proposed a model in which reconstruction of gene regulatory network is done based on gene perturbations. Its core task is to identify casual structure of a gene network, that is, to distinguish direct and indirect regulatory interactions among gene products.

Even though a lot of progress has been made, key biological features such as time delay have been left largely unaddressed in the context of inferring regulatory networks. From a biological viewpoint, time delays in gene regulation arises from the delays characterizing the various underlying processes such as transcription, translation and transport processes. Experimentally, measured time delay in gene expression has been widely reported in literature (Friedman *et al.,* 1998; Kim *et al.,* 2003; Perrin *et al.,* 2003; Dasika, 2004; Zou and Conzen, 2004; Agrawal and Mittal, 2005; Wu *et al.,* 2005).

12.2.3 *Need for Newer Techniques?*

It is well known now that identifying co-expressed genes in form of clusters from microarray data is an important and challenging task for mathematicians as well as biologists. Most of the existing clustering techniques contain some drawbacks, like, in supervised clustering some information of profiles of known genes has to be provided, while in unsupervised clustering, final results depend on initial decision taken which can result in miss-classifications. Similarly SOMs deals with too many parameters and also a specific *a priori* grid structure has to be provided, while in SVMs and other techniques using a training set, the training set might not represent the actual sample of the whole gene population. Even with the developments of new clustering algorithms, it is still not easy to cluster gene expression data. The structure of gene expression data is often complicated. Different clustering algorithms, or even the same clustering algorithm with different parameter values, can generate very different clustering results. In other words, there is no general guideline to choose appropriate parameter values. Thus, the development of a robust and efficient clustering algorithm, which can be used for gene expression profiling, addressing disadvantages of the existing clustering techniques, is definitely needed.

It is also emphasized that the choice of model structure for representing genetic regulatory networks is problem domain dependent. Although Boolean network models have been used in modeling gene regulation, there is some debate on their utility due to the assumptions inherent in these models. Assumptions include synchronous updating, all–or–nothing gene transcription, and restrictions on network connectivity. Differential equation models eliminate many of the limitations of Boolean models but add increased levels of complexity necessitating the selection of parameter values, the use of complex analysis techniques and higher CPU times. In turn, stochastic models purport to model more accurately the genetic regulation by removing the deterministic assumptions inherent in differential equation models albeit with an additional penalty of computational complexity and analysis.

Many more formulations exist and are in current use for modeling genetic regulatory networks. These are not discussed here for the sake of

brevity. Some examples are the use of neural networks (Mestl *et al*., 1996) and artificial regulatory networks (ARN) (Banzhaf and Kuo., 2004). Thus, developing a technique which is both robust and economical in sense of mathematical calculation and CPU time is always a subject of interest and development. The technique should be able to address complexity, nature of response and function of individual gene or cluster in the genetic networks. Generally, the structure of network reveals the nature of genetic interactions, and it is quite important to study the topological properties of networks. This helps in getting a better understanding of biological networks.

12.3 Role of Multi-Objective Optimization

Generally, two or more objective functions are defined for gene expression profiling and gene network analysis. Usually, these objectives are conflicting in nature. Use of traditional single objective optimization techniques to solve these multi-objective optimization problems suffer from many drawbacks. Single objective problems either use penalty function approach or use some of the objectives as constraints. Both of these approaches have user-defined biases. Thus, multi-objective optimization techniques are definitely needed to model and solve these and similar other problems.

12.3.1 *Model for Gene Expression Profiling*

The variables, parameters, mathematical models, and the genetic algorithm (GA) approach to model the gene expression profiling are discussed in this section. The various variables are discussed first.

x_{jk} = expression ratio of jth gene in k^{th} experiment (one entry of the gene expression matrix (n×m) from cDNA microarray experiments)

y_{ik} = expression ratio of ith cluster in k^{th} experiment (cluster locations calculated/predicted by GA)

z_{ik} = expression ratio of ith cluster in k^{th} experiment (cluster locations based on input gene expression data and specific association rule at each iteration/generation)

n_c = optimal number of clusters present.
d_{ji} = distance matrix, each element of this represents distance of jth gene from i^{th} cluster.
a_{ij} = 1 if j^{th} gene belongs to i^{th} cluster
= 0 if j^{th} gene does not belong to i^{th} cluster

where $i = 1,..., n_c$ (where n_c is the number of clusters), $j = 1,\ldots, n$ (where n is the number of genes), and $k = 1,\ldots, m$ (where m is the number of experiments). Note that a_{ij} are not optimization variables in GA models. They are defined to implement logical constraints in the models. These are dependent on d_{ji} values. Here, the optimization variables are the number of clusters, n, and the expression ratio of i-th cluster in k-th experiment, y_{ik}, for $i = 1, \ldots, n_c$ and $k = 1, \ldots, m$. Using these variables, various objective functions for clustering gene expression data are defined.

An objective function is proposed where the error between z_{ik} and y_{ik} is minimized. y_{ik} are randomly generated in GA while z_{ik} are calculated using gene expression data and gene associations (described in detail in the next section (model implementation)). Ideally, y_{ik} should be equal to z_{ik} to obtain inherent clusters present in the dataset. Minimizing this difference, indirectly results in optimum number of clusters. Mathematically, it is represented as,

$$\mathrm{I}_1 = \min \sum_{i=1}^{n_c} \sum_{k=1}^{m} (z_{ik} - y_{ik})^2 \tag{12.5}$$

This may result in each gene acting as an independent cluster to give zero error. Thus, another objective function is defined that minimizes number of clusters in gene expression data. This is especially useful for diagnostic purposes where lesser numbers of gene clusters are useful for quick diagnostics. Mathematically, it is represented as,

$$\mathrm{I}_2 = \min n_c \tag{12.6}$$

Another objective is proposed in terms of distance between each cluster. This objective function aims to maximize the distance between the clusters. This is an important requirement to have good generalizations for newer genes. Mathematically, it is represented as

$$I_3 = \max \sum_{j=i+1}^{n_c} \sum_{i=1}^{n_c} \sum_{k=1}^{m} (y_{ik} - y_{jk})^2 \tag{12.7}$$

A multi-objective model is proposed by considering these three objective functions. The model looks for a solution which simultaneously minimizes the number of clusters, maximizes the distance between clusters while reducing the sum of squared errors between y_{ik} and z_{ik} . Mathematically, it is represented by Eqs. (12.5)-(12.7) with the following constraints.

$$y_{ik}^{l} \le y_{ik} \le y_{ik}^{u}$$

$$1 \le n_c \le n_{c\max}$$

This model for gene expression profiling can be implemented using *evolutionary algorithms*. In this work, a genetic algorithm (GA) based approach from open literature, namely, *Non-dominated Sorting Genetic Algorithm-II (NSGA-II)* (Deb *et al.,* 2002) is employed.

12.3.2 *Implementation Details*

Various steps involved in model implementation (Sikarwar, 2005) are listed below:

Step 1: GA randomly generates the number of clusters, n_c, and their locations, y_{ik}.

Step 2: A distance matrix d_{ji} (dimension $n \times m$) is calculated; this represents the distance of jth gene from i^{th} cluster. It is the Euclidean distance between the gene and the cluster center. Mathematically, it is given by,

$$d_{ji} = \sqrt{\sum_{k=1}^{m} \left(x_{jk} - x_{ik}\right)^2} \tag{12.8}$$

Step 3: *for SCA (Single Cluster Assignment – One gene can belong to one cluster only, signifying only one function)*: Distance matrix d_{ji} is sorted for minimum distance along each row, i.e. for each gene, and the gene is assigned to the cluster (assigning a_{ij} equal to 1) for which the Euclidean's distance is minimum.

for MCA (Multiple Cluster Assignment – One gene can belong to more than one cluster, signifying more than one function): Distance matrix d_{ji} is sorted for minimum distance along each row and an upper range of association distance is obtained by taking a factor (= 1.2, this is obtained by considering reported errors in microarray experiment in the range of 10% to 30%, a mean value of 20% is taken) *times* the minimum distance for each row (i.e., for each gene). Gene is then assigned (by defining a_{ij} equal to 1) to all those clusters for which it's Euclidean's distance lies in the range of maximum association distance. It is to be noted that this upper range on distance is different for each cluster.

Step 4: z_{ik} is calculated for each a_{ij} equal to 1, i.e., for each gene that belongs to that cluster, representing the average of all the associated genes' expression values for each experiment, e.g., for cluster 1, z_{1k} is equal to average expression ratios of all genes for which a_{1j} =1. Mathematically,

$$z_{ik} = Avg(x_{jk}) \text{ For each } a_{ij} = 1. \tag{12.9}$$

The obtained value of z_{ik} is then used in GA for objective function evaluation.

Step 5: Minimize or Maximize selected objective function(s)

12.3.3 *Seed Population based NSGA-II*

In this work, a seed-based NSGA-II is used (Sikarwar, 2005). It results in reduction of the required population and generations to converge at the optimal solutions. For this, a small fraction of population is supplied to the GA, as seed population, during initialization. This does not affect the diversity of the GA population, but provides 'good' initial chromosomes. Moreover, if these happen to be 'bad' initial chromosomes, they will die out as a result of GA operators in a few subsequent generations. This is inspired by some of the previous works on gene expression profiling

(Garg, 2002; Garg *et al.,* 2003; Guda *et al.,* 2003). Empirical parameters based on the analysis of variance of the given gene expression data matrix were used in these studies and no optimization was performed (Garg *et al.,* 2003).

A simple distance based clustering on the gene expression data is used to generate seeded population. A distance matrix d_{ab} is defined, which represents Euclidean's distance between each gene. It is calculated as,

$$d_{ab} = \sqrt{\sum_{k=1}^{m} \left(x_{ak} - x_{bk}\right)^2} \ , \ \forall\, a = 1, \ldots, n \text{ and } b = 1, \ldots, n, a \neq b \tag{12.10}$$

The overall minimum and maximum distances are found. Based on these, each element of the d_{ab} matrix is mapped between 1 and 0 and represented by d_{ij}, by using linear mapping as shown below:

$$d_{ij} = \frac{\left(d_{ab} - d_{\min}\right)}{\left(d_{\max} - d_{\min}\right)}, i \neq j \tag{12.11}$$

where $i = 1, \ldots, n, j = 1, \ldots, n$ and d_{min} and d_{max} are overall minimum and maximum distances, respectively. A normalized distance matrix is thus obtained. Distance of each gene is compared with the other gene. If the distance is less than a multiple of the average of d_{min} and d_{max}, they are assigned to a single cluster (and a_{ij} is set to 1). The process continues till all the genes are clustered. The average expression ratio of each cluster is calculated, on the basis of association information. These calculated expression ratios are used as seeds in GA. A mixed population is generated for different multiple values and used in GA.

12.3.4 *Model for Gene Network Analysis*

It is envisaged that most complex naturally occurring networks are driven by small world networks. Thus, the model developed and implemented in this work is inspired from Boolean networks (Thomas and Kauffman, 2001a; 2001b; Silvescu and Honavar, 2001; Albert, 2004), weight matrices (Weaver *et al.,* 1999 and Ando and Iba, 2001) and small world phenomenon (Milgram, 1967: Watts and Strogatz, 1998;

Barrat and Weigt, 2000; Newman, 2000; Kleinberg, 2000; Strogatz, 2001, Barabasi *et al.,* 2002; de Moura *et al.,* 2003; Wang and Chen, 2003; Barabasi and Oltvai, 2004; Elettreby, 2005). In the proposed model the small world networks are reverse engineered using the Pearson correlation coefficients and the clustered gene expression data (Verma, 2006). Correlation coefficients are used to define the neighborhood for a given cluster. All clusters in the neighborhood of a given cluster are assumed to be interacting with the given cluster. Random interactions are further added based on random probability values. Shifted correlation coefficients are used for adding time delay links. Weights for all these directed interactions are calculated by minimizing the sum of squared error between model predicted values and experimental cluster data from clustering.

The model ensures a set of simple non-dominated networks (Pareto optimal) while minimizing the numbers of indirect links in the networks. The total number of indirect links is minimized to keep the network sparse (i.e., to keep the network simple). The model also maximizes the percentage difference between clustering coefficients to maintain one of the small world network (SWN) characteristic and minimizes the sum of squared errors to fit the experimental data. One of the constraints is included to ensure the minimum connectivity in all non dominated networks. The other constraint is included to make the *L(avgshortestpath)* nearly equal to $log_{(k-1)}r$ so that SWN characteristic can be ensured. Mathematically, the model (Ganta, 2007) is:

$I_1 = \text{Min } (idl)$

$$I_2 = Min \left[-\left[\frac{(avgcc - crandom)}{avgcc} \times 100 \right] \right]$$

$$I_3 = \text{Min} \sqrt{\left[\sum_{l=1}^{n_c} \sum_{k=1}^{m-1} \left(\frac{y_{\text{exp},l}^{k} - y_{\text{mod},l}^{k}}{y_{\text{exp},l}^{k}} \right)^2 \right]}$$

Subject to

$lpn \geq Ln\ (r) \quad \forall$ nodes

$constant_1 \geq (avgshortestpath - log_{(k-1)} r)^2$ where k is the degree of the node

$constant_2 \geq (dl)$

$$y_{\mathrm{mod},l}^{k+1} = \sum_{i=1}^{n_c} W_{il}\, y_{\mathrm{exp},i}^{k} + \sum_{i=1}^{n_c} \hat{W}_{il}\, y_{\mathrm{exp},i}^{k-1}$$

where, W_{ij} are weights between i^{th} *and* j^{th} cluster with no time delay and $\hat{W}_{ij}$ are weights between i^{th} *and* j^{th} cluster with time delay.

12.3.4.1 Algorithm

Step 1: Generate the Pearson correlation matrix and the squared Pearson correlation matrix to account for both positive and negative interactions from cluster data matrix ($DS_{r\times m}$, the average expression profiles of the clusters obtained from NSGA-II implementation of clustering model). Initialize *q, p,* and *weights*. The integer variable *q* is bounded by the relationship $ln(r) < (r-q) < r$ which ensures minimum connectivity. Real variable *p* is between 0 and 1 and *weights* are between -1 and 1.

Step 2: Based on squared Pearson correlations define the neighborhood of each node. Generate some links (say, $k = r-q$) per node with highly correlated nodes in the neighborhood to form a population of directed networks, where $1 < Ln(r) < k < r$ (Watts and Strogatz (1998)).

Step 3: Insert random links to the randomly generated networks with random probability (*p*) such that a total of "$p\times r\times q$" links are further added.

Step 4: Insert time delay links to the network based on shifted Pearson correlation matrices, with time delay as $(t-\tau\Delta t)$ ranging from $\tau = (0,...., \tau_{max})$. In the implementation of the model, only, $\tau = 1$ is considered.

Step 5: For the updated population of directed networks with random links and time delay links, calculate the characteristic parameters i.e. average clustering coefficient (*avgcc*) and average shortest path length (*avgshortestpath*). Calculate similar parameters, namely, *Crandom (Cran)* and *Lrandom (Lran),* for the equivalent random networks i.e. random networks with same number of nodes and links (Newman *et al* (2000)).

Step 6: Optimize the values of *q*, *p*, and *weights* on all existing links ensuring SWN's properties i.e. *avgcc*>>*Crandom* and *avgshortestpath*≈*Lrandom* (Newman *et al* (2000)) while minimizing the number of indirect links and minimizing the total error between model predicted values and the average cluster expressions at each experimental attributes by NSGA-II to generate a set of non-dominated directed networks. A flowchart for implementation details is given in Fig. 12.1.

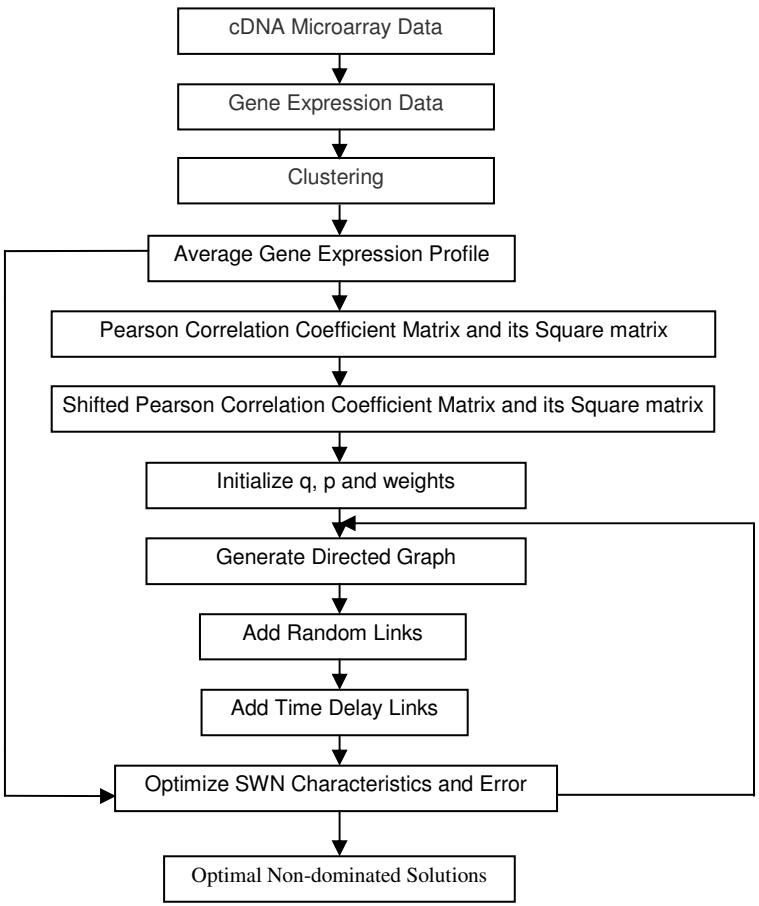


Fig. 12.1 A schematic flowchart of the overall algorithm

12.4 Results and Discussion

A synthetic gene expression dataset (DS1) was generated and subsequently used for clustering its members (which can be considered as genes) into subgroups. The dataset was prepared to test the performance of the proposed model to cluster genes into subgroups. DS1 contains expressions of 45 members across 10 experimental attributes. These members belong to 9 clusters with five members in each of them. Fig. 12.2 shows the average expression profile of the 9 clusters. Multi Cluster Assignment (MCA) was used with seed based initial population. The GA was simulated for 6000 generations with 1000 chromosomes in the initial population out of which 50 were seeds. Fig. 12.3(a) and Fig. 12.3(b) show the results obtained. Comparing Fig. 12.3(a) with the reference plot (Fig. 12.2) shows that the model has successfully captured all the trends present in the DS1. The gene associations are also observed to be true and no false positives or negatives are observed. This shows that the algorithm applied to this dataset results in a zero error. Values of various parameters obtained by sensitivity analysis are listed in Table 12.1 (supplementary information on CD). The multiplication factor (refer to Step 3 of the clustering algorithm) is taken to be 1.2 for maximum association distance calculation. To see the effect of this multiplication parameter runs with 1.1 and 1.3 (accounting for 10 and 30% error in microarray experiment) were also carried out. Results with 1.1 and 1.3 multiplication factors are given in the supplementary information on CD.

The overall objective function value [a weighted summation of the objectives used in the code, mathematically equal to $\{I1+I2+1/(1+I3)\}/3$] obtained for best the parameter set was 3.32 while in these runs it is observed to be 3.362 and 3.33. It is also noted that the same number of clusters are observed for small changes in the association distance. Thus, the clustering is robust to small changes in association distance values. This also shows the effect of multiplication factor (or association distance) on the overall function value.

A set of Pareto optimal fronts are obtained. In Fig. 12.3(b) for a given value of *I2*, if *I1* value increases the values of *1/(1+I3)* decreases. The solution (Fig. 12.3(a)) is reported for the point with minimum value of

$\{I1+I2+1/(1+I3)\}/3$. Different users may have different requirements and, thus, a different point from the Pareto set can be used.

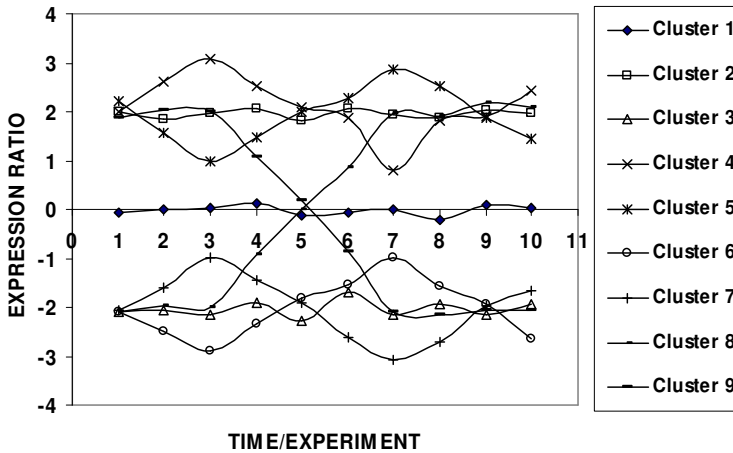


Fig. 12.2 Expression profiles of 9 clusters present in synthetic dataset (DS1)

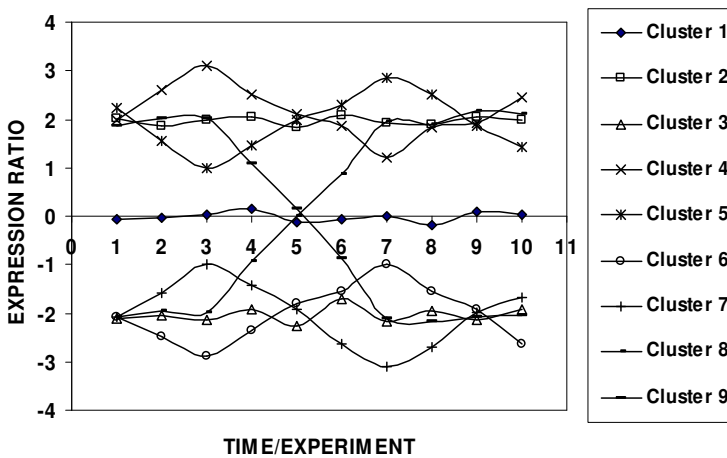


Fig. 12.3(a) Converged expression profiles for factor = 1.2

For example, for diagnostics one may be interested in a minimum number of clusters (corresponding to minimization of I_2) or for better generalization the inter cluster distances should be maximum

(corresponding to maximization of I_3). Thus, the Pareto set represents a compromise between such objectives. The network results are not reported here for DS1 as this was a synthetic dataset generated to prove the viability of the gene expression profiling algorithm. Note that the network generated has no biological meaning. Nonetheless, the network for DS1 is included in the supplementary information file in the folder: Chapter 12 on the attached CD.

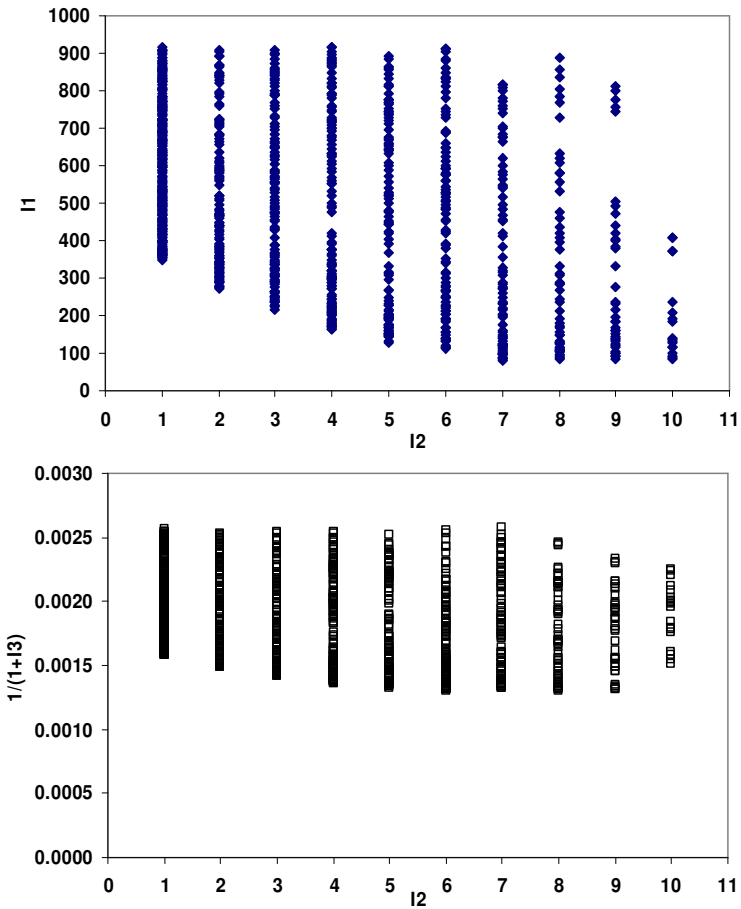


Fig. 12.3(b) Pareto optimal solutions

After getting good clustering results for synthetic dataset, DS1, a real life gene expression dataset, DS2 (Iyer *et al.,* 1999) is used. This microarray dataset contains gene expression profiles of 517 genes across 13 experiments. The dataset represents the effect of serum on human fibroblast cells. Iyer *et al.* (1999) reported 10 clusters for this dataset. The results obtained using the GA model on fibroblast-serum data are shown in Figs. 12.4(a) and 12.4(b). The results are reported for the best parameter values (given in Table 12.2, supplementary information on CD), after sensitivity analysis. Here, 13 clusters are obtained with three clusters with one gene association only. The cluster cardinalities are tabulated in Table 12.3 (supplementary information on CD).

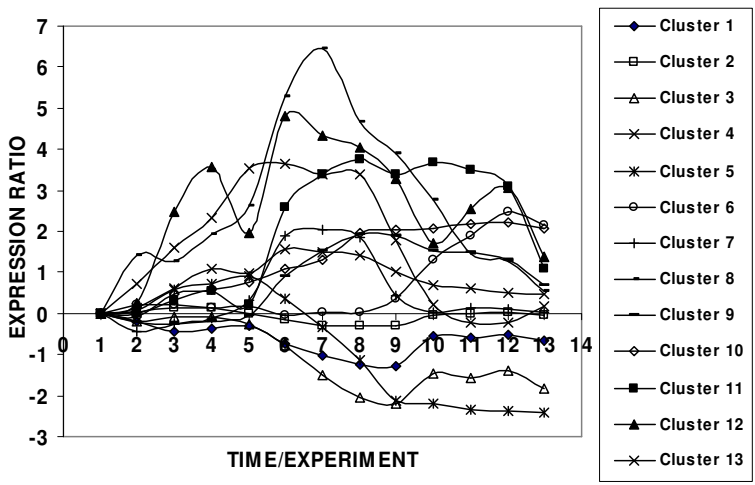


Fig. 12.4(a) Converged expression profiles

Here, all 517 genes are reported in 13 clusters. In comparison with previously reported results, it is observed that the average expression profiles of the clusters are similar to the previously reported average expression profile of the clusters. On the other hand, if one adds the cardinalities of previously reported clusters (Iyer *et al.,* 1999), it comes out to be 462. Thus, there are 55 missing genes in previously reported 10 clusters (A-J, Iyer *et al.,* 1999). However, in the present case, the number of clusters can be reduced from 13 by using correlation coefficients

between clusters and merging highly correlated clusters with low cardinalities to other clusters with high cardinalities, e.g., cluster-8, cluster-12 and cluster-13 with one gene each can be merged with cluster-4 with 64 genes because the correlation coefficients between these clusters and cluster-4 are 0.903, 0.914 and 0.859, respectively.

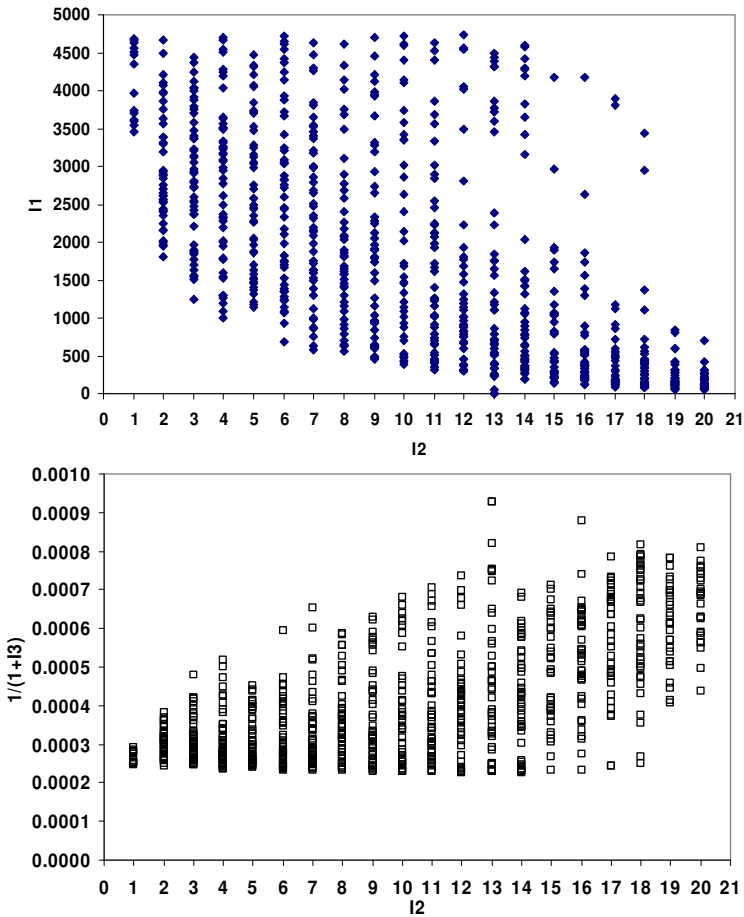


Fig. 12.4(b) Pareto optimal solutions

Pearson's correlation coefficient can in general be used for the purpose (Duggan *et al.,* 1999). If a cut-off criteria of 0.9 (-0.9 represents negative correlation while 0.9 represents positive correlation) is taken for highly

correlated clusters, then, based on this criterion, highly correlated clusters are obtained as highlighted in Table 12.4 (supplementary information on CD). The results are reported for the best parameter values (given in Table 12.2), after sensitivity analysis and, *if needed,* these can be merged to get a lower number of clusters.

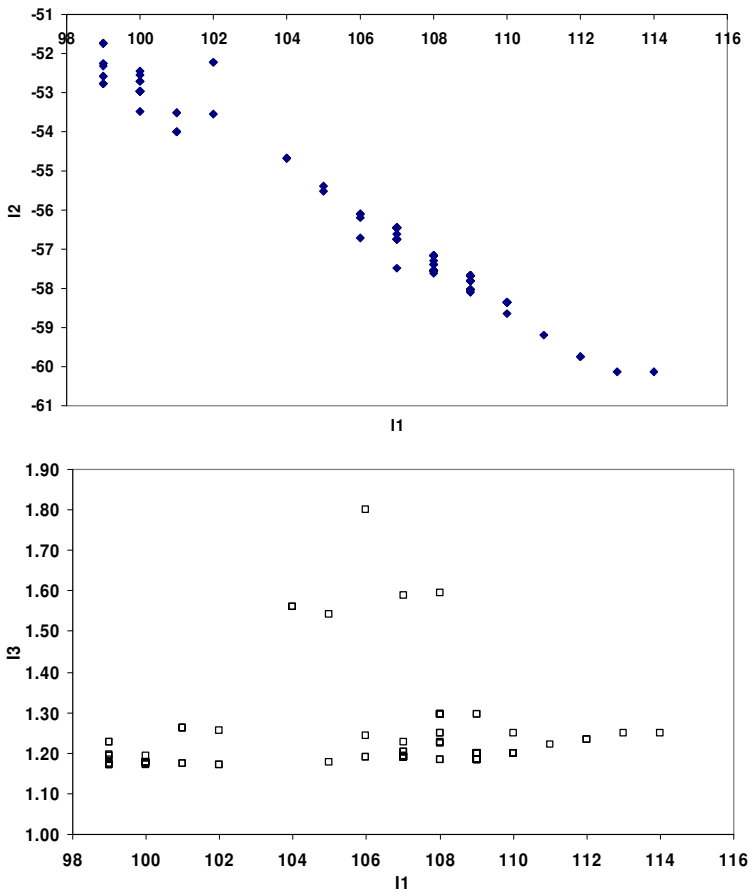


Fig. 12.5(a) Non-dominated Pareto solutions

For reverse engineering the gene networks, the initial neighborhood is defined based on Pearson correlation coefficients. The weights, signifying gene regulations, are randomly generated for a value of $constant_1$ and $constant_2$ in model constraints to be 0.0001 and 70, respectively. A small *error* range (1-1.9) as seen from Fig. 12.5(a) is observed. It is also observed that *(C-Cran)*100/C* range is 51-61, and the number of indirect links range is 98-114. The non-dominating Pareto optimal solutions obtained are shown in the Fig. 12.6(a).

Some of the non-dominated solutions for different values of indirect links are shown in Table 12.5 (supplementary information on CD). Each solution represents a unique network topology. The solution closest to the origin (corresponding to the minimum value of $\{I1+I2+I3\}/3$) obtained from the Pareto-optimal set is enclosed in a box in Fig. 12.6(a). The optimal input parameters used for this run are listed in Table 12.6 (supplementary information on CD).

The graphical representation of the network for the best non-dominated optimal solution is presented in Fig. 12.5(b) and the corresponding scatter plot of weights for the interactions present generated by the mathematical model is shown in Fig. 12.5(c).

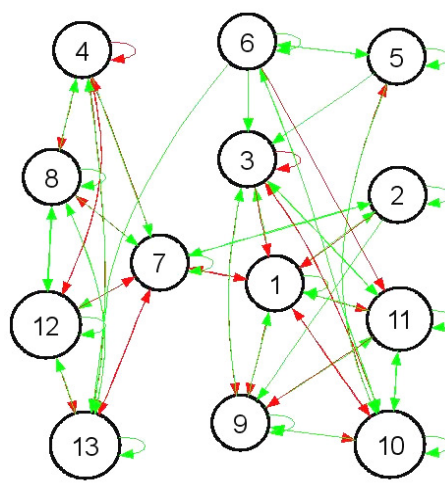


Fig. 12.5(b) Graphical Representation of Gene Network

In Fig. 12.5(b) the number inside the circles denotes the clusters, the red edges denotes negative interactions while the green edges denotes positive interactions. It is observed that no time delay link is present in the optimal model. The optimal network is also observed to be sparse as most of the weights are observed to be zero valued (Fig. 12.5(b)). Two highly clustered regions connected by a few nodes (acting as hubs) are observed in Fig. 12.5(b), signifying small world network topology. This also highlights the redundancies present in the biological network signifying that even if some links are removed, the network topology is preserved. The degree distribution obtained (supplementary information on CD) signifies that the obtained network is small world exponential network following the Poisson degree distribution. The fitness of the optimal model is checked through the plot of model predicted gene expression ratio versus experimental gene expression ratios (normalized parity plot) as shown in Fig. 12.5(d). It can be seen that there is excellent agreement between models predicted values and experimental data. This shows that the model is able to learn and depict the correct behavior of the biological data.

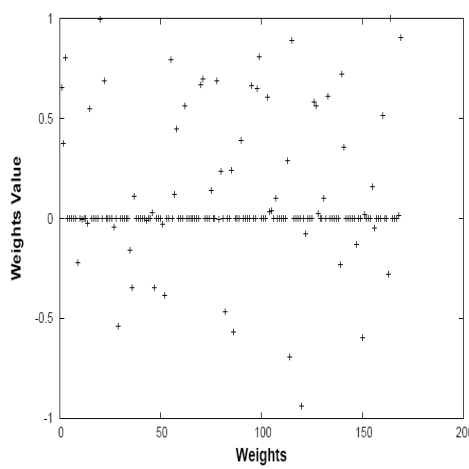


Fig. 12.5(c) Scatter Plot of Weights for a Gene Network

Also, it is observed that the errors are randomly distributed (supplementary information on CD) signifying no systematic errors in

the network model. From Fig. 12.5(b), it is observed that there are sub-groups of clusters acting together. These are (a) 3, 5, 6 (cholesterol biosynthesis, cell cycle and proliferation); (b) 9, 10, 11 (coagulation and hemostasis, signal transduction cytoskeleton reorganization, unidentified role in wound healing and tissue remodeling); (c) 4, 7, 8 (immediate-early transcription factor, signal transduction, tissue remodeling, re-epithelialization, coagulation and hemostasis and angiogenesis); (d) 7, 8, 12 (re-epithelialization, angiogenesis); and (e) 7, 12, 13 (re-epithelialization, coagulation and hemostasis, angiogenesis, and inflammation).

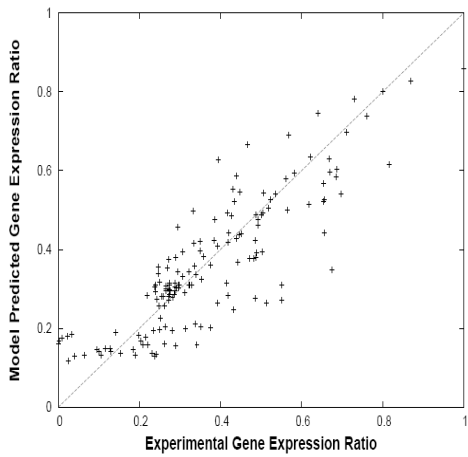


Fig. 12.5(d) Normalized Model Predicted Gene Expression Ratio vs. Normalized Experimental Gene Expression Ratio

Clusters 1, 7 and 10 have more outgoing links compared to other clusters. Cluster 1 is having seven incoming and seven outgoing links with predominately genes encoding for tissue remodeling. Cluster 7 is forming a hub between two highly clustered gene groups. Genes of Cluster 7 are involved mainly with angiogenesis. Cluster 7 is highly correlated with clusters 2 and 4, with seven outgoing interactions and seven incoming interactions signifying a more complex role for these genes in the wound healing process. Cluster 2 has genes involved predominantly in cell cycle and proliferation while cluster 4 has genes

involved with signal transduction, immediate/early transcription factors. Cluster 10 has also seven incoming and outgoing links, with genes involved mainly with unidentified role in wound healing (Iyer *et al.,* 1999). Cluster 10 has links with Clusters 9 and 11 as shown in Fig. 12.6(b). These clusters have genes with role in coagulation and hemostasis as well as tissue remodeling. Thus, newer functionalities are observed for previously unidentified role in wound healing genes which need to be verified experimentally. All other clusters have 4 to 5 outgoing and incoming links signifying 4 to 5 different roles for each gene group (Iyer *et al.,* 1999).

Note that all algorithms were first implemented in C (results given here) and then in JAVA. The codes in JAVA, which are platform independent, are included in the supplementary information on CD and thus, the results may vary slightly from the ones presented here.

12.5 Conclusions

Development and implementation of a robust clustering algorithm and a reverse engineering graph-theoretic model for gene networks are reported. The clustering results on synthetic dataset as well as on a real life dataset are observed to be very encouraging. The clustering results of the synthetic dataset establish the viability of the proposed algorithm. For the real life dataset, the clusters obtained from the proposed clustering algorithm are used for reverse engineering the gene regulatory networks using the graph-theoretic model inspired by 'small world phenomena'. A set of Pareto optimal models, each having a different network topology, for real life dataset is observed. Out of this set, the network topology for the point closest to the origin is reported. The network model generated concurs with the available biological information. Newer functionalities and interactions are also proposed that concur with the observed cDNA microarray data. The effects of indirect interactions and time delays are also considered in the proposed model. It is observed that the network topology is significantly altered by indirect links but is not affected much by inclusion of time delays in model formulation for the dataset employed in this study. Thus, it is observed that multi-objective

optimization models can be proposed and implemented for complex real life problems which hithertofore were very difficult to solve.

Acknowledgments

This work was partially supported by a research grant by the Department of Science and Technology, Government of India to Sanjeev Garg.

References

Agrawal A. and Mittal A. (2005). A dynamic time-lagged correlation based method to learn multi-time delay gene networks. *Transactions on Engineering, Computing and Technology.* 9, pp 167-174.

Albert R. (2004). Boolean modeling of genetic regulatory networks. *Nature.* 650, pp 459–481.

Ando S. and Iba H. (2001). Inference of gene regulatory model by genetic algorithms. *Proceedings of IEEE Congress on Evolutionary Computation.* pp 712-719.

Banzhaf W. and Kuo P. (2004). Network motifs in artificial and natural transcriptional regulatory network. *Journol of Biological Physics and Chemistry.* 4, pp 85-92.

Barabasi A. L., Jeong H., Neda Z., Ravasz E., Schubert A. and Vicsek T. (2002). Evolution of the social network of scientific collaborations. *Physica A.* 311, pp 590–614.

Barabasi A. L. and Oltvai Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics.* 5, pp 101-113.

Barrat A. and Weigt M. (2000). On the properties of small-world network models. *The European Physics Journal B.* 13, pp 547-560.

Ben-Hur A. and Siegelman H. (2004). Computation in gene networks. *Chaos.* 14, pp 145–151.

Blattner F. R., Plunket-III G., Bloch C. A., Perna N. T., Burland V., Riley M., Collado-Vides J., Glasner J. D., Rode C. K., Mayhew G. F., Gregor J., Davis N. W., Kirkpatrick H. A., Goeden M. A., Rose D.J., Mau B. and Shao Y. (1997). The complete genome sequence of Escherichia coli K-12. *Science.* 277, pp 1453–1462.

Bower J. and Bolouri H. (2001). Computational modeling of genetic and biochemical networks. *MIT Press, Cambridge, MA, USA*.

Brown P. O. and Botstein D. (1999). Exploring new world of genome with DNA microarrays. *Nature Genetics Supplement*. 21, pp 33-37.

Brown M. P. S., Grundy W. N., Lin D., Cristianini N., Sugnet C. W., Furey T. S., Jr. Ares M., and Haussler D. (2000). Knowledge-based analysis of microarray gene expression data by support vector machines. *Proc. Natl. Acad. Sci. USA.* 97, pp 261-267.

Cheung V. G., Morley M., Aguilar F., Massimi A., Kucherlapati R. and Childs G. (1999). Making and reading microarrays. *Nature Genetics Supplement*. 21, pp 15-19.

Christian J., Stoeckert Jr., Causton H. C. and Ball C. A. (2002). Microarray databases: standards and ontologies. *Nature Genetics Supplement*. 32, pp 469 – 473.

Churchill G. A. (2002). Fundamentals for experimental design for cDNA microarrays. *Nature Genetics Supplement.* 32, pp 490-495.

Cole S. T., Brosch R., Parkhill J., Gernier T., Churcher C., Harris D., Gordon S. V., Eiglmeir K., Gas S., Barry C. E., Tekaia F., Badcock K., Basham D., Brown D., Chillingworth T., Connor R., Davies R., Devlin K., Feltwell T., Gentles S., Hamlin N., Holroyd S., Hornsby T., Jagels K. and Barrell B. G. (1998). Deciphering the biology of *Mycobacterium tuberculosis* from the complete genome sequence. *Nature.* 393, pp 537–544.

Collins F. S., Patrinos A., Jordan E., Chakravati A., Gesteland R. and Walters L. (1998). New goals for the U.S. human genome project. *Science.* 282, pp 682–689.

Crick F. (1970). Central dogma of molecular biology, *Nature*. 227, pp. 561-563.

Dasika M. S., Gupta A., Maranas C. D. and Varner J. D. (2004). Mixed integer linear programming (MILP) framework for inferring time delay in gene regulatory network. *Pacific Symposium on Biocomputing*. 9, pp 474-485.

de Moura A. P. S., Lai Y. C. and Motter A. E. (2003). Signatures of small-world and scale-free properties in large computer programs. *Physical Review. E*. 68, 017102-1-017102-4.

Deb K., Pratap A., Agarwal S. and Meyarivan T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE transaction on evolutionary computation.* 6, pp 182-197. (Code available from http://www.iitk.ac.in/kangal/code/nsga2/nsga2-v1.1.tar.)

Debouck C. and Goodfellow P. N. (1999). DNA microarray in drug discovery and development. *Nature Genetics Supplement*. 21, pp 48-50.

Dhammika A. and Javier C. (2004). Exploration and analysis of DNA microarray and protein array data, *InterScience, John Wiley & Sons, New Jersey*.

Duggan D. J., Bittner M., Chen Y., Meltzer P., and Trent J. M., (1999). Expression profiling using cDNA microarrays. *Nature Genetics Supplement*. 21, pp 10-14.

Duyk G. M. (2002). Sharper tools and simpler methods. *Nature Genetics*. 32, pp 465-468.

Edwards D. R. (1994). Cell signaling and the control of gene transcription. *Trends in Pharmacological Sciences*. 15, pp 239–244.

Edwards R. (2000). Analysis of continuous–time switching networks. *Physica D*. 146, pp 165–199.

Edwards R. (2001). Chaos in neural and gene networks with hard switching. *Differential Equations and Dynamical Systems*. 9, pp 187–220.

Edwards R. and Glass L. (2000). Combinatorial explosion in model gene networks. *Chaos*. 10, pp 691–704.

Elettreby M. F. (2005). Multiobjective bak–sneppen model on a small-world network. *Chaos Solitons and Fractals*. 26, pp 1009–1017.

Friedman N., Murphy K. and Russell S. (1998). Learning the structure of dynamic probabilistic networks. *Proceedings of the 14th Conference on the Uncertainty in Artificial Intelligence*. pp 139–147.

Ganta A. K. (2007). Reverse engineering 'small world' gene networks. *M.Tech. Thesis,* IIT, Kanpur, India.

Garg S. (2002). Systems engineering approaches to computational biology. *Ph.D. Thesis,* University of Connecticut, Storrs, CT, USA.

Garg S., Hansen M. F., Rowe D. W. and Achenie L. E. K. (2003). An adaptive strategy for single- and multi-cluster assignment. *Biotechnology Progress.* 19, pp 1142-1148.

Gerhold D.L., Jensen R.V. and Gullans S. R. (2002). Better therapeutics through microarrays. *Nature Genetics Supplement*. 32, pp 547-552.

Gillespie D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*. 61, pp 2340–2361.

Glass L. and Kauffman S. A. (1973). The logical analysis of continuous, nonlinear biochemical control networks. *Journal of Theoretical Biology*. 39, pp 103–129.

Goodwin B. C. (1965). Oscillatory behavior of enzymatic control processes. *Advances in Enzyme Regulation.* 3, pp 425–439.

Guda K., Cui H., Garg S., Dong M., Nambiar P. R., Achenie L. E. and Rosenberg D. (2003). Multistage gene expression profiling in a differentially susceptible mouse colon cancer model. *Cancer Letters* 191, pp 17-25.

Holloway A. J., Van Laar R. K, Tothill R. W., and Bowtell D. D. L. (2002). Options available—from start to finish—for obtaining data from DNA microarrays II. *Nature Genetics*. 32, pp 481-489.

Ingen C. V. (2002). Biology's century: just beginning for microarrays. *Nature Genetics Supplement*. 32, pp 463.

Iyer V. R., Eisen M. B., Ross D. T., Schuler G., Moore T., Lee J. C. F., Trent J. M., Staudt L. M., Hudson J., Boguski M. S., Lashkari D., Shalon D., Botstein D. and Brown P.O. (1999). The transcriptional program in the response of human fibroblast to serum. *Science.* 283, pp 83-87.

Jacob F. and Monod J. (1961). On the regulation of gene activity. *Cold Spring Harbor Symposia on Quantitative Biology.* 26, pp 193–211.

Kastner J., Solomon J. and Fraser S. (2002). Modeling a hox gene network *in silico* using a stochastic simulation algorithm. *Developmental Biology*. 246, pp 122–131.

Kauffman S. A. (1969a). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*. 22, pp 437–467.

Kauffman S. A. (1969b). Homeostasis and differentiation in random genetic control networks. *Nature*. 224, pp 177–178.

Keasling J. D., Kuo H. and Vahanian G. (1995). A Monte Carlo simulation of the *Escherichia coli* cell cycle. *Journal of Theoretical Biology*. 176, pp 411–430.

Kim S. Y., Imoto S. and Miyano S. (2003). Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*. 4, pp 228-235.

Kleinberg J. M. (2000). The Small-world phenomenon: an algorithmic perspective. *Proceedings of STOC Conference*. pp 163-170.

Lander E. S. (1999). Array of hope. *Nature Genetics Supplement*. 21, pp 3-4.

Maine E.M. and Kimble J. (1989). Identification of genes that interact with glp-1, a gene required for inductive cell interactions in *Caenorhabditis elegans*. *Development*. 106, pp 133-143.

Mason J., Linsay P., Collins J. and Glass L. (2004). Evolving complex dynamics in electronic models of genetic networks. *Chaos*. 14, pp 707–715.

Mathieu J., Griffin K., Herbomel P., Dickmeis T., Strähle U., Kimelman D., Rosa F. M. and Peyrieras N. (2004). Nodal and Fgf pathways interact through a positive regulatory loop and synergize to maintain mesodermal cell populations. *Development*. 131, pp 629-641.

McAdams H. and Arkin A. (1997). Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*. 94, pp 814–819.

McAdams H. and Arkin A. (1998). Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*. 27, pp 199–224.

McAdams H. and Shapiro L. (1995). Circuit simulation of genetic networks. *Science*. 269, pp 650–656.

Mestl T., Bagley R. and Glass L. (1997). Common chaos in arbitrarily complex feedback networks. *Physical Review Letters*. 79, pp 653–656.

Mestl T., Lemay C. and Glass L. (1996). Chaos in high-dimensional neural and gene networks. *Physica. D*. 98, pp 33–52.

Milgram S. (1967). The small world problem. *Psychology Today*. 2, pp 60-67.

Newman M. E. J., Moore C. and Watts D. J. (2000). Mean-field solution of the small-world network model. *Physical Review Letters*. 84, pp 3201-3204.

Okayama H., Nagata A., Jinno S. and Murakami H. (1996). Cell cycle control in fission yeast and mammals: identification of new regulatory mechanisms. *Advances in Cancer Research.* 69, pp 17–62.

Packer A. I., Crotty D. A., Etwell V. A. and Wolgemuth D. J. (1998). Expression of the murine Hoxa4 gene requires both autoregulation and a conserved retinoic acid response element. *Development*. 125, pp 1991-1998.

Perrin B. E., Ralaivola L., Mazurie A., Bottani S., Mallet J. and dAlche–Buc F. (2003). Gene network inference using dynamic Bayesian networks. *Bioinformatics*. 19, pp 138-142.

Plahte E., Mestl T. and Omholt S. W. (1998). A methodological basis for description and analysis of systems with complex switch-like interactions. *Journal of Mathematical Biology*. 36, pp 321–348.

Quackenbush J. (2001). Computational analysis of microarray data. *Nature Genetics*. 2, pp 418-427.

Quackenbush J. (2002). Microarray data normalization and transformation. *Nature Genetics Supplement*. 32, pp 496-501.

Rossant J. and Hopkins N. (1992). Of fin and fur: mutational analysis of vertebrate embryonic. *Genes & Development*. 6, pp 1–13.

Sikarwar G. S. (2005). Array informatics: robust clustering of cDNA microarray data. *M.Tech. Thesis,* IIT, Kanpur, India.

Silvescu A. and Honavar V. (2001). Temporal Boolean network models of genetic networks and their inference from gene expression time series. *Complex Systems*. 13, pp 54–70.

Smolen P., Baxter D. A. and Byrne J. H. (2000). Modeling transcriptional control in gene network-methods, recent results and future directions. *Bulletin of Mathematical Biology*. 62, pp 247–292.

Snoussi E. H. and Thomas R. (1993). Logical identification of all steady states: the concept of feedback loop characteristic states. *Bulletin of Mathematical Biology*. 55, pp 973–991.

Speed T. (2003). Statistical analysis of gene expression microarray data. *Chapman and Hall/CRC, FL, USA.*

Strogatz S. H. (2001). Exploring complex networks. *Nature.* 410. pp 268-276.

Thomas R. and Kaufman M. (2001a). Multistationarity, the basis of cell differentiation and memory, structural conditions of multistationarity and other nontrivial behavior. *Chaos.* 11, pp 170–179.

Thomas R. and Kaufman M. (2001b). Multistationarity, the basis of cell differentiation and memory, logical analysis of regulatory networks in terms of feedback circuits. *Chaos.* 11, pp 180–195.

Tyson J. and Othmer H. G. (1978). The dynamics of feedback control circuits in biochemical pathways. *Progress in Theoretical Biology*. 5, pp 2–62.

Verma R. (2006). Building "small world gene networks" from cDNA microarray data. *M.Tech. Thesis,* IIT, Kanpur, India.

Voit E. (2000). Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists. *Cambridge University Press, UK.*

Wagner A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps. *Bioinformatics*. 17, pp 1183-1197.

Wang X. F. and Chen G. (2003). Complex networks: small-world, scale-free and beyond, *IEEE Circuits and Systems Magazine*. 3, pp 6-20.

Watts D. J. and Strogatz S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*. 393, pp 440-442.

Weaver D., Workman C., and Stormo G. (1999). Modeling regulatory networks with weight matrices. *Pacific Symposium on Biocomputing*. 4, pp 112-123.

Wu F. X., Poirier G.G. and Zhang W. J. (2005). Inferring gene regulatory networks with time delays using a genetic algorithm. *IEEE Proceedings – Systems Biology*, 152, pp 67-74.

Yuh C. H., Bolouri H. and Davidson E. H. (1998). Genomic cis-regulatory logic, experimental and computational analysis of a sea urchin gene. *Science*. 279, pp 1896–1902.

Zill D. G. (2000). First course in differential equations with modeling applications. *Thomson Brooks Cole, CA ,USA.*

Zou M. and Conzen S. D. (2004). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics.* 21, pp. 71-79.

Chapter 13

Optimization of a Multi-Product Microbial Cell Factory for Multiple Objectives – A Paradigm for Metabolic Pathway Recipe

Fook Choon Lee, Gade Pandu Rangaiah* and Dong-Yup Lee
Department of Chemical & Biomolecular Engineering,
National University of Singapore,
Engineering Drive 4, Singapore 117576

*chegpr@nus.edu.sg

Abstract

Genetic algorithms, which have been used successfully in chemical engineering in recent years, enable us to probe deeper into the influences of alternate metabolic pathways on multi-product synthesis. In this chapter, optimization of enzyme activities in *Escherichia coli* for multiple objectives is proposed and explored using a detailed, non-linear dynamic model for central carbon metabolism in *E. coli* and a non-dominated sorting genetic algorithm for multi-objective optimization (MOO). A wide range of optimal enzyme activities regulating the amino acids synthesis is successfully obtained for selected, integrated pathway scenarios. The predicted potential improvements due to the optimized metabolic pathway recipe using the MOO strategy are highlighted and discussed in detail.

Keywords: Multi-product microbial cell factory, Mixed integer MOO (MIMOO), Metabolic pathway recipe, Pareto-optimal set.

13.1 Introduction

Emerging mathematical models for multi-product microbial cell factories such as *Escherichia coli, Corynebacterium glutamicum* and *Saccharomyces cerevisiae* enable chemical engineers to extend the reach of their competencies in systems area such as optimization and process control to biotechnology and biochemical manufacturing. *Corynebacterium glutamicum* is used commercially to produce amino acids (Bongaerts *et al.*, 2001; Ratledge and Kristiansen, 2006) such as L-glutamate (leading to monosodium glutamate, MSG), L-lysine (an animal feed additive), L-phenylalanine (a nutraceutical, a flavour enhancer and an intermediate for synthesis of pharmaceuticals), tryptophan (an animal feed additive and a nutritional ingredient in milk formula for human infants) and L-aspartate (a food additive and a sweetener). Likewise, *Saccharomyces cerevisiae* (yeast) is used commercially to produce ethanol and carbon dioxide (for the baking process in the food industry) as well as bio-fuel. Depending on the bacteria strain, *E. coli* is potentially capable of producing more than twenty types of amino acids (Ratledge and Kristiansen, 2006; Lee *et al.*, 2007; Park *et al.*, 2007).

Optimization of a multi-product microbial cell factory, a systems biotechnology specialty, is increasingly useful in predicting feasible outcomes that fulfill specified objectives, in tandem with rising reliability of the mathematical models for describing the microbial cell metabolic pathways (Lee *et al*., 2005a). This methodology complements the well-developed experimental procedures in classical strain development, genomic techniques and intra-cellular flux analysis, used in engineering a microbial cell factory targeted for industrial production. Rising demand in countries such as China and India is driving the annual growth for amino acids. Hence, improvements in the production of amino acids are of considerable importance to both industries and consumers (Scheper *et al*., 2003). This chapter proposes a mixed-integer multi-objective optimization (MIMOO) study to find a range of better metabolic pathway recipe for improving amino acids production using *E. coli*.

Optimization involves the search for one or more feasible solutions, which correspond to extreme values of one or more objectives. Until about 1980, virtually all problems in chemical engineering were optimized for only one objective (Bhaskar *et al*., 2000). The objective was often economic efficiency expressed as a scalar quantity. In the area

of bio-process modelling and optimization, biochemists such as Voit (2000) have compiled a list of modelling works related to bio-processes using S-system (or synergistic system) to represent the metabolic kinetics of cell factories. The S-system models were then adopted for single objective optimization of citric acid production in *Aspergillus niger*, ethanol production in *Saccharomyces cerevisiae* and tryptophan production in *E. coli* (Torres and Voit, 2002). Conflicting objectives are commonly encountered in chemical and bio-processes (e.g. Lee *et al*., 2007; Chapter 2 in this book). Multi-objective optimization (MOO) involves the search for tradeoffs (or Pareto-optimal front or equally good solutions) when there are conflicting objectives. Up to now, there have been several works in the MOO of bioprocesses but only one, to the best of our knowledge, on the MOO of a multi-product microbial cell factory. See Chapter 2 for the reported applications of MOO in biotechnology.

Almost all works on optimization of a multi-product microbial cell factory focussed on a single objective (e.g., Schmid *et al*., 2004; Visser *et al*., 2004; Vital-Lopez *et al*., 2006). A common feature in these works is the pseudo-stationary assumption. Enzymatic reaction kinetics in a microbial cell factory are reversible and interdependent. In reality, the fluxes due to enzymatic reactions are never stationary. Given the limitations of a model, it is necessary to assume a pseudo-stationary state where some variables fluctuate about an averaged steady state within certain bounds.

Schmid *et al*. (2004) used a nonlinear kinetic model to maximize tryptophan production via enzyme modulations. In their study, the results obtained from piece-wise optimization were combined selectively to form the results of the integrated optimization. Visser *et al*. (2004) used a lin-log kinetic model of *E. coli* to determine the optimal glycolytic (see first paragraph in 13.2) enzyme modulations required to either maximize glucose uptake through phosphotransferase sub-system (PTS) or the production of serine. In this study, only ten (and eleven in the case of maximizing serine production) out of the thirty enzymatic fluxes present in the complete model were the decision variables. These two recent works of nonlinear programming (NLP) illustrate the potential applications of systems biotechnology in generating metabolic pathway recipe.

Vital-Lopez *et al*. (2006) used a linearized kinetic model, possibly anticipating a complex problem to be solved, as the basis for maximizing serine production. In their study, gene overexpression/repression and

knockout[1] are considered for the whole model – an example of mixed-integer nonlinear programming (MINLP). Uncertainty of their results generally increases due to approximation when the optimization search domain recedes further from the initial steady state conditions. Though they have proposed a procedure for unconstrained optimization incorporating both linearized and non-linear kinetic models, its applicability and effectiveness for multi-objective MINLP remain untested.

There has been very little work on MOO of multi-product microbial cell factories. Vera *et al*. (2003) have studied MOO in metabolic processes leading to ethanol production by *Saccharomyces cerevisiae*. Ethanol production which is driven by the enzyme sub-system pyruvate kinase (PK) was maximized and the concentrations of various intermediate metabolites (intra-cellular glucose, g6p, fdp, pep and atp – full form of all abbreviations used is given in the Nomenclature towards the end of the chapter) were minimized under pseudo-stationary conditions where the five metabolite concentrations are assumed to be time-invariant. This allows the system of six differential equations to be converted into an equivalent system of linear algebraic equations (also known as the S-system) by applying natural logarithms to PK kinetic expression (an objective function) and the various influx and efflux terms associated with each of the five metabolites (pseudo-stationary constraints). While searching for a Pareto-optimal set, the metabolites concentrations and enzymes activities – temporally invariant under pseudo-stationary conditions – vary within their respective lower and higher bounds. Enzymes levels (equivalently genes overexpression or repression) and metabolites concentrations were manipulated in the study of Vera *et al*. (2003) without explicit consideration of the impact of gene knockout – a case of multi-objective linear programming (MOLP).

Lee *et al*. (2005b) evaluated the correlation between maximum biomass and succinic acid production for various combinatorial gene knockout strains. This sets the stage for the simultaneous maximization of biomass and succinic acid production using the ε-constraint method. In this method, an MOO problem is converted into an equivalent single

[1]Gene expression is the process by which DNA sequence of a gene is converted into functional proteins. Many proteins are enzymes that catalyze biochemical reactions vital to metabolism. Gene overexpression (e.g. copying genes) creates larger quantity of an enzyme. Gene repression or knockdown reduces the quantity of an enzyme; gene knockout deletes an enzyme-producing gene.

objective problem by constraining all objectives except one to be within specified limits, and then the resulting single objective problems are solved using a suitable NLP or MINLP method. See Chapters 1 and 6 for more details on the ε-constraint method.

In contrast to the widely used continuous processes to produce large quantities of a few products, the strategic importance of MOO of a microbial cell factory is elevated as the challenge to dynamically cater to various market segments and the competition increase. MOO as a first line predictor, has the potential to shorten the time to develop new commercial strains when used in conjunction with the well-established experimental procedures. Clearly, there has been very little MIMOO study of a multi-product microbial cell factory. This motivated us to optimize a multi-product microbial cell for multiple objectives using the elitist non-dominated sorting genetic algorithm (NSGA-II, Deb *et al*., 2002), which has been successfully employed for many chemical engineering applications (see Chapter 2).

13.2 Central Carbon Metabolism of *Escherichia coli*

Fig. 13.1 shows the metabolic network of the central carbon metabolism of *Escherichia coli*. It depicts 30 enzymatic sub-systems (shown in rectangles), 18 metabolites or precursors in between the enzymatic sub-systems and 7 co-metabolites (amp, adp, atp, nadp, nadph, nad and nadh). Enzymes are shown in rectangles; precursors (balanced metabolites) are in bold between enzymes; allosteric effectors (atp, adp and fdp), activators (positive sign), inhibitors (negative sign) and regulators (without sign) are given in circles/ellipses. The glycolytic (consisting of PTS, PGI, PFK, ALDO, TIS, GAPDH, PGK, PGluMu, ENO, PK and PDH enzymatic sub-systems) and pentose-phosphate (consisting of G6PDH, PGDH, Ru5P, R5PI, TKa, TA and TKb enzymatic sub-systems) metabolic pathways are central channels of carbon fluxes. The fluxes of the enzymatic sub-systems have strong influences in the form of feedback regulation (example: changes in PEPCxylase flux affect both the serine and aromatic amino acids synthesis, which occur earlier in the pathway) and feedforward regulation (example: changes in G6PDH flux affect the aromatic amino acids synthesis, which occurs later in the pathway). Higher order effects such as cascade and combined feedback-feedforward regulations are also embedded in the model. Further, metabolites and co-metabolites regulate

the enzymatic sub-systems (example: pep has negative regulatory effect on PFK flux whereas adp and amp have positive regulatory effects on PFK flux); these effects are shown in circles next to the enzymatic sub-systems.

We selected the nonlinear dynamic model of the central carbon metabolism of *E. coli* formulated by Chassagnole *et al*. (2002), to study the effects of genes/enzymes knockouts, overexpression and repression on amino acids synthesis. This detailed model consists of 18 nonlinear differential equations (arising from mass balances) and 30 nonlinear rate equations for the enzymatic sub-systems, which take into account the impacts of gene expression. No differential mass balance equations are available for the 7 co-metabolites; and concentrations of these co-metabolites are assumed to be constant. The co-metabolites (also known as co-factors) contain the food and energy needed to sustain the metabolism of *E. coli*. The microbial cells consume as well as re-generate the co-factors. Due to the biological need of the cell to maintain the concentration of the co-factors and the cyclic nature of co-factor consumption and re-generation, we assume constant concentration for co-metabolites.

Model equations in Chassagnole *et al*. (2002) are not repeated for brevity. But, these equations along with values of parameters in them are available in the text file **E-coli.txt** within the folder **Chapter 13** on the attached compact disk. The kinetic parameters (except for the maximum enzymatic reaction rates), experimentally measured initial steady state values of the metabolites/co-metabolites and fed-batch process parameters are available in Chassagnole *et al*. (2002). The maximum enzymatic reaction rates are taken from an online resource (http://jjj.biochem.sun.ac.za/database/index.html; accessed in May 2007). We solved the model equations using DIVPRK subprogram in the IMSL software, with an integration step of 0.1 sec and a glucose impulse (height = 16 mM; width = 0.1 sec). The transient profiles of the metabolite concentrations and enzymatic reaction fluxes generally agree with those in Chassagnole *et al*. (2002). Initial steady state values of the 18 metabolites computed using the nonlinear equation solver, DNEQNF of the IMSL software, are shown in Table 13.1. They are close to the experimentally measured initial steady state values of Chassagnole *et al*. (2002). All these confirm the validity of the model equations, parameters and programs used in this study.

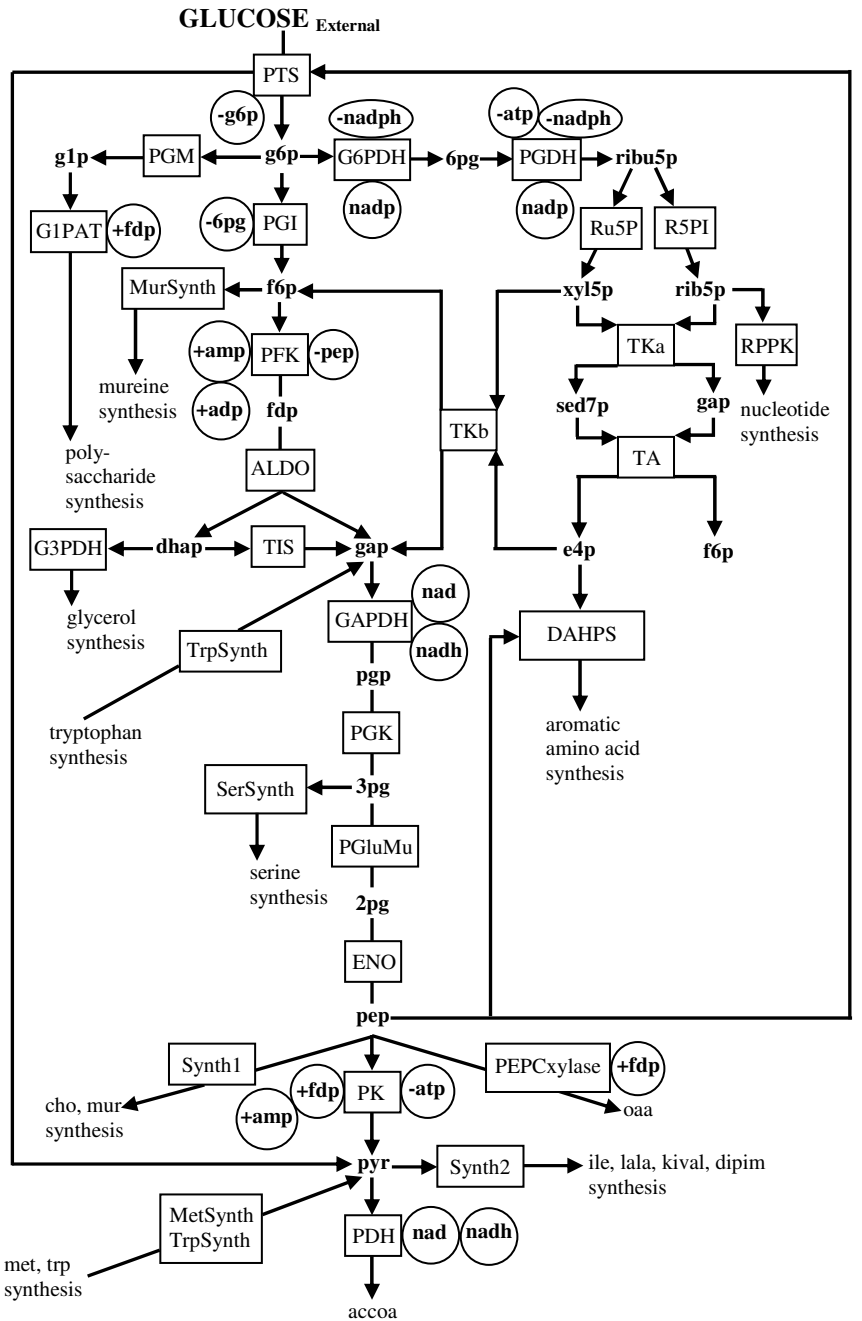


Fig. 13.1 Central carbon metabolism of *Escherichia coli*. See Section 13.2 for details.

13.3 Formulation of the MOO Problem

The DAHPS enzymatic sub-system is the first among a series of steps in the aromatic amino acids (tryptophan, phenylalanine and tyrosine) synthesis pathways. The PEPCxylase enzymatic sub-system produces the oxaloacetate (oaa precursor in Fig. 13.1) to generate the aspartate precursor (not shown in Fig. 13.1) needed for the lysine, methionine, threonine and isoleucine synthesis. The SerSynth enzymatic sub-system governs the steps leading to serine synthesis pathways. The complex interactions among the DAHPS, PEPCxylase and SerSynth enzymatic sub-systems exert definite and possibly conflicting influences on the synthesis of three distinct groups of amino acids (Fig. 13.1).

Maximizing DAHPS, PEPCxylase and SerSynth enzymatic flux ratios are expected to enhance the desired amino acids synthesis rates. Here, enzyme flux refers to the reaction rate facilitated by that enzyme. We opted to study two bi-objective scenarios.

Case A: Maximize DAHPS flux ratio and PEPCxylase flux ratio (13.1)

Case B: Maximize DAHPS flux ratio and SerSynth flux ratio (13.2)

The flux ratio is the ratio of a flux after genetic engineering (to knock out, overexpress and/or repress the genes regulating the enzymatic sub-systems) to that of the reference system before genetic engineering (referred to as wild strain by biotechnologists). We have calculated the initial metabolite/co-metabolite concentrations and steady-state fluxes (given in Table 13.1) to be used as reference values in the MIMOO study, by setting the time derivatives of the metabolites to zero.

The two crucial system constraints in the optimization are homeostasis and total enzymatic flux (Schmid *et al*., 2004). First is the homeostatic constraint:

$$\frac{1}{m}\sum_{i=1}^{m}\frac{\left|C_i - C_{i,ref}\right|}{C_{i,ref}} \le 0.3 \qquad (13.3)$$

The summation is over all metabolites (m in number, 18 in our case). C_i is the concentration of ith metabolite. $C_{i,ref}$ is the reference concentration of the ith metabolite given in Table 13.1. The principle of homeostasis requires the microbial cell to maintain intra-cellular metabolite concentrations within certain bounds (±30% in our study) - a physiological constraint so that the microbial cell does not suffer from

Table 13.1 Initial metabolite/co-metabolite concentrations and steady-state fluxes of enzymes used as reference values in the homeostasis and total enzymatic flux constraints. Experimentally measured values of Chassagnole *et al*. (2002) are in brackets. The co-metabolite concentrations are assumed to be constant.

Metabolite/Co-metabolite with serial number	Concentration (mM)	Enzyme with serial number	Flux (mM/s)
Metabolite		1. PTS	0.2000
1. Glucose (extracellular)	0.05549 (0.0556)	2. PGI	0.05825
2. g6p	3.4767 (3.48)	3. PFK	0.1410
3. f6p	0.5994 (0.60)	4. ALDO	0.1410
4. fdp	0.2703 (0.272)	5. TIS	0.1394
5. gap	0.2173 (0.218)	6. GAPDH	0.3199
6. dhap	0.1665 (0.167)	7. PGK	0.3199
7. pgp	0.00798 (0.008)	8. PGluMu	0.3023
8. 3pg	2.1268 (2.13)	9. ENO	0.3023
9. 2pg	0.3982 (0.399)	10. PK	0.03811
10. pep	2.6648 (2.67)	11. PDH	0.1878
11. pyr	2.6689 (2.67)	12. PEPCxylase	0.04312
12. 6pg	0.8138 (0.808)	13. PGM	0.002319
13. ribu5p	0.1108 (0.111)	14. G1PAT	0.002301
14. xyl5p	0.1378 (0.138)	15. RPPK	0.01031
15. sed7p	0.2760 (0.276)	16. G3PDH	0.001658
16. rib5p	0.3974 (0.398)	17. SerSynth	0.01749
17. e4p	0.09776 (0.098)	18. Synth1	0.01421
18. g1p	0.6520 (0.653)	19. Synth2	0.05355
		20. DAHPS	0.006836
Co-metabolite		21. G6PDH	0.1393
1. amp	0.955 (0.955)	22. PGDH	0.1393
2. adp	0.595 (0.595)	23. Ru5P	0.08370
3. atp	4.27 (4.27)	24. R5PI	0.05559
4. nadp	0.195 (0.195)	25. TKa	0.04527
5. nadph	0.062 (0.062)	26. TKb	0.03843
6. nad	1.47 (1.47)	27. TA	0.04526
7. nadh	0.1 (0.1)	28. MurSynth	0.00043711
		29. MetSynth	0.0022627
		30. TrpSynth	0.001037

toxic or inhibitory effects - to avoid impediment of cellular functions and undesirable flux diversions. Large changes in metabolite concentrations cause unforeseeable effects on gene expression (and hence kinetic rate parameters) that are not captured in the existing model.

The second is the total enzymatic flux constraint:

$$\frac{1}{z}\sum_{i=1}^{z}\frac{r_i}{r_{i,\,ref}} \le 1.0 \qquad (13.4)$$

This is a technological constraint, where the summation covers all enzyme fluxes (z in number, 30 in our case). $r_{i,ref}$ is the ith reference enzymatic reaction rate given in Table 13.1. Total enzymatic activity is constrained not to exceed 1.0 to avoid diffusion problem (due to increased cytoplasm viscosity), protein precipitation, secondary kinetic effects (due to steric hindrance) and excessive intracellular stress leading to unpredictable regulatory effects. When either one of these two constraints is breached, the objective function value is penalized by setting it to an arbitrarily low level; under such conditions, the DAHPS, PEPCxylase and SerSynth fluxes are set to 10^{-20}.

Metabolite concentrations and enzymatic fluxes change from one steady state to another due to gene knockouts, overexpression and/or repression. Redistribution of the fluxes presents opportunities for optimizing the metabolic pathways subject to physiological and technological constraints. Translation of gene knockouts, overexpression and repression into decision variables is described in the next section. In this chapter, gene knockouts and overexpression/repression are considered separately. Work on the simultaneous knockouts and manipulation of genes is in progress.

13.4 Procedure used for Solving the MIMOO Problem

By setting the time-derivative of each metabolite concentration to zero under pseudo-stationary assumption, the set of differential equations for mass balance equations is converted into a system of algebraic equations (see the **E-coli.txt** file in the folder **Chapter 13** on the attached compact disk). Each nonlinear equation contains several rate expressions and terms. The glucose impulse term, f_{pulse}, in the mass balance equation $\left(\frac{dC_{glc}^{extracellular}}{dt} = D\left(C_{glc}^{feed} - C_{glc}^{extracellular}\right) + f_{pulse} - \frac{C_x r_{PTS}}{\rho_x}\right)$ in the work of Chassagnole *et al*. (2002) is used to generate transient profiles using the

original system of differential equations, and hence it is not relevant to the MIMOO study. The system of 18 algebraic equations is solved for 18 metabolite concentrations, using the DNEQNF subprogram in the IMSL FORTRAN libraries. The entire study was done using a personal computer with 2 GHz Pentium$^{(R)}$ IV CPU, 1 GB RAM and Windows XP Professional. Each solution of the system of algebraic equations took no more than 3 seconds of CPU time on this computer; each optimization run for 500 generations (using NSGA-II as described later in this section) required less than 20 minutes of CPU time for each of the bi-objective cases.

One main difficulty encountered in the optimization of the microbial cell factory is to identify the enzymes to be knocked out. Enzymes cannot be deleted arbitrarily – certain enzymes are essential for the metabolic network integrity. An attempt to delete essential enzymes results in the termination of the DNEQNF subprogram and consequently the optimization program too. To overcome this problem, we have identified in advance feasible sets of 1-enzyme, 2-enzyme and 3-enzyme knockouts through a manual combinatorial exercise (by setting the maximum reaction rates of the selected enzymatic sub-systems to zero and solving the model equations). Although this takes considerable effort and time, identifying enzymes which can be deleted, whether singly or in groups, circumvents numerical difficulties in the MIMOO study. The number of feasible sets (with all combinations in brackets) of 1-, 2- and 3-enzyme knockouts are 15 (30), 114 (435) and 665 (4060) respectively. These sets are neither available in the literature nor known *a priori*. This manual combinatorial exercise provides the Pareto-optimal sets by enzyme knockouts as well. Instead of a manual combinatorial exercise, other strategies to identify genes which can be knocked out prior to optimizing simultaneous gene knockout and manipulation are being explored.

Gene manipulations (overexpression and/or repression) are optimized using the NSGA-II and the FORTRAN program containing the model and its solution. Decision variables can be implemented with binary or real coding in the NSGA-II program. Decision variables in the form of integers from 1 to 30 are used to denote the enzymatic sub-systems (or simply enzymes). A 5-bit binary variable which covers integers ranging from 1 to 32 (where 31 and 32 are not used) is used as a decision variable for 1-enzyme manipulation. Two or more 5-bit variables are needed in the multi-enzyme manipulation. For gene overexpression/repression, real

decision variables in the range 0.5 to 2.0 are also used to multiply the maximum enzymatic reaction rates; these bounds are selected after preliminary optimization runs with several ranges for these decision variables. In this study, the number of gene knockouts or expression for optimization is limited to a maximum of 3 due to potential difficulties in achieving more knockouts/expression experimentally. Translation of gene knockouts and expression into decision variables and their implementation in the optimization can easily be done without any concern on continuity since evolutionary algorithms such as NSGA-II are applicable to non-differentiable functions.

Using the glucose-6-phosphate dehydrogenase (G6PDH) enzymatic sub-system as an example, reaction rate of G6PDH (i.e., rate of reaction facilitated by G6PDH) is given as:

$$r_{G6PDH} = \frac{r^{max}_{G6PDH} C_{g6p} C_{nadp}}{\left(C_{g6p} + K_{G6PDH,g6p}\right)\left(1 + \frac{C_{nadph}}{K_{G6PDH,nadph,g6pinh}}\right)\left(K_{G6PDH,nadp}\left(1 + \frac{C_{nadph}}{K_{G6PDH,nadph,nadpinh}}\right) + C_{nadp}\right)} \quad (13.6)$$

where the maximum enzymatic reaction rate is r^{max}_{G6PDH}. The integer number representing an enzyme follows the sequence given in Table 13.1. Therefore, G6PDH is numbered 21. The integer number which is a discrete decision variable is selected randomly by NSGA-II. For a selected enzymatic sub-system, its maximum reaction rate is set to zero in gene knockout. In 1-enzyme manipulation study, two decision variables are involved: an integer number representing an enzymatic sub-system and a real number representing gene overexpression or repression. If G6PDH (numbered 21) were selected, its maximum reaction rate will be multiplied by a real number in the range from 0.5 to 2.0, selected by NSGA-II. Gene is overexpressed or repressed if the chosen real number is greater or less than 1.0 respectively.

NSGA-II parameters used in this study are: maximum number of generations (up to 500), population size (100 chromosomes), probability of crossover (0.85), probability of mutation (0.05), distribution index for the simulated crossover operation (10), distribution index for the simulated mutation operation (20) and random seed (0.6). Except for the first and last parameter listed here, rest of the NSGA-II parameter values are taken from Tarafder *et al*. (2005). Values for maximum number of generations and random seed are obtained by trial and error. Our preliminary gene manipulation optimization runs show convergence within 500 generations for the random seed of 0.6.

13.5 Optimization of Gene Knockouts

The Pareto-optimal metabolic pathway recipe through multi-gene knockout combinations shows that the triple-gene knockout has the best non-dominated flux ratios due to greater flexibility in manipulating fluxes for the various pathways (Fig. 13.2). Deleting PGM in the single-gene knockout generates the Pareto-optimal flux ratios for both the bi-objective scenarios (Fig. 13.2) subject to the homeostasis and total enzyme activity constraints. No flux appears in PGM, and the fdp activation results in negligible G1PAT flux. Fluxes of the glycolytic (consisting of PTS, PGI, PFK, ALDO, TIS, GAPDH, PGK, PGluMu, ENO, PK and PDH) and pentose-phosphate pathway (starting from G6PDH) undoubtedly increase.

PGM and G6PDH are the main catalysts that channel carbon sources for the polysacharride synthesis and pentose-phosphate pathway respectively. The pentose-phosphate pathway has a higher carbon utilization rate and level than that needed in polysacharride synthesis. Under unconstrained condition, deleting G6PDH which is a major user of carbon sources, leads to a significant increase in the Pareto-optimal fluxes. However, knocking out G6PDH violates the total enzymatic flux constraint. If another gene is also knocked out, the total enzymatic flux and homeostatic constraints are breached in 54% and 77% of the cases, respectively.

Deleting PK and G1PAT (Fig. 13.2 – chromosome B_1) or G6PDH and MetSynth (Fig. 13.2 - chromosome B_2) in the double-gene knockout generates Pareto-optimal flux ratios for Case B of the bi-objective scenarios; deleting G6PDH and MetSynth (Fig. 13.2 – chromosome A_1) generate Pareto-optimal flux ratios for Case A also. Deleting G6PDH and MetSynth increases fluxes (Table 13.2 and Fig. 13CD.1 on the attached compact disk) of the glycolytic pathway. In contrast, most of reactions in the pentose-phosphate pathway were not actively utilized as exhibited by zero fluxes of G6PDH and PGDH, largely attenuated fluxes of R5PI, TKa and TA, and inverse fluxes of RU5P and TKb. The inverse (or negative) fluxes are the result of product formation rate being greater than the reactant influx rate. Inverse fluxes signify that carbon sources are being drained from the pentose-phosphate pathway by the glycolytic pathway more quickly than they are replenished via the same pathway – this is equivalent to the backflow of carbon sources. TKb is a gateway supplying carbon from the pentose-phosphate pathway to the glycolytic

precursors (f6p and gap). Constant PTS flux and to a smaller extent MetSynth deletion assist in maintaining PK, PDH and Synth2 flux levels.

The DAHPS flux ratio is the highest among the three objectives due to relatively greater increase in its precursor concentrations (28% for pep and 33% for e4p) in comparison to the enhancing effects of precursors dictating the flux of PEPCxylase (28% for pep coupled with activation via fdp) and SerSynth (28% for 3pg). Double knockout (PK and G1PAT) results in an overall increase of the concentration levels of glycolytic precursors, slightly attenuating the glycolytic fluxes. The concentration levels of 3pg and pep, precursors for the three amino acids synthesis pathways, increase by 41%; the concentration level of e4p (a precursor of DAHPS) increases by 29%. SerSynth flux ratio increases as its precursor (3pg) concentration level increases from 28% to 41%. However, both the DAHPS and PEPCxylase flux ratios decrease due to self-regulatory effects, as their precursor concentration levels increase beyond the corresponding levels obtainable from deleting G6PDH and MetSynth.

Triple knockout of PK, G1PAT and G3PDH generates Pareto-optimal flux ratios in Case B of the bi-objective scenarios while deleting G1PAT, RPPK and DAHPS generates Pareto-optimal flux ratios in Case A. Similar to that of the double-gene knockout, deleting PK, G1PAT and G3PDH results in an overall increase of the glycolytic precursors concentration levels and a slight attenuation of the glycolytic fluxes. Also, the concentration levels of the three precursors (3pg, pep and e4p) increase by almost the same percentage points as those of the double-gene knockouts. In Case A, deleting G1PAT, RPPK and DAHPS maximizes the PEPCxylase flux ratio as the DAHPS flux ratio is set to zero. The glycolytic and pentose-phosphate pathway fluxes are amplified and the carbon sources of e4p and pep are diverted from DAHPS to the glycolytic pathway and PEPCxylase, respectively. Another Pareto-optimal metabolic pathway recipe in Cases A and B is obtained by deleting G6PDH, MurSynth and TrpSynth (chromosomes adjacent to A_1 and B_2 in Fig. 13.2). Similar to that of the 2-enzyme knockout, the DAHPS flux ratio is the highest among all the three scenarios. Except for the zero concentration of 6pg, the concentration levels of the remaining metabolites are elevated. There is an overall flux increase in glycolytic pathway, polysaccharide synthesis, nucleotide synthesis and glycerol synthesis. The pentose-phosphate pathway exhibits zero fluxes (G6PDH and PGDH), largely attenuated fluxes (R5PI, TKa and TA) and inverse fluxes (RU5P and TKb).

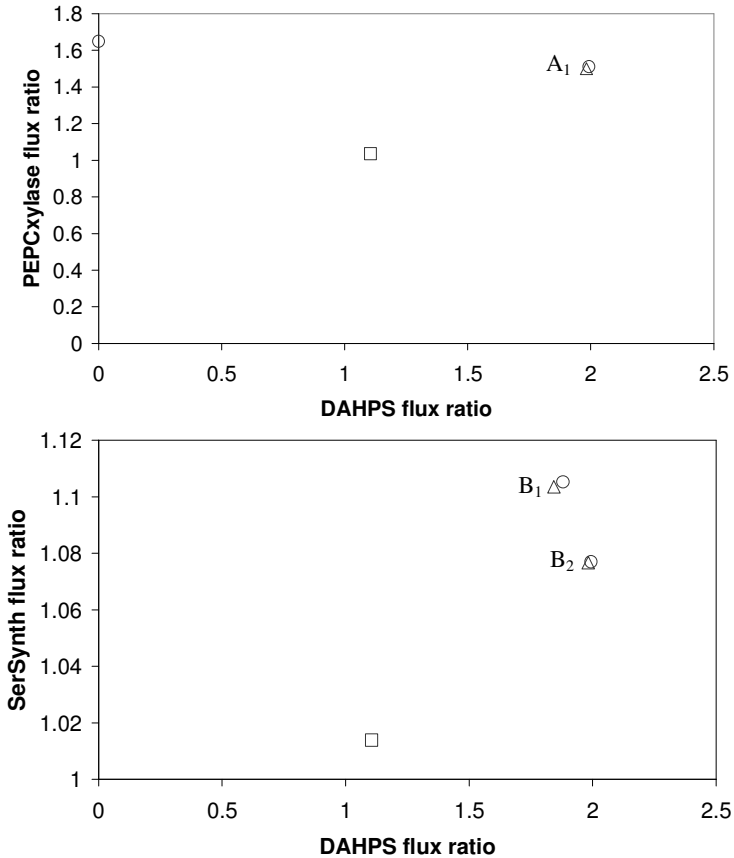


Fig. 13.2 Pareto results for gene knockouts (single-gene □; double-gene Δ; triple-gene ○) in simultaneous maximization of (a) DAHPS and PEPCxylase flux ratios, and (b) DAHPS and SerSynth flux ratios. The chromosomes in double-gene knockouts are labelled.

13.6 Optimization of Gene Manipulation

The optimization of the Pareto-optimal metabolic pathway recipe by genetic manipulations shows that the triple-enzyme manipulation has the best non-dominated flux ratios due to the flexibility in changing enzymatic reaction rates to enhance the desired flux ratios (Figs.13.3A and 13.4A). The left and right Pareto-optimal segments of single-enzyme

manipulation (Figs. 13.3A and 13.3B) are governed respectively by PEPCxylase and DAHPS in Case A, and SerSynth and DAHPS in Case B (Figs. 13.4A and 13.4B). The single chromosome in between the two segments is the result of manipulating PFK in both Cases A and B. Fluxes of the polysaccharide and glycerol synthesis, and to a lesser extent pentose-phosphate pathway and nucleotide synthesis are continually attenuated on ascending the left Pareto-optimal segment and descending the right Pareto-optimal segment when the governing gene of each segment is being overexpressed. As a result, the fluxes are redistributed among DAHPS, PEPCxylase and SerSynth. Metabolite concentrations follow similar trend as carbon sources are diverted towards building gene molecules. The in-between chromosomes, with manipulation factors set to an upper bound of 2.0, are pivotal to the generation of the two distinct Pareto-optimal segments when a governing enzyme is being switched.

In the double-enzyme manipulation, the left and right Pareto-optimal segments (Figs. 13.3A and 13.3B) in Case A are governed by PEPCxylase/G6PDH and DAHPS/G6PDH, respectively. The manipulation factor of G6PDH is actively constrained at 0.5 (lower bound) to divert fluxes away from the pentose-phosphate pathway (the concentration of 6pg decreases by about 50%), thus resulting in higher fluxes in glycolytic pathway (PGI flux is the highest), polysaccharide synthesis, nucleotide synthesis and glycerol synthesis. The PEPCxylase flux ratio increases as PEPCxylase manipulation factor approaches 2.0 (upper bound) on ascending the left Pareto segment (Figs. 13.3A and 13.3B). Similarly, the DAHPS flux ratio increases as DAHPS manipulation factor approaches the upper bound on descending the right Pareto segment. Interestingly, when the paired enzyme is switched from A_1 to A_2, fluxes from other pathways are drawn towards DAHPS instead of PEPCxylase (Table 13.3 and Fig. 13CD.2 on the attached compact disk), indicating a distinct change in the metabolic pathway recipe.

The leftmost Pareto-optimal segment (Figs. 13.4A and 13.4B) in Case B of the double-enzyme manipulation is governed by SerSynth and GAPDH. The manipulation factor of SerSynth hardly deviates from its upper bound of 2.0 (Figs. 13.4A and 13.4B) and the manipulation factor of GAPDH changes from 2 (at chromosome B_1) to 1.27 (Figs. 13.4A and 13.4B). Fluxes are drained from DAHPS, polysaccharide synthesis and glycerol synthesis to sustain high SerSynth reaction rate. The concentrations of fdp and gap, both precursors being the proximate

Table 13.2 Pareto-optimal metabolic pathway recipe for 2-enzyme knockouts represented by the three labelled chromosomes in Fig. 13.2. The flux ratios are listed in the second, third and fourth column (chromosomes A_1, B_1 and B_2, respectively) for each enzyme. The same enzymes (G6PDH and MetSynth) are knocked out in both chromosomes A_1 and B_2.

Enzyme with serial number	Flux ratios of chromosomes A_1 and B_2	Flux ratios of chromosome B_1
1. PTS	1.000	1.000
2. PGI	3.348	0.772
3. PFK	1.294	0.964
4. ALDO	1.294	0.964
5. TIS	1.294	0.960
6. GAPDH	1.098	0.971
7. PGK	1.098	0.971
8. PGluMu	1.099	0.963
9. ENO	1.099	0.963
10. PK	1.024	0 (knocked out)
11. PDH	0.993	0.802
12. PEPCxylase	1.500	1.460
13. PGM	2.078	0.0093
14. G1PAT	2.086	0 (knocked out)
15. RPPK	1.022	1.034
16. G3PDH	1.340	1.291
17. SerSynth	1.077	1.104
18. Synth1	1.064	1.087
19. Synth2	0.999	0.983
20. DAHPS	1.982	1.843
21. G6PDH	0 (knocked out)	1.112
22. PGDH	0	1.112
23. Ru5P	− 0.138	1.098
24. R5PI	0.208	1.132
25. TKa	0.022	1.154
26. TKb	− 0.327	1.032
27. TA	0.022	1.154
28. MurSynth	1.000	1.000
29. MetSynth	0 (knocked out)	1.000
30. TrpSynth	1.000	1.000

Table 13.3 Pareto-optimal metabolic pathway recipe for 2-enzyme manipulations represented by the four labelled chromosomes in Figs. 13.3A and 13.4A. The flux ratios are listed in second, third, fourth and fifth column (chromosomes A_1, A_2, B_1 and B_2, respectively) for each enzyme.

Enzyme with serial number	Flux ratios of chromosome A_1	Flux ratios of chromosome A_2	Flux ratios of chromosome B_1	Flux ratios of chromosome B_2
1. PTS	1.000	1.000	1.000	1.000
2. PGI	2.117	2.139	1.037	1.096
3. PFK	1.143	1.137	1.024	1.008
4. ALDO	1.143	1.137	1.024	1.008
5. TIS	1.143	1.137	1.030	1.010
6. GAPDH	1.051	1.035	1.033	0.996
7. PGK	1.051	1.035	1.033	0.996
8. PGluMu	1.052	1.036	0.978	0.994
9. ENO	1.052	1.036	0.978	0.994
10. PK	1.011	1.007	0.996	0.985
11. PDH	1.002	1.001	0.999	0.997
12. PEPCxylase	1.287	1.067	0.962	0.907
13. PGM	1.383	1.242	0.455	0.782
14. G1PAT	1.385	1.244	0.450	0.781
15. RPPK	1.009	1.002	0.959	0.985
16. G3PDH	1.146	1.095	0.558	0.889
17. SerSynth	1.034	1.022	1.978	1.022
18. Synth1	1.029	1.018	0.991	0.961
19. Synth2	1.000	1.000	1.000	1.000
20. DAHPS	1.378	2.077	0.316	1.498
21. G6PDH	0.526	0.520	0.994	0.964
22. PGDH	0.526	0.520	0.994	0.964
23. Ru5P	0.463	0.437	1.015	0.947
24. R5PI	0.621	0.643	0.962	0.988
25. TKa	0.533	0.561	0.962	0.989
26. TKb	0.382	0.291	1.077	0.898
27. TA	0.532	0.561	0.962	0.989
28. MurSynth	1.000	1.000	1.000	1.000
29. MetSynth	1.000	1.000	1.000	1.000
30. TrpSynth	1.000	1.000	1.000	1.000

carbon sources of 3pg (SerSynth precursor), substantially decrease by 70% and 50% (at chromosome B_1), respectively, thereby leading to the enhanced production of serine. The three leftmost chromosomes consisting of SerSynth and PFK on the same Pareto segment furthest from chromosome B_2 are pivotal to the generation of the two distinct Pareto-optimal segments containing chromosomes B_1 and B_2 which are governed by the paired enzymes SerSynth/GAPDH and DAPHS/SerSynth, respectively. The DAHPS flux ratio increases and the SerSynth flux ratio decreases (Table 13.3 and Fig. 13CD.2 on the attached CD) as the manipulation factor of DAHPS approaches 2.0 (1.07 for SerSynth) on moving towards chromosome B_2. The concentrations of fdp and gap decrease by 23% and 13% (at chromosome B_2) respectively as serine production declines. The three chromosomes on the right side of chromosome B_2 consisting of DAHPS and G6PDH (manipulation factor of 0.5) represent high DAHPS production rate and relatively constant serine production rate. The DAHPS flux ratio increases considerably when the DAHPS manipulation factor approaches 2.0 resulting in simultaneous increase of glycolytic and decrease of pentose-phosphate fluxes.

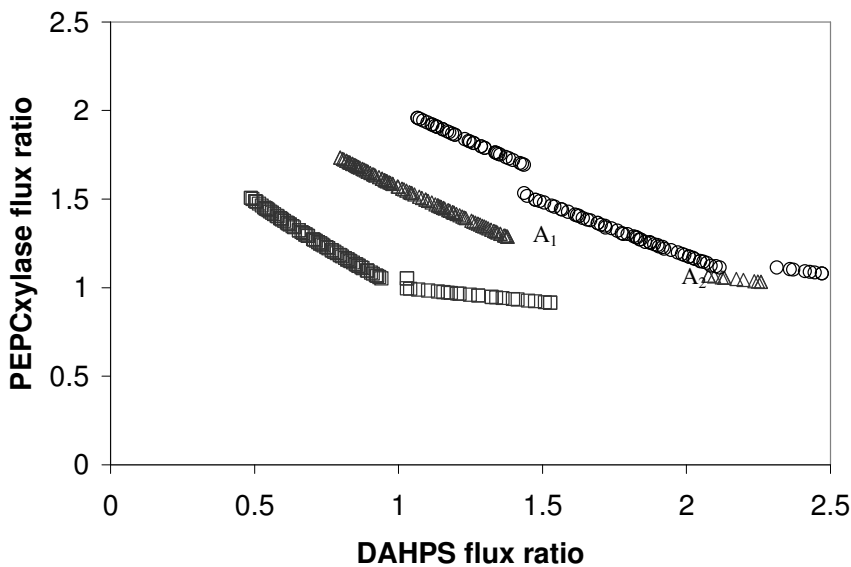


Fig. 13.3A Pareto-optimal fronts for gene manipulations (1-enzyme □; 2-enzyme Δ; 3-enzyme ○) in simultaneous maximization of DAHPS and PEPCxylase flux ratios (Case A).

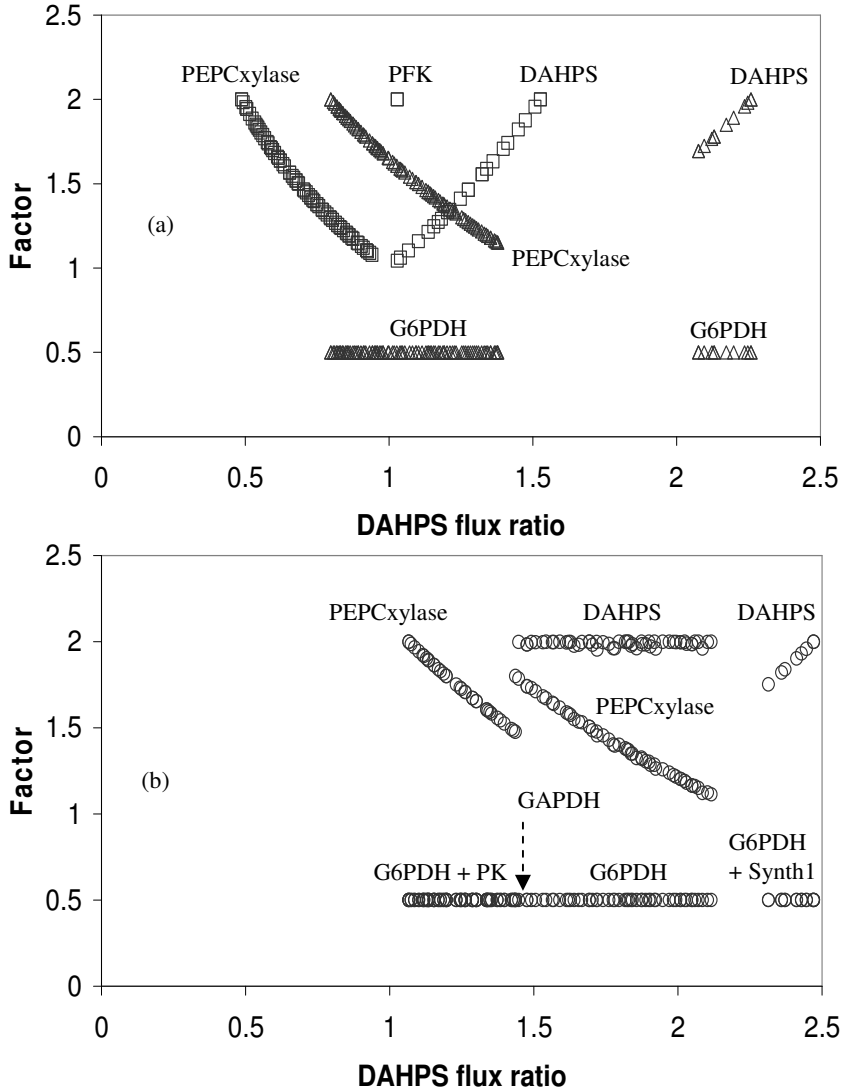


Fig. 13.3B Pareto-optimal enzyme manipulation factors in simultaneous maximization of DAHPS and PEPCxylase flux ratios (Case A) (a) 1-enzyme (□) and 2-enzyme (Δ) manipulation factor and (b) 3-enzyme manipulation factor (○).

In the triple-enzyme manipulation, the left Pareto segment in Case A with comparatively high PEPCxylase flux ratio (Figs. 13.3A and 13.3B) is obtained by repressing the activities of genes for PK and G6PDH and by overexpressing gene for PEPCxylase, which result in largely attenuated pentose-phosphate pathway (50% decrease in 6pg concentration) and PK fluxes. The leftmost chromosome of the central optimal Pareto segment, which is obtained by repressing genes related to GAPDH and G6PDH and overexpressing gene related to PEPCxylase, is pivotal in switching the flux control to genes related to DAHPS, PEPCxylase and G6PDH. The enzyme manipulation factors of G6PDH and DAHPS are actively constrained at 0.5 and 2.0, respectively. On descending the central segment of the Pareto-optimal front, the enzyme manipulation factor of PEPCxylase decreases from 1.79 to 1.11 (Figs. 13.3A and 13.3B) to generate successively higher DAHPS flux ratio. Even higher DAHPS flux ratio is obtained by switching the flux control to genes related to SYN1, G6PDH and DAHPS. The enzyme manipulation factors of SYN1 and G6PDH are actively constrained at 0.5 to restrict competing fluxes in the chorismate and mureine synthesis, and pentose-phosphate pathways respectively. Under such conditions, the right Pareto-optimal segment is formed as the enzyme manipulation factor of DAHPS increases from 1.75 to 2.0.

Except for the few leftmost chromosomes in Case B (Figs. 13.4A and 13.4B) obtained by overexpressing genes related to PFK and SerSynth and repressing gene related to G6PDH, the metabolic pathway recipe on descending the Pareto-optimal segment is formed by simultaneously reducing, increasing and constraining the enzyme manipulation factors of SerSynth, DAHPS and G6PDH, respectively. Similar to that of Case A, the relatively flat profile on the right is formed by manipulating genes related to SYN1, G6PDH and DAHPS.

In general, optimization results are as good as the model used in the study. Some uncertainty in any model and its parameters is unavoidable, particularly in case of complex biological systems such as *E. coli* and after genetic engineering of living organisms. Hence, results of optimizing multi-product microbial cell factories will have to be confirmed through experimental studies. These will be explored in our future work.

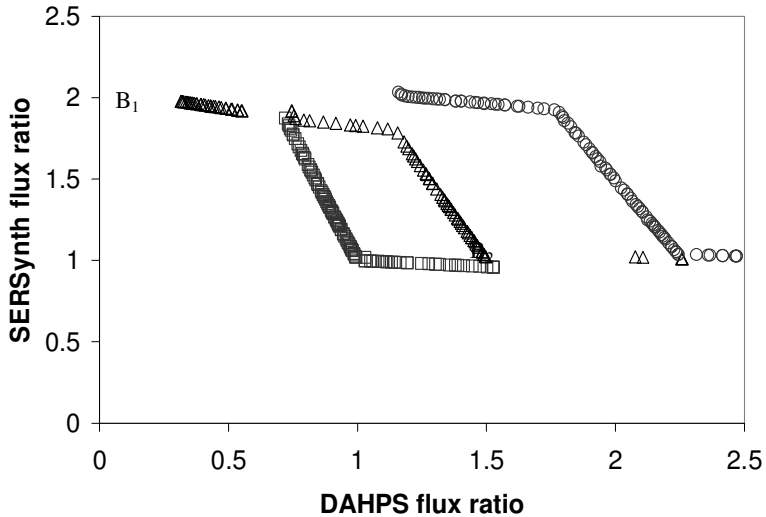


Fig. 13.4A Pareto-optimal fronts for gene manipulations (1-enzyme □; 2-enzyme Δ; 3-enzyme ○) in simultaneous maximization of DAHPS and SerSynth flux ratios (Case B).

13.7 Conclusions

In this chapter, MOO of fluxes of desired enzymatic sub-systems in *E. coli* is described; two cases - maximization of DAHPS and PEPCxylase fluxes, and maximization of DAHPS and SerSynth fluxes, by 1-, 2- and 3-gene knockout and manipulation are considered. Optimal Pareto solutions were successfully obtained using the NSGA-II program wherein integer and/or continuous decision variables can be used; this flexibility allowed seamless use of both types of variables in the MOO of enzyme fluxes in *E. coli*. Knocking out PGM gives the 1-enzyme Pareto-optimal set. In 1-enzyme manipulation cases, the gene that generates the Pareto-optimal set is directly related to the desired enzymatic activity. In paired and triple enzyme knockout or manipulation, G6PDH is instrumental in diverting fluxes to the desired metabolic pathways. The triple enzyme knockout/manipulation gives the best Pareto-optimal set due to greater flexibility in redistributing fluxes to the desired pathways. In the triple enzyme knockout, DAHPS and PEPCxylase fluxes increase by 51% and 99% respectively in one case while DAPHS and SerSynth fluxes increase by around 95% and 9% respectively in another case. In the triple enzyme manipulation, the DAHPS and PEPCxylase fluxes

increase up to 247% and 96% respectively in one case while DAHPS and SerSynth fluxes increase up to 247% and 203% respectively in another case. It is possible to consider gene knockout and manipulation simultaneously as well as more number of genes for MOO of fluxes of desired enzymatic sub-systems; this investigation is in progress.

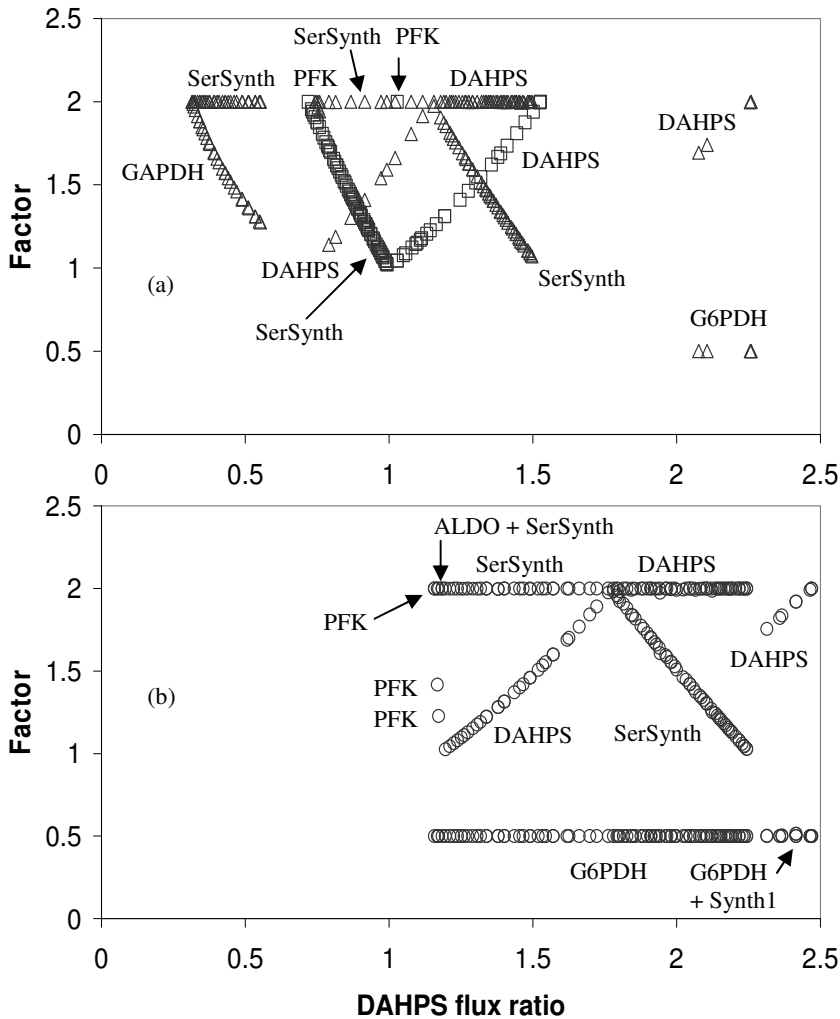


Fig. 13.4B Pareto-optimal enzyme manipulation factors in simultaneous maximization of DAHPS and SerSynth flux ratios (case B) (a) 1-enzyme (□) and 2-enzyme (Δ) manipulation and (b) 3-enzyme manipulation (○).

Nomenclature

Enzymes

ALDO	aldolase
DAHPS	3-deoxy-D-arabino-heptulosonate-7-phosphate synthase
ENO	enolase
G1PAT	glucose-1-phosphate adenyltransferase
G3PDH	glycerol-3-phosphate dehydrogenase
G6PDH	glucose-6-phosphate dehydrogenase
GAPDH	glyceraldehyde-3-phosphate dehydrogenase
MetSynt	methionine synthesis
MurSynth	mureine synthesis
PFK	phosphofructokinase
PGDH	6-phosphogluconate dehydrogenase
PGI	glucose-6-phosphate isomerase
PGK	phosphoglycerate kinase
PGluMu	phosphoglycerate mutase
PDH	pyruvate dehydrogenase
PEPCxylase	PEP carboxylase
PGM	phosphoglucomutase
PK	pyruvate kinase
PTS	phosphotransferase system
R5PI	ribose-phosphate isomerase
RPPK	ribose-phosphate pyrophosphokinase
Ru5P	ribulose-phosphate epimerase
SerSynth	serine synthesis
Synth1, Synth2	synthesis 1 and synthesis 2
TA	transaldolase
TIS	triosephosphate isomerase
TKa	transketolase, reaction a
TKb	transketolase, reaction b
TrpSynth	tryptophan synthesis

Metabolites

2pg, 3pg	2-, 3- phosphoglycerate
6pg	6-phosphogluconate
accoa	acetyl-coenzyme A
cho	chorismate
dhap	dihydroxyacetonephosphate

dipim	diaminopimelate
e4p	erythrose-4-phosphate
f6p	fructose-6-phosphate
fdp	fructose-1,6-biphosphate
g1p, g6p	glucose-1-phosphate and glucose-6-phosphate
gap	glyceraldehyde-3-phosphate
glc	glucose
ile	isoleucine
kival	α-ketoisovalerate
lala	L-alanine
met	methionine
mur	mureine
oaa	oxaloacetate
pep	phosphoenolpyruvate
pgp	1,3-diphosphoglycerate
pyr	pyruvate
rib5p	ribose-5-phosphate
ribu5p	ribulose-5-phosphate
sed7p	sedoheptulose-7-phosphate
trp	tryptophan
xyl5p	xylulose-5-phosphate

Co-metabolites (unbalanced)

adp	adenosindiphosphate
amp	adenosinmonophosphate
atp	adenosintriphosphate
nad	diphosphopyridindinucleotide, oxidized
nadh	diphosphopyridindinucleotide, reduced
nadp	diphosphopyridindinucleotide-phosphate, oxidized
nadph	diphosphopyridindinucleotide-phosphate, reduced

Symbols

$C_{glc}^{extracellular}$	extracellular glucose concentration (mM)
C_{glc}^{feed}	glucose feed concentration (mM)
c_{g6p}	concentration of glucose-6-phosphate (mM)
c_i	i^{th}-metabolite concentration (mM)
$c_{i,ref}$	i^{th}-metabolite concentration (mM) at reference conditions

c_{nadp}	concentration of diphosphopyridindinucleotide-phosphate, oxidized (mM)
c_{nadph}	concentration of diphosphopyridindinucleotide-phosphate, reduced (mM)
c_x	biomass concentration (g dry weight/L broth)
D	dilution factor (/s)
f_{pulse}	glucose impulse
$K_{G6PDH,g6p}$	kinetic constant (mM)
$K_{G6PDH,nadp}$	kinetic constant (mM)
$K_{G6PDH,nadph,g6pinh}$	inhibition constant (mM)
$K_{G6PDH,nadph,nadpinh}$	inhibition constant (mM)
m	number of metabolites (= 18)
r_{G6PDH}^{max}	maximum reaction rate of glucose-6-phosphate dehydrogenase
r_i	i^{th}-enzymatic reaction rate (mM/s)
$r_{i,ref}$	i^{th}-enzymatic reaction rate (mM/s) at reference conditions
r_{PTS}	phosphotransferase system flux (mM/s)
z	number of enzymatic sub-systems (= 30)

Greek symbols

ρ_x	microbial cell density (g dry weight/L cell)

References

Bhaskar, V., Gupta, S. K. and Ray, A. K. (2000). Applications of multiobjective optimization in chemical engineering. *Reviews in Chemical Engineering, 16*(1), 1-54.

Bongaerts, J., Krämer, M., Müller, U., Raeven, L. and Wubbolts, M. (2001). Metabolic engineering for microbial production of aromatic amino acids and derived compounds. *Metabolic Engineering, 3*(4), 289-300.

Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K. and Reuss, M. (2002). Dynamic modeling of the central carbon metabolism of *Escherichia coli*. *Biotechnology and Bioengineering, 79*(1), 53-73.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182-197.

Lee, F. C., Rangaiah, G. P. and Ray, A. K. (2007). Multi-objective optimization of an industrial penicillin V bioreactor train using non-dominated sorting genetic algorithm. *Biotechnology and Bioengineering, 98*(3), 586-598.

Lee, K. H., Park, J. H., Kim, T. Y., Kim, H. U. and Lee, S. Y. (2007). Systems metabolic engineering of *Escherichia coli* for L-threonine production. Molecular Systems Biology, *3*, 149.

Lee, S. Y., Lee, D.-Y. and Kim, T. Y. (2005a). Systems biotechnology for strain improvement. *Trends in Biotechnology, 23*, 349-358.

Lee, S. J., Lee, D-Y., Kim, T. Y., Kim, B. H., Lee, J. and Lee, S. Y. (2005b). Metabolic engineering of *Escherichia coli* for enhanced production of succinic acid, based on genome comparison and in silico gene knockout simulation. *Applied and Environmental Microbiology, 71*(12), 7880-7887.

Park, J. H., Lee, K. H., Kim, T. Y. and Lee, S. Y. (2007). Metabolic engineering of *Escherichia coli* for the production of L-valine based on transcriptome analysis and *in silico* gene knockout simulation. *Proc. Nat. Acad. Sci. USA*, *104*(19), 7797-7802.

Ratledge, C. and Kristiansen, B. (2006). *Basic biotechnology.* Cambridge: Cambridge University Press.

Scheper, T., Faurie, R. and Thommel, J. (2003). *Advances in biochemical engineering/biotechnology: microbial production of L-amino-acids, 79*. Berlin, Germany: Springer-Verlag.

Schmid, J. W., Mauch, K., Reuss, M., Gilles, E. D. and Kremling, A. (2004). Metabolic design based on a coupled gene expression-metabolic network model of tryptophan production in *Escherichia coli, Metabolic Engineering, 6*(4), 364-377.

Tarafder, A., Lee, B. C. S., Ray, A. K. and Rangaiah, G. P. (2005). Multiobjective optimization of an industrial ethylene reactor using a nondominated sorting genetic algorithm. *Industrial and Engineering Chemistry Research, 44*(1), 124-141.

Torres, N. V. and Voit, E. O. (2002). *Pathway analysis and optimization in metabolic engineering.* New York: Cambridge University Press.

Vera, J., de Atauri, P., Cascante, M. and Torres, N. V. (2003). Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by *Saccharomyces cerevisiae*. *Biotechnology and Bioengineering, 83*(3), 335-343.

Visser, D., Schmid, J. W., Mauch, K., Reuss, M. and Heijnen, J. J. (2004). Optimal re-design of primary metabolism in *Escherichia coli* using linlog kinetics. *Metabolic Engineering, 6*(4), 378-390.

Vital-Lopez, F. G., Armaou, A., Nikolaev, E. V. and Maranas, C. D. (2006). A computational procedure for optimal engineering interventions using kinetic models of metabolism. *Biotechnology Progress, 22*(6), 1507-1517.

Voit, E. O. (2000). *Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists.* Cambridge: Cambridge University Press.

This page intentionally left blank

Index

a posteriori methods, 8, 154
a priori methods, 9, 154
achievement (scalarizing) function, 163
air separation, 41, 47
alkylation process, 13–18
amino acids, 402, 406, 408
    isoleucine, 408
    L-aspartate, 402
    L-glutamate, 402
    L-lysine, 402
    L-phenylalanine, 402
    methionine, 408
    serine, 403
    threonine, 408
    tryptophan, 402
    tyrosine, 408
analyst, 157
annual cash flow, 308
aspergillus niger, 403
aspiration level, 158
auto refrigeration, 239

benchmark problems, 108, 110
bi-objective optimization, 4
biochemical manufacturing, 402
biotechnology
    systems, 402
biotechnology and food industry, 30
    bioreactor, 38, 39
    column chromatography, 39
    food drying, 37
    gluconic acid production, 37
    membrane filtration, 37
BLEVE, 22
boundary chromosomes, 101
box plot, 113
building protection, 339

capital cost, 238
    compressors, 256
    multi-phase turbine, 245, 256
    plate-fin heat exchanger, 245, 257
    turbo-expanders, 258
cDNA microarray, 363, 364
chemical plants, 339
chemical process systems modeling, 81
chromosome, 94
citric acid, 403
classification, 164
classification-based methods, 164
classifying objective functions, 165
clustering, 371
    agglomerative clustering, 370
    cluster cardinalities, 389
    cluster locations, 378
    divisive clustering, 370
    expression profiling, 367
    hierarchical clustering, 370
    K-means clustering, 370

measure of similarity, 370
supervised clustering, 371
unsupervised clustering, 370
clustering coefficient, 374
composite curves
dual independent expander process, 6, 238
nitrogen cooling process, 253
computational biology
cell, 364
central dogma of molecular biology, 365
DNA, 365
gene, 364
gene expression, 364
replication, 365
RNA, 364, 365
transcription, 365
translation, 365
concordance index, 201
constraint, 4, 408, 413
homeostasis, 408
total enzymatic flux, 408, 410
convex multi-objective optimization, 159
corynebacterium glutamicum, 402
cost analysis, 297
crossover, 97
crossover probability, 97, 252, 262
crowding distance, 101

decision maker, 8, 154, 157
decision making, 339–341
decision space, 341
decision variables, 4, 156
desalination, 41, 47
discordance index, 199, 201
distance matrix, 379
dynamic optimization, 36, 48

economic criteria, 302
discounted payback period, 337
economic performance index, 325
e-constraint method, 68, 73
Edgeworth-Pareto optimal solutions, 5
e-dominance, 84
efficient set of solutions, 339
ELECTRE, 189, 196
elitism, 119
emdollars, 321
emergency planning, 339, 340
emergency response, 342
emergy, 321
enantioseparation, 47
energy gain, 293, 295–298
environment, 339
environment impact, 2, 31
environmental criteria, 302
environmental selection, 352
enzyme
DAHPS, 408, 414, 416, 419, 421, 422
PEPCxylase, 408, 410, 414, 416, 421
SerSynth, 408, 410, 416, 419, 422
epsillon constraint method, 9, 12, 16, 160
escherichia coli, 401, 405, 421, 422
central carbon metabolism, 401
evacuation, 339
evolutionary algorithm, 339
Excel ® solver, 307
expected rate of return, 308

feasible goals method, 170
feasible region, 156

feed optimization, 277
fibroblast-serum data, 389
first principles models, 30
fitness function, 95
fixed capital investment, 308
    six-tenths rule, 336
flowchart, 100
fluidized catalytic cracking, 277
flux
    control, 421
    ratio, 408, 413, 415, 416, 419, 421

gas-phase refrigeration, 21, 237
gene
    expression, 22, 23, 363, 404, 406
    knockout, 404, 410, 412–414, 422
    manipulation, 412, 421, 422
    overexpression, 403, 412
    repression, 403
gene network, 382
    random network, 373
    regular network, 374, 375
    small world network, 375
gene regulatory networks, 81
generating methods, 8
generational distance, 144
genes, 386, 392
genetic algorithm, 9, 20
global optima, 3, 91
gluconic acid, 193
goal programming, 9
graphical representation of network, 392
graphical user interface, 277

hazardous materials, 339
health care, 340
health effects, 343
health impact, 345
heat exchanger
    design, 271
    network, 29, 240
heat recovery system design, 178
hybrid methods, 170
HYSYS, 251

ideal objective vector, 7, 156
IND-NIMBUS, 168
indifferent threshold, 197, 198, 204, 232
indirect interactions, 376
industrial ecosystem (IE), 320
    IE with 4 plants, 320, 337
    IE with 6 plants, 322, 326–333, 337
industrial ethylene reactor, 76
industrial hydrocracking unit, 77
industrial semi-batch nylon-6 reactor, 119
industrial styrene reactor, 76
    adiabatic styrene reactor, 74
    steam-injected styrene reactor, 76
interactive methods, 9, 154, 159
inter cluster distance, 370, 387
interior point optimizer, 38, 174
internal rate of return, 302
intra cluster distance, 370
inverse fluxes, 413, 414

jumping genes (JG), 103

Kalundborg, Denmark, 320
k-means clustering, 140

learning phase, 163
lexicographic ordering, 9
light beam search, 164

linear programming, 13, 35, 43
LNG, 238, 239
low-density polyethylene, 314

major accident, 339
manufacturing, 402
mapping, 95
Markov, 348
Markovian type stochastic model, 347
maximum spread, 110
mean squared error, 140
metabolic pathway
  glycolytic, 413
  pentose-phosphate, 413, 414, 416, 421
  recipe, 402, 403, 413
metabolites, 404–406, 410
methyl ethyl ketone, 41, 46
metrics, 110
microarray technology, 365, 366
  down regulated, 368
  expression profiling, 367
  hybridization, 366
  normalization, 368
  over-expressed, 368
  under-expressed/down-regulated, 368
  up-regulated, 368
minimum temperature driving force, 237, 245
mixed integer multi-objective optimization, 401, 402
model
  non-linear dynamic, 401
  S-system, 403, 404
most preferred solution, 157
multi cluster assignment, 386
multi-criteria optimization, 3
multi-language environment, 240
multi-objective differential evolution, 73, 74
multi-objective evolutionary algorithm, 61, 277, 339, 342
multi-objective genetic algorithm, 64
multi-objective optimization, 3, 61, 189, 302, 339
multi-objective simulated annealing, 107, 108
  aJG, 108
  JG, 108
multiple cluster assignment, 381
multiple criteria, 153, 154
multiple criteria decision making, 154
multiple objectives, 190
multiple solution sets, 5
multi-product cell factory, 23, 401
multi-product microbial cell factory, 401–404
mutation, 97
mutation probability, 97

Nadir objective vector, 7, 157
natural gas liquefaction, 6
neighborhood and archived genetic algorithm (NAGA), 69
net flow method, 30, 37, 189, 194, 196
net present worth, 302
network topology
  clustering co-efficient, 374, 375
  degree, 373, 374
  degree distribution, 373, 375, 393
  path length, 374, 375, 384
niched-Pareto genetic algorithm, 65
NIMBUS method, 170
non-dominated solutions, 5
non-dominated sorting genetic algorithm, 64, 303
  penalty function method, 330

non-dominated sorting genetic algorithm-II (NSGA-II), 67, 108, 109, 238, 240, 251, 381
aJG, 302, 303
mJG, 104
saJG, 104, 105
sJG, 104, 105
non-inferior solutions, 5
no-preference methods, 8, 154

objective function, 155
conflicting, 2, 3, 48
objective value, 156
objective vector, 156
optimal Pareto
multiple solutions, 192, 254
trade-off, 285
optimization, 1
outranking matrix, 199, 200, 202

paper making, 176, 178
Pareto-archived evolution strategy, 67
Pareto-domain, 189, 192, 194, 198, 200, 206
Pareto-optimal, 302, 310, 392
front, 6, 16, 18
segment, 416, 419, 421
set, 290, 343, 401, 422
Pareto-optimality
global, 157
local, 157
Pareto-optimal front, 6, 238
Pareto-optimal solutions, 5
payback period, 302
Pearson correlation matrix, 384
Pearson's correlation coefficient, 390
petroleum refining and petrochemicals, 40
fluidized bed catalytic cracking, 42
fuel oil blending, 40, 45
hydrocracking, 45
hydrogen production, 42
naphtha catalytic reforming, 40, 45, 70
Parex process, 44
steam reformer, 40, 42
styrene, 41, 43
Phillips optimized cascade process, 239
polymerization, 48, 314
catalytic esterification, 50
continuous casting process, 48, 49
continuous tower process, 49
copolymerization, 48, 50
emulsion homo-polymerization, 49
epoxy polymerization, 50
free radical polymerization, 49
number average molecular weight, 49–51
polydispersity index, 50
polynomial mutation, 137
precursors, 405, 414, 416
preference information, 154, 156, 157, 161, 162, 165
preference-based methods, 8
preference threshold, 197–199, 232
pressure swing adsorption, 30, 120
PRICO process, 239
process design and operation, 20, 29, 30
batch plant, 33, 38
cyclic adsorption, 32
cyclone separator, 31
fluidized bed dryer, 31
froth floatation circuits, 29, 34
NIMBUS, 30, 35, 38

supply chain networks, 33
venturi scrubber, 29, 32
waste incineration, 34
profit before taxes, 303, 308
progress ratio, 288
PROMETHEE, 189
protein and structure prediction, 80
pseudo-stationary, 403, 404
pulping process, 41, 46

radial basis functions, 138
random networks, 373, 375
random number, 94
ranking, 189
real-coded NSGA-II, 50, 119
reference point, 158, 165
reference point method, 163
refrigeration
cascade refrigeration, 237, 238
compression refrigeration, 241
dual independent expander process, 238
gas-phase refrigeration, 237, 240
single mixed refrigeration, 237
regulation
allosteric, 405
cascade, 405
combined feedback-feedforward, 405
feedback, 405
feedforward, 405
negative (inhibiting), 405
positive (activating), 405
regulators (without sign), 405
rough set method, 21, 46, 189, 194
roulette wheel selection, 96
rules, 189

saccharomyces cerevisiae, 39, 402– 404
bio-fuel, 402
ethanol, 403, 404
satisficing trade-off method, 165
scalarization, 157
scalarizing function, 8, 157, 162
scatter plot, 392, 393
scheduling, 30, 36
seed, 247, 248
semi-batch reactive crystallization process, 78
sequence and structure alignment, 80
set coverage metric, 110
simple genetic algorithm, 93
simple simulated annealing, 106
simply pareto solutions, 5
simulated binary crossover, 136
simulated moving bed (SMB), 46, 79, 172
chiral separation, 33, 46
DNA purification, 39
evolutionary multi-objective optimization in VARICOL process, 33, 37
hydrolysis, 41, 47, 80
isomerization, 38
pseudo SMB, 36, 45
single cluster assignment, 381
small world networks, 375, 383
small world phenomena, 375
socio-economic costs, 343
solution process, 157
Solver tool in Excel, 16
spacing, 110
steam consumption, 22, 293
strength Pareto evolutionary algorithm (SPEA), 35, 66
strength Pareto evolutionary algorithm 2 (SPEA2), 66, 83

sugar separation, 173
superstructure, 173
surrogate assisted evolutionary algorithm, 131, 135
surveys of methods, 154
sustainable development, 303, 321

temperature, 4, 5
temperature cross, 250, 263
test problems
  alkylation process optimization, 13, 146
  OSY, 145
  TNK, 134, 146
  ZDT, 142
threshold, 189
time value of money, 308
total annual cost, 2, 308
total capital investment, 309
tournament selection, 65, 102
toxic release, 345, 346
trade-off, 157
transcription factor, 371, 376
translation, 365, 376

value function methods, 9
varicol process, 33, 37
vector evaluated genetic algorithm, 63
veto threshold, 179, 199
Vitamin C, 41, 47

waste incineration plant, 34, 81
water allocation problem, 21, 176
weak Pareto optimality, 156
weighting method, 9, 12, 158
weight, 189, 385
Williams-Otto process, 5, 304
WWW-NIMBUS, 168

ZDT2 problem, 108, 110
ZDT3 problem, 108, 110
ZDT4 problem, 93, 108, 110, 119