MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION: MODIFICATIONS AND APPLICATIONS TO CHEMICAL PROCESSES

SHIVOM SHARMA

(M. Tech., Indian Institute of Technology, Roorkee, India)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

To My Family

&

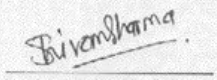
Lord Shiva

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in this thesis.

This thesis has not been submitted for any degree in any university previously.



Shivom Sharma

13 August 2012

Acknowledgements

In the last four years, many people have been an integral part of my research and their support and good wishes helped me to reach the finishing line. I feel extremely blessed to be the doctoral student of Professor G. P. Rangaiah of the Department of Chemical and Biomolecular Engineering at the National University of Singapore. He has been an extraordinary supporter during the entire journey of my research. I find him very simple and organized person who is always ready to help his students. I consider myself fortunate that I got the opportunity to work in the area of my interest under his supervision. His invaluable knowledge, advices and continuous guidance always encouraged me to analyze the problems and look for solutions.

This thesis would have not been possible without the army of lab mates and final year students. I feel grateful to Haibo for his comprehensive discussions and providing alternate suggestions. I highly value the support of Anton, Aseem, Yung Chuan, Frankie, Gim Hoe, Hao Meng, Hua Qiang, Kai Ling, Luke, Luo Koi and Zi Chao. I am thankful to my colleagues Seyed, Suraj, Krishna, Vaibhav, Naviyn, Sumit, Chi, Jorge, Bhargava, Shruti and Wendou for their help and keeping the lab environment energetic. I would like to thank Professor A. Bonilla-Petriciolet from the University of Guanajuato, Mexico, for the thought provoking discussions.

Special thanks goes to Mr. K. H. Boey and Ms. Samantha Fam for taking care of lab related issues, and also to Doris, Hui Ting and Vanessa for taking care of academic and administrative matters. I would like to thank the National University of Singapore for providing the funding and research facilities.

I would also like to acknowledge moral support of my friends Srinath, Abhishek, Amit, Sunand, Shilpi, Shailesh, Manoj, Naresh, Praveen, Lakshmi, Shashi, Varun,

Pransnna, Thaneer, Rajnish, Pankaj, Shyam, Saurabh, Vikas and Atul. Special credit goes to Sumit, Ashwini and KMG for those memorable conversations during meals and tea breaks.

Most important for me is to evince my endless gratitude to my parents for their everlasting love and support. I further want to convey my heartfelt thanks to Mr. V. K. Sharma, Shivdutt Sharma, Rajneesh Sharma, Dinesh Sharma, Nakul and all other relatives and friends. A big thanks to my sisters and brother for their love and cherished moments, it helped in keeping the sprit. Finally, by no means the least, I want to thank Vaishali for her support.

I dedicate this thesis to my maternal grandfather Mr. M. P. Sharma for his immense belief in me.

Shivom Sharma

August, 2012

Table of Contents

Summary

Industrial problems are complex and often have multiple conflicting objectives. Multi-objective optimization (MOO) helps to explore the trade-offs among different objectives. There are several stochastic MOO techniques but suitable modifications to them are required for more effective solution of application problems. This study improves multi-objective differential evolution (MODE) in key aspects such as search termination based on the improvement in non-dominated solutions obtained with generations, better exploration of search space using taboo list, and handling of equality constraints by dynamically relaxing them. The improved/integrated MODE (I-MODE) algorithm has been tested on many benchmark functions and then used to solve chemical engineering application problems.

First, current MOO techniques and their use in optimizing chemical engineering applications are reviewed. Next, several performance metrics for MOO problems are modified and their variations with generations have been assessed on test functions. Variance in the values of two selected performance metrics, obtained in recent generations, is checked individually, and it is proposed to terminate the search if the improvement in both metrics is statistically insignificant. The developed I-MODE includes DE with taboo list (DETL) for multiple objectives, self adaptation of algorithm parameters, improvement-based termination criterion and taboo list to record and avoid recently visited search regions. Use of a suitable termination criterion (instead of maximum number of generations) and taboo list improves efficiency and reliability of the search algorithm. It has been implemented in MS-Excel and Visual Basic for Applications (VBA). I-MODE algorithm is tested on several constrained benchmark MOO problems, and its performance is compared with

best the algorithm (namely, DMOEA-DD) in IEEE Congress on Evolutionary Computation 2009.

Effectiveness of the proposed termination criterion is tested with the elitist non-dominated sorting genetic algorithm, on several MOO benchmark functions. Additionally, I-MODE is combined with a deterministic method for obtaining accurate optimal solutions quickly; for this, I-MODE search is terminated using the proposed termination criterion, and then normalized normal constraint (NNC) method is used to precisely find the optimum. Further, I-MODE algorithm has been evaluated on the alkylation, Williams-Otto and fermentation processes.

In general, feasibility approach works well for solving problem with inequality constraints. It may not be effective for solving problems with equality constraints, as feasible search space is extremely small for them. For this, all constraints are dynamically relaxed, which makes certain individuals temporarily feasible during selection of individuals for the next generation in the I-MODE algorithm. The adaptive constraint relaxation with feasibility approach is tested on two MOO benchmark problems with equality constraints, and then applied to optimize two fermentation processes.

A three-stage fermentation process integrated with cell recycling and pervaporation for bio-ethanol is modeled and optimized for multiple objectives, using MODE and I-MODE. Improvements in the performance of the fermentation process, after integrating with pervaporation and extraction unit, are compared. The obtained non-dominated solutions in one optimization case are ranked using the net flow method. Subsequently, a bio-diesel production process, using waste cooking oil as the feed, is developed, simulated and optimized for environmental and economic

objectives. Finally, a membrane distillation module and a desalination process are optimized for water production rate and energy consumption simultaneously, using I-MODE algorithm.

The modifications made in MODE to develop I-MODE algorithm are useful for solving MOO application problems. The studied applications and findings in this thesis are of particular interest because of increasing demand for renewable energy and drinking water.

List of Tables

List of Figures

Nomenclature

Acronyms

AACV	average absolute constraint violation
ACO	ant colony optimization
ACM	Aspen Custom Modeler
ACRFA	adaptive constraint relaxation and feasibility approach
BB	branch and bound
BD	bio-diesel
CEC	congress on evolutionary computation
COM	cost of manufacturing
CORL	combined objective repeated line search
CSTR	continuous stirred tank reactor
DCMD	direct contact membrane distillation
DE	differential evolution
DETL	differential evolution with taboo list
DMOEA-DD	dynamic multi-objective evolutionary algorithm with domain decomposition
DV	decision variables
EA	evolutionary algorithm
ELECNRTL	electrolyte NRTL
FA	feasibility approach
FAME	fatty acid methyl esters
FCI	fixed capital investment
FFA	free fatty acid
FO	forward osmosis
GA	genetic algorithm
GAMS	general algebraic modeling system
GD/ GD^t	modified/ true generational distance
GD^t_{max}	max GD value obtained using NSGA-II-JG after 100 generations
GDE	generalized differential evolution
GRG	generalized reduced gradient
GT	generation of search termination

HE	heat exchanger
HV/ HVR	hyper volume/ hyper volume ratio
IDE	integrated differential evolution
IGD/ IGD^t	modified/ true inverse generational distance
IGD^t_{max}	max IGD value obtained using NSGA-II-JG after 100 generations
I-MODE	integrated multi-objective differential evolution
JG	jumping gene
LINGO	linear interactive general optimizer
MCr	membrane crystallizer
MD	membrane distillation
MINLP	mixed integer non-linear problem
MNG	maximum number of generations
MO	multi-objective
MODE	multi-objective differential evolution
MODE-TL	multi-objective differential evolution with taboo list
MOEA	multi-objective evolutionary algorithm
MOGA	multi-objective genetic algorithm
MOO	multi-objective optimization
MOSADE	MO self adaptive DE with elitist archive and crowding entropy-based diversity
MSDM	multi-objective steepest descent method
NBI	normal boundary intersection method
NFE	number of function evaluations
NFM	net flow method
NLP	non-linear problem
NNC	normalized normal constraint method
NPGA	niched Pareto genetic algorithm
NPW	net present worth
NRTL	non-random two liquid model
NSGA	non-dominated sorting genetic algorithm
NSR	number of successful runs
NT	number of solution in the true/known Pareto front
PBP	payback period

PBT	profit before tax
PDM	Pareto descent method
PF	Pareto-optimal front
PM	performance metric
PSO	particle swarm optimization
RO	reverse osmosis
RSM	rough set method
SA	simulated annealing
SOO	single objective optimization
SP/ SP^t	modified/ true spread
SPEA	strength Pareto evolutionary algorithm
SQP	sequential quadratic programming
SR	success rate
TACV	total absolute constraint violation
TC	termination criterion
TL	taboo list/ tolerance limit
TLS	taboo list size
TR	taboo radius
TS	taboo search
UL	utopia line
UNIQUAC	universal quasi chemical
VBA	visual basic for applications
WS	weighted sum
ZDT	Zitzler-Deb-Thiele

Symbols

a	permeation coefficient [m^3/h]
A_M	membrane surface area [m^2]
b	vector of constant for inequality constraints
b_k	bleed ratio for k^{th} stage
c	user defined parameter for self-adaptation
C_{BM}	bare module cost [$]
c/ C	concentration/ specific heat of sea water [g/lit, J/(kg.K)]

Cp	purchase cost [$]
Cr	crossover probability
C_{TM}	total module cost [$]
d	Euclidean distance
D	dilution rate [1/h]
d_i/ d_o	internal/ external membrane fiber diameter [m]
$D_{k,i/\ o}$	inlet/ outlet dilution rate for k^{th} stage fermentor [1/h]
e_m	extreme/ boundary solutions for m^{th} objective
F	mutation rate
f/ F	objective function/ objective function vector
F/ P	sea water/ permeate stream
f_b	arbitrary number used with aJG operator
g/G	vector of inequality constraints
G	generation number/ glycerol
h	vector of equality constraints
h_f	shell-side heat transfer coefficient [W/(m^2.K)]
h_m	membrane heat transfer coefficient [W/(m^2.K)]
h_p	tube-side heat transfer coefficient [W/(m^2.K)]
k_m	membrane mass transfer coefficient [kg/(m^2.h.Pa)]
l_{string}	number of bits for each decision variables
M	number of objective functions
max.	maximization
min.	minimization
n	individual number/ number of non-dominated solutions obtained
N	population size
n_d	number of individuals dominating an individual
n_e	number of equality constraints
n_i	number of inequality constraints
$N_{M,k}$	number of pervaporation units used with k^{th} stage fermentor
Nv	water vapor flux through membrane [kg/(m^2.h)]
nw	number of weights
p	probability to support χ^2-test hypothesis
p_c	crossover probability

$p_{e,k}$	ethanol concentration in mother liquor after k^{th} stage separator [kg/m^3]
p_{fm}/ p_{pm}	water vapor partial pressure [kPa]
p_{JG}	jumping gene probability
p_k	ethanol concentration in k^{th} stage fermentor [kg/m^3]
p_m	mutation probability
P	membrane permeability [m/h]
P_k	mother liquor flow rate to k^{th} stage pervaporation units [m^3/h]
P_m	preference threshold for m^{th} objective
$P_{Mi/\ o,k}$	inlet/ outlet ethanol concentration (k^{th} stage) on sweep fluid side of pervaporation units [kg/m^3]
q	flow rate of stream [m^3/h]
$q_{k,o}$	mother liquor flow rate from k^{th} stage to $(k+1)^{th}$ stage [m^3/h]
$q_{Mi/\ o,k}$	inlet/ outlet sweep fluid flow rate for k^{th} stage pervaporation units [m^3/h]
Q_m	indifference threshold for m^{th} objective
randc	random number from Cauchy distribution
randn	random number from Normal distribution
$r_{p/\ sg/\ sx/\ x,k}$	ethanol production rate/ glucose consumption rate/ xylose consumption rate/ cell mass growth rate in k^{th} stage fermentor [$kg/(m^3.h)$]
R_j	penalty weight for j^{th} inequality constraint
s, s1, s2	individual from set of non-dominated solutions (S, S1, S2)
S, S1, S2	set of non-dominated solutions obtained
$S_{CR/F}$	set of successful Cr/ F values
$s_{f,k}$	substrate concentration in feed to k^{th} stage fermentor [kg/m^3]
$s_{ge,k}$	glucose concentration in mother liquor after k^{th} stage separator [kg/m^3]
$s_{g,k}$	glucose concentration in k^{th} stage fermentor [kg/m^3]
s_T	total sugar supply [$kg/(m^3.h)$]
$s_{xe,k}$	xylose concentration in mother liquor after k^{th} stage separator [kg/m^3]
$s_{x,k}$	xylose concentration in k^{th} stage fermentor [kg/m^3]
T	temperature of reactor [^{0}C or K]
T_f/ T_p	temperature of sea water/ permeate [^{0}C or K]
x	vector of decision variables
$x_{e,k}$	cell mass conc. in mother liquor after k^{th} stage separator [kg/m^3]

x_k	cell mass concentration in k^{th} stage fermentor [kg/m^3]
$\mathbf{x^L}$	vector of lower bounds on decision variables
$\mathbf{x_{r0}/x_{r1}/x_{r2}}$	randomly selected individuals for reproduction
$\mathbf{x^U}$	vector of upper bounds on decision variables
u	trial vector
U	Techebycheff utility function
v	mutant vector
vh	volume of hypercube
V	volume of reactor or process unit [m^3]
V_f/ V_p	volumetric flow rate of sea water/ permeate [m^3/h]
V_F	volume of k^{th} stage fermentor [m^3]
V_m	veto threshold for m^{th} objective
$V_{M,k}$	total volume of k^{th} stage pervaporation units [m^3]
w_k	set of equidistant weights
w_m	weight for m^{th} objective function
Z_{NaCl}	mass fraction of salt

Greek letters

$\delta_{GD/\ SP}$	threshold value of variance for GD/ SP
ΔH_v	heat of vaporization of water [J/kg]
ε_+	additive epsilon indicator
ε_m	user specified bound for m^{th} objective in ε-constraint method
$\zeta_{p/\ s/\ x}$	ethanol condensed/ substrate condensed/ cell discard factor
η	purge fraction
λ	generations used to check variance in GD and SP values/ fraction of glucose in substrate
$\mu_{CR/\ F}$	mean CR/ F values
μ_G	constraint relaxation value for G^{th} generation
$\mu_{IGD^t\ /\ GD^t\ /\ SP^t\ /\ NFE}$	mean value of IGD^t/ GD^t/ SP^t/ NFE over 30 runs
π_k	ethanol productivity for k^{th} stage [$kg/(m^3.h)$]
ρ	ethanol density/ density of reaction mixture [kg/m^3]
$\sigma_{IGD^t/\ GD^t}$	standard deviation of IGD^t/ GD^t over 30 runs
$\chi_{g/\ x,k}$	glucose/ xylose conversion in k^{th} stage

Chapter 1

Introduction

1.1 Multi-objective Optimization

Optimization is the process of finding the best possible solution for a given problem. The goal of an optimization method or technique is to find the values of decision variables which can maximize or minimize the value of a performance criterion (i.e., objective function) and also satisfy (process) constraints. Optimization has been fruitfully applied to improve the performance in diverse areas such as science, engineering and business. Many optimization techniques have been used as quantitative tools to improve the performance of chemical processes (Edgar et al., 2001; Ravindran et al., 2006; Rangaiah, 2009a).

Profit is the most commonly used criterion for assessing the performance of many chemical processes. However, in most of the application problems, there are a number of objective functions (e.g., economic criteria, environmental criteria and safety), and these are often conflicting or partially conflicting in nature. Multi-objective optimization (MOO) is used to find the trade-off among different objectives. A MOO optimization problem, with M number of objectives, can be mathematically described as follows.

Min.	$\{f_1(\mathbf{x}), f_2(\mathbf{x}),... f_M(\mathbf{x})\}$	(1.1a)
Subject to	$\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U$	(1.1b)
	$\mathbf{h}(\mathbf{x}) = \mathbf{0}$	(1.1c)
	$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$	(1.1d)

Here, **x** is the vector of decision variables, and $\mathbf{x}^L$ and $\mathbf{x}^U$ are respectively vectors of lower and upper bounds on decision variables. **g** and **h** are set of inequality and equality constraints, respectively. A set of non-dominated solutions (known as Pareto-optimal front) can be obtained after solving the above MOO problem. Figure 1.1 shows such solutions for a MOO problem having 2 objective functions. Each non-dominated solution is better in one objective and also worse in the other objective when compared to the rest of the non-dominated solutions.

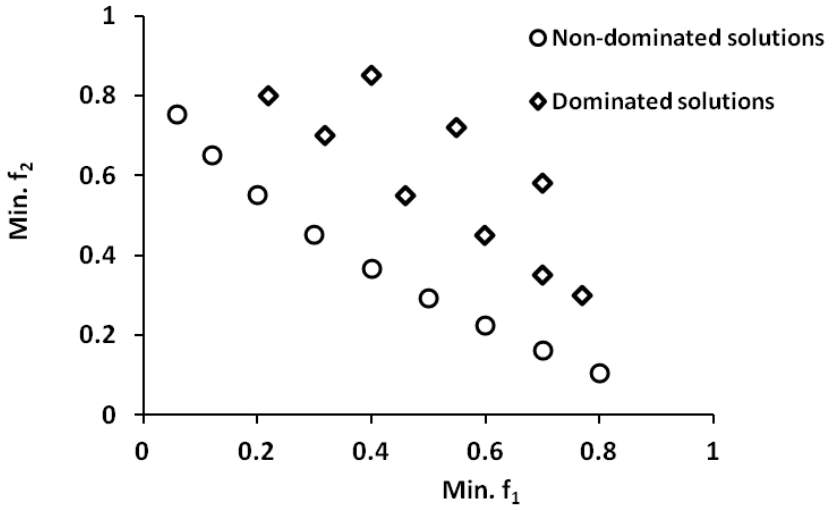


Figure 1.1: Pareto-optimal front for a two-objective optimization problem

1.2 Classification of MOO Methods

Optimization methods can be classified into two types, namely, deterministic and stochastic. Deterministic methods require derivatives of objective functions and constraints, and so these can only be applied to solve optimization problems with continuous objective functions and constraints. These methods are time-efficient and locate optimum exactly, but they may not able to solve optimization problems having discontinuous and non-smooth objective and constraints. Conversely, stochastic methods can locate the global optimum with high reliability, but they may require more computational effort. Additionally, stochastic methods can be applied to black-

box optimization problems, whose explicit equations and their characteristics are not available.

MOO methods can be classified into two broad categories: 1) Pareto generating methods - many non-dominated solutions are generated, and 2) Preference based methods - decision maker provides preference before or during optimization (Figure 1.2).

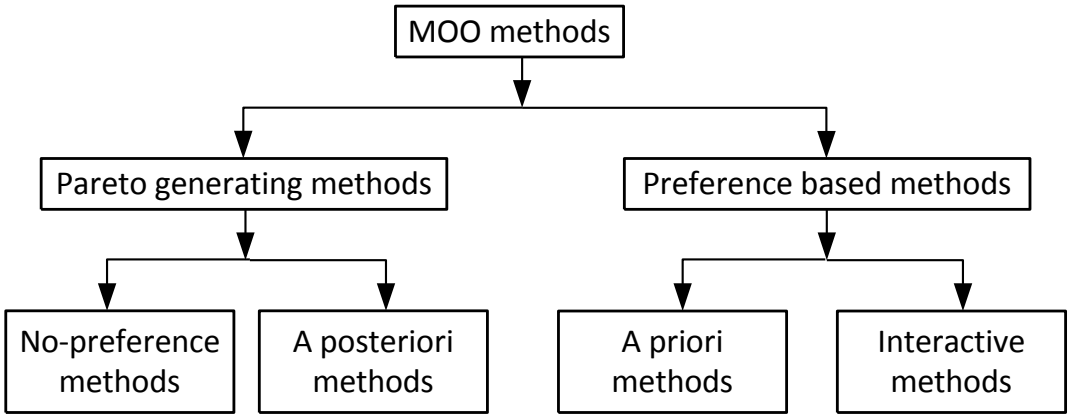


Figure 1.2: Classification of MOO techniques

Pareto generating methods are further divided into two categories, namely, no-preference methods and a posteriori methods. In the no-preference methods, few non-dominated solutions can be obtained using different metrics; one such method is the global criterion method. A posteriori methods either generate Pareto-optimal front using scalarized objective function or multi-objective approach. The scalarized single objective optimization (SOO) problem can be solved using a suitable method. Weighted sum and ε-constraint methods are two classical methods for solving MOO problem as SOO problem, and these can generate one single non-dominated solution in each run. In the weighted sum method, some scalar weight is assigned to each objective. ε-constraint method optimizes the MOO problem for the most important objective function, while other objectives are considered as additional constraints in

the SOO problem. MOO methods, like non-dominated sorting genetic algorithm-II (NSGA-II), multi-objective differential evolution (MODE) and multi-objective particle swarm optimization (MO PSO) can generate the complete Pareto-optimal front in a single run.

Preference based methods are also divided into two categories, namely, a priori methods and interactive methods. A priori methods require preference of objective functions before the optimization starts. For example, goal programming uses minimax type formulation to accommodate preference of the decision maker, and solves the MOO problem as a SOO problem. Finally, NIMBUS (Miettinen, 1999) is an interactive method, which requires preference of the decision maker during optimization.

1.3 Motivation and Scope of Work

There are a number of stochastic MOO techniques in the literature, but there is scope to improve their efficiency and reliability for solving application problems. In this thesis, multi-objective differential evolution (MODE) is improved in the following aspects.

- Improving efficiency of stochastic search by terminating search at the right generation.
- Locating global optimum with high reliability for application problems.
- Reducing number of function evaluations for computationally expensive problems.
- Effective handling of equality constraints often present in application problems.

MODE, a simple and powerful stochastic search algorithm (Zhang et al., 2009), is improved to address the above issues, and these improvements are tested on

benchmark and Chemical Engineering application problems in the literature. Further, bio-ethanol process, bio-diesel plant and membrane distillation system are modeled, simulated and then optimized for multiple objectives. The motivation for studying above issues, along with background information, is briefly review below.

1.3.1 *Improved MODE with Termination Criterion*

Maximum number of generations is the most common termination criterion in evolutionary algorithms used for solving MOO problems. Solving an optimization problem may require less or more computational effort that cannot be identified based on the optimization problem characteristics, such as number of decision variables, objectives and constraints. For optimal use of computational resources, termination of stochastic search at the right generation is necessary. Here, a search termination criterion, using the non-dominated solutions obtained in the recent generations, is developed and tested. In many applications, evaluation of objective functions and constraints is computationally expensive, as complex process model equations have to be solved. This study uses taboo list with MODE to avoid revisits and for better exploration of the search space (Srinivas and Rangaiah, 2007). Further, different problems require different values of algorithm parameters, and hence these are self-adapted in the developed MOO algorithm. In summary, the improved MODE (I-MODE) algorithm has taboo list, termination criterion and self-adaptation of algorithm parameters.

1.3.2 *Use of Termination Criterion with NSGA-II and NNC*

NSGA-II and its jumping gene adaptations have been used to optimize many process design and operation problems. The developed termination criterion has been used to check convergence of NGSA-II with four jumping gene adaptations on several

test functions. In order to improve the search efficiency without losing search reliability, stochastic and deterministic search methods are combined together. Normalized normal constraint (NNC) method (Messac et al., 2003) is used as to refine the non-dominated solutions obtained using I-MODE algorithm, and termination criterion is used to decide the switching of search from I-MODE to NNC.

1.3.3 Evaluation of Termination Criterion on Application Problems

Solutions are not known in advance for application optimization problems, and so making a decision on the search termination is difficult. In order to evaluate the effectiveness of the proposed termination criterion on application problems, I-MODE algorithm is used to optimize alkylation, Williams-Otto and fermentation processes, and the non-dominated solutions obtained are compared with the Pareto-optimal fronts obtained using the maximum number of generations.

1.3.4 Improved Constraint Handling Technique for MOO

Constraints besides bounds are frequently present in MOO application problems. Penalty function and feasibility approaches are commonly used for handling constraints in stochastic MOO methods. Feasibility approach gives higher priority to feasibility of the solution over objective function value, and performs well on optimization problems with inequality constraints. Feasible search space is extremely small for problems with equality constraints. Therefore, feasibility approach is not effective to solve such problems. Adaptive constraint relaxation with feasibility approach addresses this issue by dynamically relaxing the limits on different constraints. In this thesis, adaptive relaxation of constraints with feasibility approach is modified for solving constrained MOO problems.

1.3.5 Modeling and Optimization of Bio-ethanol, Bio-diesel and Membrane Distillation Processes

These selected applications are of particular interest because of increasing demand for renewable energy and drinking water. Bio-ethanol and bio-diesel are two main liquid bio-fuels, and they have lower environmental impact compared to fossil fuels. Desalination of sea water is essential for addressing water scarcity in many regions of the world.

- Ethanol concentration inside the fermentor inhibits conversion of fermentable sugars to ethanol, which leads to low yield and productivity. Ethanol can be removed from the fermentor by using extraction or pervaporation. In this work, a three-stage bio-ethanol process integrated with cell recycling and pervaporation is modeled and optimized for multiple objectives, using MODE and I-MODE algorithms. Performance of the three-stage fermentation process integrated with pervaporation is compared with that integrated with extraction.
- Waste cooking oils have significant impact on the environment, and so their use to produce bio-diesel is attractive for both economic and environmental reasons. The present study optimizes the design of a bio-diesel plant for three important objectives (maximum profit, minimum fixed capital investment and minimum organic waste), using MODE + taboo list and I-MODE algorithms. Further, one process design is selected, and then studied for variation in waste cooking oil flow rate.
- Membrane distillation (MD) is a thermally driven process, where low-grade waste heat or renewable energy can be used to produce drinking water. Here, a MD system is modeled, simulated and then optimized for multiple objectives.

1.4 Outline of the Thesis

This thesis has ten chapters in total. The next chapter reviews popular stochastic and deterministic methods for solving MOO problems. It also reviews recent applications of MOO in Chemical Engineering. Chapter 3 describes the development of I-MODE algorithm in detail. Performance metrics, their modifications and variations with generations on the selected test functions are also presented in this chapter. In Chapter 4, the developed termination criterion is used with jumping gene adaptations of NSGA-II and NNC methods. I-MODE algorithm is used to optimize alkylation, Williams-Otto and fermentation processes in Chapter 5. Chapter 6 discusses an equality constraint handling technique for constrained MOO problems.

Chapter 7 models and optimizes a three-stage fermentation process integrated with cell recycling and pervaporation. Performance of pervaporation and extraction with fermentor, to remove ethanol, are quantitatively compared in this chapter. In Chapter 8, a bio-diesel plant using waste cooking oils is developed, simulated and then optimized for three important objectives. Similarly, a membrane distillation system for producing pure water from sea water is modeled, simulated and optimized in Chapter 9. The last chapter of this thesis provides conclusions of this work and recommendations for future works.

Chapter 2

Literature Review

2.1 Introduction

Both deterministic and stochastic MOO techniques have been used to solve optimization problems. Weighted sum, ε-constraint, normal boundary intersection and normalized normal constraint methods are commonly used deterministic methods for solving MOO problems. Stochastic methods are mostly inspired by natural phenomena, and many of them employ a population of trial solutions. Evolutionary algorithms are inspired by the evolution of different species. They offer robust and adaptive search mechanisms based on the rules of selection, recombination, mutation and survival. Ant colony optimization and particle swarm optimization are meta-heuristic searches inspired by social behavior of swarms. Simulated annealing, taboo search and differential evolution are other prominent meta-heuristics for solving optimization problems. Originally, above stochastic algorithms are proposed for solving single objective optimization (SOO) problem; later, these are adapted for solving MOO problems. Table 2.1 lists popular stochastic optimization algorithms proposed for SOO problems.

This chapter briefly reviews MOO techniques and their applications in Chemical Engineering. In addition, many of the subsequent chapters contain a brief review of relevant papers in the Introduction section. The next section of this chapter discusses deterministic optimization methods, whereas third section covers the development of stochastic techniques for solving MOO problems. Section 4 describes some recent

applications of MOO in Chemical Engineering. Finally, conclusions from this chapter are summarized in the last section.

Table 2.1: Popular stochastic optimization algorithms and development timeline

Algorithm	**Proposed by**
Genetic algorithm (GA)	Holland (1975)
Simulated annealing (SA)	Kirkpatrick et al. (1983)
Taboo/tabu search (TS)	Glover (1986)
Particle swarm optimization (PSO)	Kennedy and Eberhart (1995)
Differential evolution (DE)	Storn and Price (1995)
Ant colony optimization (ACO)	Dorigo and Gambardella (1997)

2.1 Deterministic Methods for Solving MOO Problems

Although stochastic algorithms have been commonly applied to solve MOO problems, deterministic methods are also used by some researchers for solving these problems. Following sub-sections briefly describe main deterministic methods.

2.2.1 *Weighted Sum Method*

In weighted sum (WS) method, M number of objectives are scalarized into a single objective, as follows.

$$\text{Min.} \quad f(\mathbf{x}) = \sum_{m=1}^{M} w_m . f_m(\mathbf{x}) \tag{2.1a}$$

$$\text{Subject to} \quad \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \tag{2.1b}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \text{ and } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \tag{2.1c}$$

A set of weights is used to generate a series of SOO problems, $w_m \in [0, 1]$. Further, sum of weights, for each SOO problem, is equal to one (i.e., $\sum_{m=1}^{M} w_m = 1$). Solution of each SOO problem gives one Pareto point. Figure 2.1 shows the Pareto-optimal points obtained for ZDT1 test function (Zitzler et al., 2000), using WS method. These

points are obtained with equidistance weights [A ≡ (0.7, 0.3), B ≡ (0.65, 0.35), …, I ≡ (0.3, 0.7)]. It can be seen that Pareto-optimal points are not evenly distributed with equidistance weights. Although weighted sum method is intuitive, its disadvantages include selection of suitable weights and the need to solve many SOO problems.

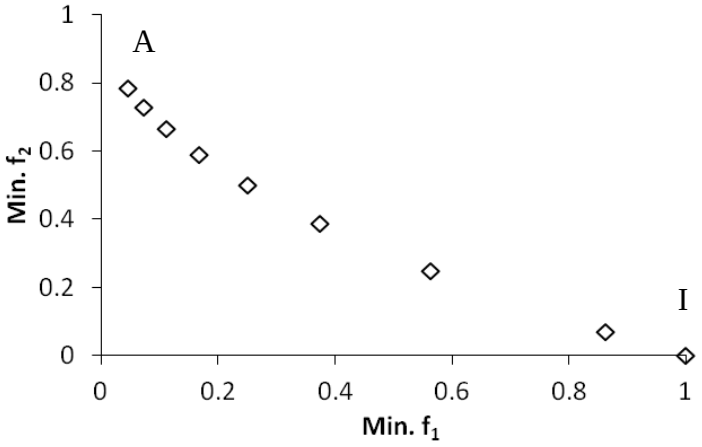


Figure 2.1: Pareto-optimal front for ZDT1 test function using weighted sum method

2.2.2 *ε-Constraint Method*

ε-constraint method solves MOO problem as SOO problem for the most important objective, while considering the remaining objectives as additional inequality constraints in the problem formulation. The SOO problem is solved repeatedly for different user specified bounds on the additional inequality constraints (i.e., ε-vector), in order to obtain Pareto-optimal points.

$$\text{Min.} \quad f_{m'}(\mathbf{x}) \tag{2.2a}$$

$$\text{Subject to} \quad \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \tag{2.2b}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \text{ and } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \tag{2.2c}$$

$$\mathbf{f}_m(\mathbf{x}) \leq \boldsymbol{\varepsilon}_m, \qquad m \neq m' \tag{2.2d}$$

Here, $f_{m'}$ is the objective function, and $\mathbf{f}_m$ ($m \neq m'$) are the additional inequality constraints in the problem formulation.

In Figure 2.2, Pareto points are obtained by solving the ZDT1 test function for second objectives, while first objective is converted into an inequality constraint [A ≡ ($f_1 \leq 0$), B ≡ ($f_1 \leq 0.1$)… K ≡ ($f_1 \leq 1$)]. Similar to weighted sum method, it is difficult to obtain evenly distributed Pareto-optimal front with equidistance ε-vector, and also requires solution of many SOO problems.

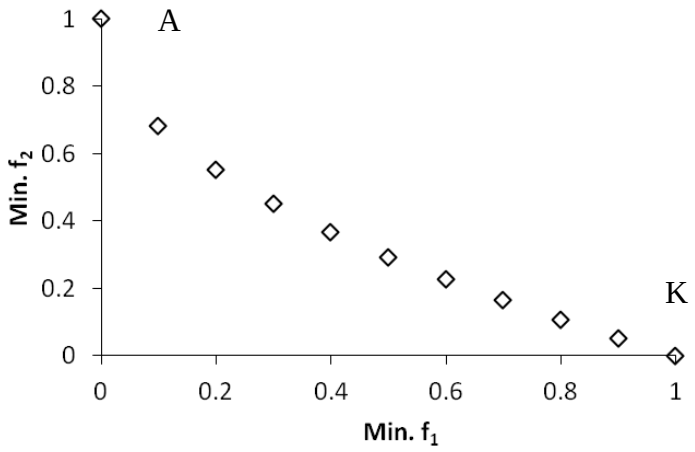


Figure 2.2: Pareto-optimal front for ZDT1 test function using ε-constraints method

2.2.3 *Other Methods*

Weighted sum and ε-constraint methods cannot accommodate preferred values for different objectives (Deb, 2001). Some methods, like goal programming and compromise programming can accommodate preference of decision maker. Here, the desirable solution is the one which gives the smallest difference between different objectives and their respective goals. Newton method, Pareto descent method (PDM), normal boundary intersection (NBI) and normalized normal constraint method (NNC) are other deterministic methods to solve MOO problems (Harada et al., 2006; Das and Dennis, 1996; Messac et al., 2003).

Newton method has been extended to solve unconstrained MOO problems, but the objective functions should be convex and twice differentiable, to calculate Hessian

matrix (Fliege et al., 2009). PDM can be used as a local search method; it is efficient in improving solution near to the search boundaries. PDM finds the feasible Pareto descent direction by solving linear programming problems, and search moves in the descent direction. Multi-objective steepest descent method (MSDM; Fliege and Svaiter, 2000) and combined objective repeated line search (CORL; Bosman and Jong, 2005) also work on the similar principle. These deterministic search methods require continuous and smooth objective functions and constraints.

NBI is independent of scales of objectives, and can produce uniformly distributed Pareto points. It can work with inexact or approximate Hessian using first order derivatives. If the first order derivatives of the objective functions with respect to decision variables do not exist at each point in the objective domain (discontinuous or non-smooth function), then NBI method may not be suitable to solve this type of optimization problems. NNC method can be used for optimization problems with discontinuous Pareto-optimal front; it does not assign any weights to different objectives, rather includes some additional inequality constraints in the problem formulation. If the MOO problem has non-convex search space, NNC method will not give solutions from the global Pareto-optimal front. NNC method is described in more detail in Chapter 4.

2.3 Stochastic Methods for Solving MOO Problems

Although stochastic techniques are time consuming, they are widely applied to solve MOO problems due to their ability to provide many Pareto-optimal solutions in one run and to locate the global optimum. Generally, stochastic search algorithms support exploration in the initial stage of search followed by exploitation in the later stage of search. These techniques are briefly reviewed in the subsequent sub-sections.

2.3.1 *MOO Methods based on Genetic Algorithms*

Genetic algorithm (GA) is inspired by natural evolution phenomenon. Originally, binary strings (or chromosomes) were used to implement GA; later, GA is encoded using real numbers. In either implementation, each individual in the population is randomly initialized. These individuals (or chromosomes) undergo selection, crossover and mutation operations. Selection operation ensures diversity of population with high probability of selecting better individuals for crossover and mutation operations. Crossover operation exchanges information between parent individuals, whereas mutation operation adds new information in the offspring.

In order to solve MOO problems using GA, several researchers have developed different procedures to select individuals (ranking procedure) for the subsequent generation. In vector evaluated GA (Schaffer, 1985), individuals in the population are randomly divided into k sub-populations. Selection of better individuals for the next generation is performed based on one objective function in each sub-population. After selecting the required number of individuals from each sub-population, the combined population is shuffled before applying genetic operations. Multi-objective GA (MOGA), niched Pareto GA (NPGA), strength Pareto evolutionary algorithm (SPEA) and non-dominated sorting GA (NSGA) are other important variants of GA for multiple objectives.

In MOGA (Fonseca and Fleming, 1993), an individual is ranked based on the total number of individuals dominating that individual. This type of ranking puts high selection pressure on the dominated individuals; hence, search may end up with premature convergence. In NPGA (Horn et al., 1994), selection of individuals is performed by a tournament based on niched Pareto dominance; two individuals are

randomly chosen from the entire population and compared against a subset of entire population. Ericson et al. (2002) used Pareto ranking in place of Pareto dominance, and this modified NPGA is called as NPGA 2. In SPEA (Zitzler and Thiele, 1999), an external archive is used to preserve the previously found non-dominated solutions. At each generation, newly found non-dominated individuals compete with the existing individuals in the external archive on the basis of fitness and diversity. A modified SPEA, namely, called SPEA 2 (Zitzler et al., 2001), uses a better fitness assignment, nearest neighbor density estimation and preserves the boundary solutions.

Srinivas and Deb (1994) proposed another variant of GA for multiple objectives with a modification in the ranking procedure, called non-dominated sorting GA (NSGA). In this variant, population is ranked on the basis of non-dominance (Pareto rank), and individuals are selected based on the Pareto rank for the subsequent generation. If two individuals have the same Pareto rank, shared fitness (a measure of solution density) is used for relative ranking of individuals. Deb et al. (2002) modified NSGA, called NSGA-II, for the preservation of elite individuals, faster ranking and use of crowding distance in place of shared fitness. A macro-macro mutation operator (jumping gene) has been used by several researchers to improve the convergence of NSGA-II (Kasat and Gupta, 2003; Agarwal and Gupta, 2008a; Ripon et al., 2007).

2.3.2 MOO Methods based on Differential Evolution

Differential evolution (DE) was proposed by Storn and Price (1997) for solving optimization problems over continuous search space. Chapter 3 provides details on classical DE. Several researchers have improved classical DE in different aspects, such as use of stochastic sampling method to choose individuals, alternative mutation strategies, and binomial and exponential crossover (see Price et al., 2005). DETL uses

taboo check to accept or reject trial individuals (Srinivas and Rangaiah, 2007). It took fewer number of function evaluations and gave high success rate compared to other algorithms, on 16 NLP and 8 MINLP problems (Srinivas and Rangaiah, 2007). Chapter 3 briefly discusses these and other improvements of classical DE.

DE has been successfully adapted by several researchers to solve MOO problems (Abbass et al., 2001; Madavan, 2002; Xue et al., 2003). In generalized DE (GDE; Kukkonen and Lampinen, 2004a) selection rule of basic DE has been modified. Modified GDE (GDE2; Kukkonen and Lampinen, 2004b) makes selection between trial and target individuals based on the crowding distance, if both the individuals are feasible and non-dominated to each other. GDE3 (Kukkonen and Lampinen, 2007 and 2009) incorporates a pruning technique to calculate diversity of non-dominated solutions. Initially, crowding distance is used in GD3 for crowding estimation; later, crowdedness is estimated using nearest neighbors of solutions in Euclidean sense. Chen et al. (2008) introduced niche theory to estimate the diversity, time variant mutation factor, and modified mutation operator in PDE of Abbass et al. (2001). Ji et al. (2008) adapted DE for multiple objectives; contour line and ε-dominance are used along with Pareto ranking and crowding distance calculation to select the individuals for the subsequent generation.

Li et al. (2008) proposed an improved DE, called CDE, for MOO problems. In this, each trial individual is compared with its neighbor to decide whether to preserve it or not for the next generation. This is done on the basis of Pareto ranking followed by crowing distance value. Dong and Wang (2009) proposed DE for multiple objectives with opposite initialization of population and opposite operations on the candidate solutions. Gong and Cai (2009) combined several features of previous evolutionary algorithms like orthogonal initialization of population, ε-dominance

sorting of individuals stored in external archive, storing and inserting extreme points into final archive, and use of random and elitist selection mechanism alternatively. Park and Lee (2009) applied an approach similar to Xue et al. (2003), by dividing the population into several sub-populations and grouping the external archive into clusters. Each cluster is assigned to the nearest sub-population and the best individual from each cluster participates in offspring generation. Qu and Suganthan (2010) proposed MODE with a diversity enhancement mechanism; here, several randomly generated individuals are combined with the current population. Later, Qu and Suganthan (2011) replaced non-dominated sorting based selection for the subsequent generation by normalized objectives and diversified selection.

Stochastic algorithms are sensitive to the values of parameters, and hence several researchers have tried adaptation of DE parameters. Cao et al. (2007) adapted mutation rate (F) and crossover probability (Cr) based on the fitness value of the individuals (non-dominated rank and density). Huang et al. (2007) extended their own work on self-adapted DE for multiple objectives (MOSaDE) where mutation strategy, F and Cr are updated based on results obtained in the previous generations. Huang et al. (2009) modified MOSaDE for objective-wise learning, and called it OW-MOSaDE. Zamuda (2007) adapted MODE parameters similar to the evolutionary strategy, whereas Zielinski and Laur (2007) adapted MODE parameters based on the design of experiments. Qian and Li (2008) proposed self-adaptive MODE where F is modified on the basis of number of current Pareto fronts and diversity of the current population. Qin et al. (2008) used strength Pareto approach to extend DE for multiple objectives. An adaptive Gauss mutation is used to avoid any premature convergence, and Cr value is self-adapted.

Zhang and Sanderson (2008) proposed a self-adaptive multi-objective DE (JADE2), which utilizes information from inferior solutions to modify the values of parameters. Wang et al. (2010) proposed a multi-objective self-adaptive DE (MOSADE) with crowding entropy strategy (distribution of a solution along each objective) to measure crowding degree of the solutions. Li et al. (2011) improved DE for multiple objectives by including tree neighborhood density estimator, strength Pareto dominance to promote convergence, and adaptation of Cr and F values. Zhong and Zhang (2011) proposed a probability based approach to tune values of DE parameters; stochastic coding is applied to improve the solution quality. Qian et al. (2012) encoded algorithm parameters as part of solution, which undergo recombination operations.

It can be seen that strategies used for adapting DE for multiple objectives are similar to those for GA. Performance of different multi-objective DE algorithms are summarized in Table 2.2. In this, performance of GDE3 is comparable to NSGA2_SBX (best algorithm in CEC 2007; Suganthan, 2007) on several MOO test problems. Further, GDE3 performed comparable to several other evolutionary algorithms in CEC 2009 (Zhang et al., 2009). Recently, Adap-MODE (Li et al., 2011) and AS-MODE (Zhong and Zhang, 2011) performed better than GDE3, but they have self-adaptation of algorithm parameters.

Table 2.2: Performance of different multi-objective DE algorithms

Reference	Algorithm name	No. of test functions	Comments on performance
Abbass et al. (2001)	PDE	2	Better than SPEA
Madavan et al. (2002)	-	10	Comparable to NSGA-II
Xue et al. (2003)	MODE	5	Better than SPEA
Cao et al. (2007)	SEA	5	Better than NSGA-II
Huang et al. (2007)	MOSaDE	19	Outperformed by GDE3
Kukkonen and Lampinen (2007 and 2009)	GDE3	19 & 23	Comparable to NSGA-II_SBX (best in CEC 2007) and comparable with other EAs (e.g., DMOEA-DD, MTS) in CEC 2009
Zamuda (2007)	DEMOwSA	19	Outperformed by GDE3
Zielinski and Laur (2007)	MO_DE	19	Outperformed by GDE3
Chen et al. (2008)	MDE	5	Comparable to NSGA-II and inferior to CDE
Ji et al. (2008)	IMODE		Comparable to NSGA-II, SPEA2 and MODE
Li et al. (2008)	CDE	8	Comparable to NSGA-II
Qian and Li (2008)	ADEA	5	Comparable to other EAs
Qin et al. (2008)	ESPDE	5	Better than NSGA-II, SPEA2 and MODE
Zhang and Sanderson (2008)	JADE2	22	Better than NSGA-II
Dong and Wang (2009)	-	5	Better than PDE
Gong and Cai (2009)	paε-ODEMO	10	Better than NSGA-II and SPEA2
Huang et al. (2009)	OW-MOSaDE	13	Outperformed by GDE3
Park et al. (2009)	CMDE	2	Better than PDE
Wang et al. (2010)	MOSADE	18	Better than NSGA-II, SPEA2 and MOPSO
Li et al. (2011)	Adap-MODE	12	Outperformed GDE3 and NSGA-II
Qu and Suganthan (2010)	MODE-DE	19	Comparable to multi-objective DE
Qu and Suganthan (2011)	SNOV_IS	15	Comparable to NSGA2_SBX, GDE3, MOSaDE, DEMOwSA, etc.
Zhong and Zhang (2011)	AS-MODE	10	Outperformed GDE3, OW-MOSaDE and NSGA-II
Qian et al. (2012)	SADE-αCD	11	Superior or comparable to NSGA-II

2.3.3 Other Methods

SA uses concept of annealing process in metallurgy (Kirkpatrick et al., 1983). At each search iteration, a trial point is generated in the neighborhood of the current solution, and the current solution is replaced by the trial point if the latter has better objective value or satisfies Metropolis criterion. Metropolis criterion is used to avoid the SA trapping in local optima. Serafini (1994) modified SA for solving MOO problems. Taboo search (Glover, 1986 & 1989) iteratively search for a better solution in the neighborhood; very importantly, it maintains a short memory which prohibits reverse moves. Gandibleux et al. (1997) adapted taboo search to solve multi-objective combinatorial problems. PSO mimics the social behavior of swarms (Kennedy and Eberhart, 1995). Here, particles (or swarms) iteratively search for better solutions in their neighborhood, and shares their experiences with other particles. Several researchers (Moore and Chapman, 1999; Coello Coello and Salazar Lechuga, 2002) have adapted PSO for solving MOO problems.

Initially, ant colony optimization (ACO) was proposed to solve routing problems; later it was used to solve job-shop scheduling, batch scheduling and combinatorial problems. ACO works on the principle of self organization and transfer of information between individual ants through pheromones. Ants always search shortest path between nest and available food. Mariano and Morales (1999) adopted ACO for multiple objectives; the agents (individuals) are divided into as many families as number of objectives, and each family is independently optimized for a single assigned objective. Information can be shared between different families. As this thesis mainly uses genetic algorithms and differential evolution, recent improvements in the remaining algorithms are not reviewed.

2.4 Recent Applications of MOO in Chemical Engineering[1]

MOO approach has been widely applied in design and operation of chemical and refinery processes. It has also been used in biotechnology, food technology and pharmaceutical industry. Most recently, MOO approach has found applications in new areas such as fuel cells, power plants and bio-fuel production plants. Bhaskar et al. (2000) have reviewed applications of MOO in Chemical Engineering. Later, Masuduzzaman and Rangaiah (2009) have reviewed over hundred reported applications of MOO in Chemical Engineering from the year 2000 until middle of the year 2007. Very recently, Sharma and Rangaiah (2012) have reviewed about 220 MOO articles in Chemical Engineering and related areas, published from the year 2007 until middle of the year 2012. These articles are summarized under six categories (see Table 2.3). Important MOO applications and used objectives in different categories are also given in Table 2.3.

Weighted sum, ε-constraint, NNC, NBI, NSGA-II, NSGA-II-JG, NSGA-II-aJG, NSGA-II-sJG, SPEA, SPEA-2, MOEA (multi-objective evolutionary algorithm), MOGA, MOTS, MOSA, MOPSO methods have been used to optimize several application problems summarized in Table 2.3. Fuzzy approach, NIMBUS, RSM (rough set method), goal attainment, lexicographic goal programming, constraints programming, semi-definite programming, linear physical programming and compromise programming are also applied to optimize one or two applications. In some MOO applications, scalarized SOO problems have been solved using CONOPT,

[1] This section is based on the book chapter: Sharma, S. and Rangaiah, G. P. (2013), Multi-objective optimization applications in chemical engineering, Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.

DICOPT, SQP, BB, CPLEX, LINGO and BARON tools, implemented in GAMS platform.

Tabl3 2.3: Selected MOO applications and used objectives in the period: 2007 to mid-2012

Applications	Performance objectives
1. Process design and operation	
Parameter estimation, heat exchanger networks, crystallization, pervaporation, distillation, reactive distillation, simulated moving bed reactors, batch plants, supply chain, membrane bioreactors and water purification.	Profit, capital/equipment cost, operating cost, cycle time, hot and cold utilities, heat recovery, productivity, conversion, efficiency, product qualities, recycle flow rate, number of equipments, pressure drop, eco indicator 99, potential environmental impact and global warming potential.
2. Petroleum refining, petrochemicals and polymerization	
• Crude distillation units, steam reformer, fuel blending, fluidized bed catalytic cracker, thermal cracker, naphtha pyrolysis, gas separation, hydrogen network and liquefaction of natural gas.	• Profit, investment cost, energy and water consumption, yield, conversion, emissions of greenhouse gases and hydrocarbon inventory.
• Styrene reactor, phthalic anhydride reactor system and butadiene production.	• Cost, productivity and selectivity.
• Low density polyethylene tubular reactor, polymer filtration, nylon-6 and injection molding.	• Monomer conversion, degree of polymerization and batch time.
3. Food industry, biotechnology and pharmaceuticals	
• Lactic acid production, baking of bread, thermal processing and milk concentration.	• Cost, product quality and water content.
• Large scale metabolic networks, protein recovery, flux balance for metabolic networks and bio-synthesis factory.	• Productivity, conversion, yield, metabolic burden and production rate.
• Drug design, bioremediation, antibiotic and penicillin V production, scheduling and product development.	• Productivity, conversion, make-span, treatment time, cost of production media and product concentration.

4. Power generation and carbon dioxide emission	
Pulverized coal power plants and their retrofitting, natural gas power plant, integrated gasification and combined cycle power plant and cogeneration plant.	Capital cost, fuel cost, emissions of CO, CO_2 and NO_x, exergetic efficiency and net power.
5. Renewable energy	
Bio-diesel, bio-ethanol, biomass gasification plant, combined SNG (synthetic natural gas) and electricity production, solar Rankine cycle and reverse osmosis.	Cost, profit, NPV (net present value), operating cost, energy efficiency, water consumption, global warming potential, eco indicator 99, greenhouse gas emissions, productivity and conversion.
6. Hydrogen production and fuel cells	
• Methane steam reforming, photovoltaic-battery-hydrogen storage system and hydrogen plant with CO_2 absorber.	• Hydrogen production rate, energy cost and CO_2 emissions.
• Polymer electrolyte membrane fuel cell, solid oxide fuel cell, tubular solid oxide fuel cell, alkaline fuel cell, fuel cell electrode assembly, phosphoric acid fuel cell system, molten carbonate fuel cell and its system.	• Cost of fuel cell system, efficiency, current density and size of stack.

2.5 Conclusions

The stochastic search methods can locate the global optimum with high reliability although they may require considerable computational time. NSGA-II has been commonly used for solving MOO application problems. Two strategies have been mainly used for adapting GAs for MOO; non-dominated sorting followed by crowding distance calculation and maintaining an external archive to store non-dominated solutions. Further, MODE is a reliable and efficient algorithm, based on its performance in CEC 2007 and CEC 2009 competitions. Selection strategies used for adapting DE for multiple objectives are similar to those for adapting GAs.

DETL (Srinivas and Rangaiah, 2007) has proven to be highly reliable and requires fewer number of function evaluations. DE has inherent characteristics to exploit the search space at the end of search, and so use of taboo check improves its exploration capability. Most of the reported adaptations of DE for multiple objective uses classical DE; hence, DETL is chosen to develop I-MODE algorithm in the next chapter. Similarly, deterministic search methods are likely to be better for obtaining the Pareto-optimal front precisely and efficiently. Additionally, termination of a stochastic search is very often using maximum number of generations/iterations, which is simple but it is unlikely to take a timely decision about the stochastic search termination. These provide motivation and scope for developing I-MODE and hybrid algorithms in the subsequent chapters.

Chapter 3

An Improved Multi-objective Differential Evolution with a Termination Criterion[2]

3.1 Introduction

Multi-objective optimization (MOO) is often required due to conflicting objectives in engineering applications, and there have been many studies on evolutionary algorithms (EAs) for MOO and their applications in the past decade. Most of these studies in the literature have used maximum number of generations (MNGs) as the search termination criterion. The reliability and efficiency of any stochastic search for practical applications depend on the termination criterion used in the iterative method. If the optimization problem is easy to solve, the algorithm may obtain the global solution quite early in comparison to the given MNGs. But, for difficult problems, the specified MNGs can be insufficient for converging to the global solution. Further, solving a problem using the same algorithm may require different MNGs in different runs due to stochastic nature of evolutionary algorithms. So, monitoring the search with generations is important to make the right decision on the search termination.

Although many EAs have been developed and applied, there have been only a few studies on the search termination. Furthermore, performance metrics such as generational distance, GD^t (Van Veldhuizen and Lamont, 1998), spread, SP^t (Deb et al., 2000), hyper volume, HV (Zitzler and Thiele, 1998) and epsilon indicators

[2] This chapter is based on the manuscript: Sharma, S. and Rangaiah, G. P. (2013), An improved multi-objective differential evolution with a termination criterion for constrained optimization problems, Computers and Chemical Engineering, under review.

(Zitzler et al., 2003) have been proposed in the literature. These performance metrics have been used to assess the quality of the final non-dominated solutions obtained, for comparing the performance of multi-objective EAs (Zhang and Sanderson, 2008; Wang et al., 2010). However, to the best of our knowledge, only a few studies (reviewed below) have used performance metrics to monitor the search progress and to terminate the search.

Rudenko and Schoenauer (2004) introduced a termination criterion based on the density (i.e., crowding distance) of non-dominated solutions. They noticed that maximal crowding distance stabilizes last after the minimal and average crowding distances; and so the search is terminated if the maximal crowding distance is not changing more than a given limit over a fixed number of generations. Trautmann et al. (2008) and Wagner et al. (2009) proposed termination criterion using different performance metrics to monitor the search progress; if variation in the performance metric is small over a fixed number of generations or variance of the performance metric values decreases below some specified limit, then search is terminated. Recently, a variant of online convergence detection (OCD), based on HV, has been studied by Wagner and Trautmann (2010).

Sindhya et al. (2008) developed local search based evolutionary MOO for fast and accurate convergence where some of the newly generated individuals are improved using local search. The hybrid search terminates after a fixed number of generations or when the local search does not improve the newly generated individuals much. Marti et al. (2009) introduced another termination criterion for MOO algorithms, which combines the mutual domination rate as an improvement indicator and a simplified Kalman filter for evidence gathering. The mutual domination rate is the number of non-dominated individuals obtained in the current generation dominating

the individuals obtained in the last generation. The search is terminated when mutual domination rate is below a threshold value.

In the studies reviewed above, termination criteria based on performance metrics have been tested on three to six test functions, many of which do not have constraints commonly present in engineering application problems. Further, variations in the selected performance metrics have not been analyzed, which is required to check their suitability for reliable termination of the search. This chapter addresses these issues. For this, five performance metrics, namely, GD, SP, HV ratio (HVR), R2 and additive epsilon (ε_+) are studied for assessing the search progress. Some of these performance metrics are modified to avoid the use of the true Pareto-optimal front, yet to be found in applications. Variations in the modified performance metrics are observed on five test functions, and GD and SP are selected for the development of the termination criterion. Based on this analysis, it is proposed to terminate the search if the improvement in variance of GD and SP in recent generations is statistically insignificant. Then, the proposed termination criterion is tested on nine constrained problems from Zhang et al. (2009), which make the findings useful for applications. This study is carried out in conjunction with a multi-objective differential evolution (MODE) algorithm.

Differential evolution (DE) was proposed by Storn and Price (1995, 1997) for solving optimization problems over continuous search space. Later, several researchers have improved classical DE in different aspects such as population initialization, mutation, crossover and selection operations (Price et al., 2005; Brest et al., 2006; Rahnamayan et al., 2008). A recent review of these developments and applications of DE is available in Chen et al. (2010). Srinivas and Rangaiah (2007) used DE with taboo list (DETL) to accept or reject trial individuals, which is useful in

reducing unnecessary function evaluations. Omran et al. (2009) proposed bare-bones DE (BBDE) which performed better than DE with little or no parameter tuning. Qin et al. (2009) proposed self-adaptive DE (SaDE) for adapting learning strategy, mutation rate (F) and crossover probability (Cr) values. Li et al. (2010) have used simplified quadratic approximation for the enhancing the performance of classical DE; performance of the proposed approach is comparable with the state of the art stochastic global optimization methods. Recently, Zhang and Rangaiah (2011) have presented integrated DE (IDE), which uses taboo list to avoid the revisit of search space, parameter adaptation, a new termination criterion and local optimization after the global search. IDE performed better than the recent global optimization algorithms on 26 test functions.

DE has been successfully adapted to solve MOO problems. Generalized DE (GDE) was proposed by Kukkonen and Lampinen (2004a) with a modified selection rule of basic DE. The latest version is GDE3 (Kukkonen and Lampinen, 2007 and 2009), which incorporates non-dominated sorting of combined population and a pruning technique to calculate solution diversity. Performance of GDE3 is comparable to many other MOEAs on several MOO test problems (see Table 2.2). Similarly, other works (Chen et al., 2008; Ji et al., 2008; Li et al., 2008; Qin et al., 2008; Dong and Wang, 2009; Gong and Cai, 2009; Park and Lee, 2009) have adapted DE for multiple objectives. These works have been reviewed in Chapter 2.

In many applications, evaluation of objective functions and constraints is computationally expensive. Hence, taboo list is used with MODE to avoid revisits and for better exploration of the search space. In other words, DETL is adapted and improved to solve MOO problems, and the resulting algorithm is referred to as integrated multi-objective differential evolution (I-MODE). In brief, features of I-

MODE include the use of taboo list/check for efficient exploration, self-adaptation of parameters based on the strategy of Zhang and Sanderson (2008), and the use of the proposed termination criterion. I-MODE is tested on many test functions having constraints from Zhang et al. (2009). Additionally, effect of taboo radius on the performance of I-MODE is studied. In short, contributions of this chapter are: development and assessment of an effective termination criterion for evolutionary algorithms, and development and evaluation of I-MODE for constrained MOO problems.

Next section of this chapter discusses the adaptation of DETL algorithm for multiple objectives. Section 3.3 describes the self adaptation of algorithm parameters, followed by dominance based constraints handling approach. Section 3.4 presents different performance metrics and some modifications in them, to avoid the use of true Pareto-optimal front. Suitable performance metrics are also selected in Section 3.4. Section 3.5 presents development of search termination criterion using selected performance metrics. Section 3.6 briefly describes I-MODE algorithm. Effect of termination parameters on the performance of I-MODE algorithm is explored in Section 3.7. Section 3.8 presents the effect of taboo radius on I-MODE performance. Conclusions from this work are drawn in the last section of this chapter.

3.2 Adaptation of DETL for Multiple Objectives

In this study, DETL of Srinivas and Rangaiah (2007), which was proven to be highly reliable and requires fewer NFEs for global optimization, is adapted for multiple objectives. DE has inherent characteristics to exploit the solution space towards the end of the search, whereas use of taboo list/ check will improve its exploration capabilities. DE/rand/1 mutation strategy and binomial crossover are used

in the I-MODE algorithm. Classical DE for SOO has four main steps: (i) initialization, (ii) mutation, (iii) crossover, and (iv) selection (Price et al., 2005). These steps are discussed in the following sub-sections.

i) Initialization

In this, a population of N individuals is randomly initialized inside the bounds on decision variables.

$$x_j = x_j^L + \text{rand}(0,1)\,(x_j^U - x_j^L) \qquad j = 1, 2, \ldots, \text{no of DVs} \qquad (3.1)$$

Here, rand (0, 1) is a random number from uniform distribution between 0 and 1, and x_j^L and x_j^U are the lower and upper bounds on j^{th} decision variable. For producing a single individual, equation 3.1 has to be applied on each decision variable.

ii) Mutation

Mutation is performed using different strategies; selection of a strategy depends on the type of problem. Price et al. (2005) proposed several DE strategies.

a) DE/rand/1 $\quad \mathbf{v_i} = \mathbf{x_{r0}} + F(\mathbf{x_{r1}} - \mathbf{x_{r2}})$ (3.2a)

b) DE/best/1 $\quad \mathbf{v_i} = \mathbf{x_{best}} + F(\mathbf{x_{r1}} - \mathbf{x_{r2}})$ (3.2b)

c) DE/rand/2 $\quad \mathbf{v_i} = \mathbf{x_{r0}} + F(\mathbf{x_{r1}} - \mathbf{x_{r2}}) + F(\mathbf{x_{r3}} - \mathbf{x_{r4}})$ (3.2c)

d) DE/best/2 $\quad \mathbf{v_i} = \mathbf{x_{best}} + F(\mathbf{x_{r1}} - \mathbf{x_{r2}}) + F(\mathbf{x_{r3}} - \mathbf{x_{r4}})$ (3.2d)

e) DE/rand-to-best/1 $\quad \mathbf{v_i} = \mathbf{x_i} + F(\mathbf{x_{best}} - \mathbf{x_i}) + F(\mathbf{x_{r3}} - \mathbf{x_{r4}})$ (3.2e)

Here, $\mathbf{x_{best}}$ and $\mathbf{x_i}$ are respectively the best and target individuals from the parent population. $\mathbf{x_{r0}}$, $\mathbf{x_{r1}}$, $\mathbf{x_{r2}}$ and $\mathbf{x_{r3}}$ are randomly selected individuals from the parent population. The general convention used in the above strategies is DE/p/q. Here, p stands for perturbation vector (first term on the right hand side) and q is the number of difference vectors (remaining terms on the right hand side). Figure 3.1 shows

generation of mutant vector using strategy DE/rand/1 in two-dimensional search space.

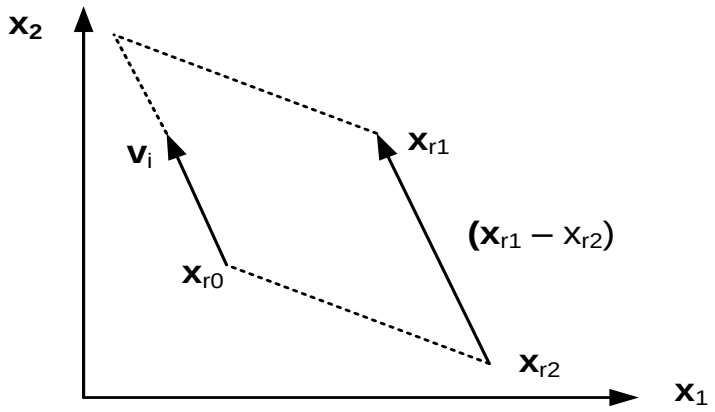


Figure 3.1: Generation of a mutant vector based on strategy in equation 3.2a for two decision variables

iii) Crossover

In the binomial crossover, elements of mutant (**v**) and target (**x)** vectors compete with each other, with a probability Cr to generate trial vector (**u)**.

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } \text{rand}(0,1) \leq \text{Cr}, \text{or } j = j_{\text{rand}} \\ x_{i,j} & \text{otherwise} \end{cases} \quad (3.3)$$

In the exponential crossover, one randomly selected element of mutant vector is copied into the trial vector, so that trial vector will be different from the target vector. After that, a random number is generated between 0 and 1 using uniform distribution; if this random number is lower than Cr, then one randomly selected element of mutant vector is copied into trial vector. This process is repeated until the generated random number is greater than Cr; then, current and remaining elements of the trial vector are copied from the target vector.

iv) Selection

After crossover, the trial vector goes through a check on decision variable violation. If the trial vector is above the upper or below the lower bound for any decision variable, it is randomly re-initialized within the bounds on that decision variable. DE performs selection between the trial and target vectors based on the objective function value (in case of minimization of objective).

$$\mathbf{x}^{\mathbf{G+1}} = \begin{cases} \mathbf{u}^{\mathbf{G}} & \text{if } f(\mathbf{u}^{\mathbf{G}}) \leq f(\mathbf{x}^{\mathbf{G}}) \\ \mathbf{x}^{\mathbf{G}} & \text{otherwise} \end{cases} \tag{3.4}$$

Here, $\mathbf{x}^{\mathbf{G+1}}$ is the selected individual for the next generation.

Mutation, crossover and selection operations are performed for each individual in the population. This completes one generation (or iteration) of the algorithm. Many generations are performed until the specified termination criterion is met. The common termination criterion is the maximum number of generations.

DETL is adapted for a MOO problem as follows. In the initialization step, population is initialized randomly inside the bounds on decision variables. Values of objectives and constraints are calculated for each individual in the initial population, and the taboo list (TL) is randomly filled using the initial population. In each generation, a trial vector for each target vector in the initial/current population is generated by mutation of three randomly selected individuals from the population followed by binomial crossover. Taboo check is implemented in the evaluation step of trial vector; if the trial individual is near to any individual in the TL by a specified distance, then it is rejected without calculating values of objectives and constraints (Figure 3.2). Accepted trial individual is stored in the child population, and also added

to the TL. In each generation, child population is mixed with the current/parent population. Pareto dominance ranking with crowding distance calculations of combined population is used to select the individuals for the subsequent generation. See Deb (2001) for details on Pareto dominance ranking and crowding distance calculations.

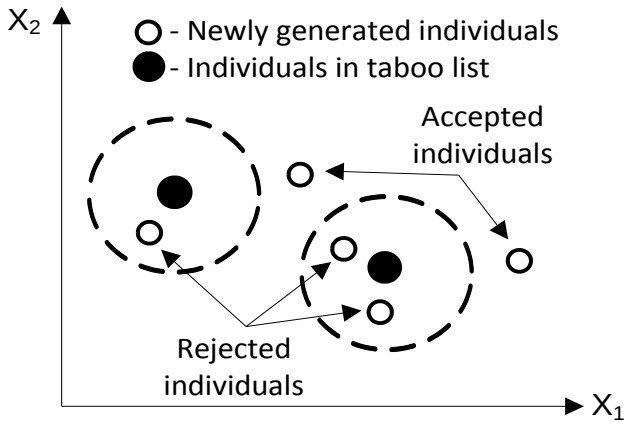


Figure 3.2: Illustration of use of TL: trial individual, near to any individual in the TL by a specified distance, is not evaluated for objectives and constraints.

3.3 Self Adaptation of Algorithm Parameters and Constraints Handling

Stochastic search algorithms are sensitive to values of their parameters, and different problems require different values of parameters for good performance. Hence, tuning of parameters requires prior experience on the particular problem and/or extensive effort. In the recent past, self-adaption of algorithm parameters in MODE has been studied by several researchers. Zamuda (2007) used self-adaptation mechanism from evolution strategy to adapt F and Cr values. Zielinski and Laur (2007) adapted F and Cr values based on the design of experiments. Cao et al. (2007) proposed a performance-based self-adaptation; if fitness values of all individuals in the population have become similar, then F and Cr are increased for greater exploration of search space; otherwise, they are decreased for greater exploitation. Zhang and Sanderson (2008) chose Cr and F values randomly using Normal and

Cauchy distribution respectively, for generating a trial individual; then, mean value of Cr and location parameter of F are updated after each generation. Their algorithm, JADE2 outperformed non-dominated sorting genetic algorithm - II (NSGA-II) for 19 out of 22 test functions, and performed better than GDE3 in terms of HV and epsilon performance indicators. Huang et al. (2009) proposed a self-adaptive MODE algorithm with objective-wise learning strategies, where mutation strategy and Cr value are adapted for each objective separately. Wang et al. (2010) proposed a multi-objective self adaptive DE, where Cr and F values are randomly reinitialized for an individual if it does not produce a better trial vector in certain number of consecutive generations.

The self-adaption in JADE2 is easy to use and has shown good performance. Hence, self-adaptation strategy proposed by Zhang and Sanderson (2008) is employed in I-MODE. In this strategy, CR and F values are obtained, for generating each trial individual, as follows.

$$CR_i = randn_i(\mu_{CR}, 0.1) \tag{3.5a}$$

$$F_i = randc_i(\mu_F, 0.1) \tag{3.5b}$$

Here, randn is a random value from the Normal distribution with mean μ_{CR} and standard deviation of 0.1, whereas randc is a random value from Cauchy distribution with location parameter μ_F and scale parameter of 0.1.

Initially, $\mu_{CR} = \mu_F = 0.5$. Values of Cr and F thus generated for each target individual are considered successful if the trial vector is selected for the subsequent generation. After each generation, μ_{CR} and μ_F are updated, based on the performance of generated Cr and F, as follows.

$$\mu_{CR} = \mu_{CR}(1\text{-}c) + \text{mean}_A(S_{CR})c \quad (3.6a)$$

$$\mu_F = \mu_F(1\text{-}c) + \text{mean}_L(S_F)c \quad (3.6b)$$

Here, parameter c is set to be equal to 0.1, and mean_A and mean_L denote arithmetic and Lehmer mean respectively. S_{CR} and S_F are sets of successful crossover probabilities and mutation rates in the generation completed.

Evolutionary algorithms were developed for solving optimization problems with bounds on variables but without constraints; later, several approaches have been proposed to handle the constraints. Coello Coello (2002) summarized constraints handling methods under five categories: penalty function approach; separation of constraints and objectives; special representation; repair algorithms; hybrid methods. Both penalty function and feasibility approaches are commonly used in the literature for constraints handling. Of these two, feasibility approach is an attractive choice as it does not have any parameter and its good performance has been shown by Deb et al. (2002) for MOO. So, this approach is chosen for handling inequality constraints in I-MODE.

In the feasibility approach, number of constraint violations and total constraints violation for each individual are calculated and used to select individuals for the subsequent generations. The following criteria are employed to handle inequality constraints in I-MODE algorithm.

i) If two individuals A and B have no constraints violation (i.e., feasible solutions), then the selection of individual is based on the usual dominance criteria.

ii) If one individual is feasible and another is infeasible, then the former is selected over the infeasible individual.

iii) If both the individuals are infeasible, then the individual with less number of constraint violations followed by less total constraints violation is selected.

I-MODE algorithm has been implemented in the commonly available MS-Excel and VBA (Visual Basic for Applications) platform. VBA is used to implement the algorithm steps, calculation of performance metrics and termination criterion, while Excel worksheets are used to calculate objectives, constraints, linking between cells, input parameters, and display the results. This organization allows any engineer familiar with Excel to use I-MODE for his/her applications.

3.4 Selection of Performance Metrics for Termination Criteria

3.4.1 Existing Performance Metrics

Some performance metrics such as GD^t and set convergence ratio are used to check the convergence of the non-dominated solutions to the true Pareto-optimal front while some others such as SP^t and maximum spread are used to check the spread of the non-dominated solutions obtained along the Pareto-optimal front. Further, some performance metrics use only one Pareto-optimal set, called unary quality indicators. Other performance metrics, called binary quality indicators, quantitatively compare two approximation sets. Important performance metrics are discussed below; all these are defined in the objective function space. For clarity, the phrase "non-dominated solutions" is used for the optimal solutions obtained by the numerical algorithm, and the phrase "true Pareto-optimal front" refers to the analytical/known Pareto-optimal front.

(i) Generational distance is used to evaluate the closeness of non-dominated solutions to the true Pareto-optimal front (Van Veldhuizen and Lamont, 1998).

$$GD^t = \frac{1}{N}\sqrt{\sum_{i=1}^{N} d_i^2} \qquad (3.7)$$

Here, N is the number of non-dominated solutions obtained, and d_i is the Euclidean distance of each of these solutions to its nearest point in the true Pareto-optimal front.

(ii) Spread, introduced by Deb et al. (2000) for bi-objective problems, measures the distribution of non-dominated solutions obtained. Its calculation requires the Euclidean distance between neighboring solutions in the non-dominated solutions obtained, and also between the boundary solutions in the non-dominated solutions and those in the true Pareto-optimal front. Zhou et al. (2006) extended spread metric for more than two objectives, by considering the distance of each point to its nearest point, as follows.

$$SP^t = \frac{\sum_{m=1}^{M} d(e_m,S) + \sum_{i=1}^{N} |d_i - \bar{d}|}{\sum_{m=1}^{M} d(e_m,S) + N\,\bar{d}} \qquad (3.8a)$$

$$d_i = \min \| S_i - S_j \|^2 \text{ with respect to } j = 1, 2, \ldots, N \text{ (except } j = i) \qquad (3.8b)$$

Here, M is the number of objective functions, $\{e_1, e_2, \ldots, e_M\}$ are M extreme/boundary solutions from the true Pareto-optimal front, and S is the set of non-dominated solutions obtained. Further, $d(e_m, S)$ is the Euclidean distance between the extreme solution of m^{th} objective in the true Pareto-optimal front to its nearest non-dominated solution obtained (set S). d_i is the Euclidean distance of solution, S_i in the set S to its nearest solution in the same set, and $\bar{d}$ is the average of d_i for all non-dominated solutions in set S.

(iii) Hyper volume (Zitzler and Thiele, 1998) provides a quantitative measure of both the convergence to the true Pareto-optimal front and diversity of non-dominated solutions obtained. A hypercube is constructed for each point in the Pareto-optimal

front (which can be true or obtained as desired), and hyper volume is the amount of space occupied by the union of hyper-cubes constructed at all solutions. In the case of bi-objective optimization problems, hyper-cubes will be rectangular in shape and combined area of these rectangles at different solution points will be the hyper volume. For bi-objective optimization problem, width of each rectangle is equal to the difference between two consecutive non-dominated solutions obtained, while length/height is obtained using nadir point and solution from the obtained Pareto-optimal front.

$$HV = U_{i=1}^{N-1} vh_i \tag{3.9}$$

Here, vh_i is the volume of i^{th} hypercube. When ranges of different objectives are different from one another, hyper volume ratio (HVR) is used instead of hyper volume (see equation 3.10).

$$HVR = HV(\text{obtained Pareto})/HV(\text{reference Pareto}) \tag{3.10}$$

(iv) R indicators (Hasan and Jaskiewicz, 1998) compare the true Pareto-optimal front with the non-dominated solutions obtained on the basis of a set of utility functions (e.g., weighted sum). In this quantification, the mapping domain is the utility function and not the set of non-dominated solutions obtained. R indicators require the ideal point, nadir point and weight vector (Hasan and Jaskiewicz, 1998). Generally, normalized/equidistant weight vectors, which divide the objective space equally, are employed. Utility functions are defined as:

$$U(w_k, S_i) = \max_{1 \le m \le M} \left\{ w_{k,m} \left[1 - \frac{abs(Ideal_m - S_{i,m})}{abs(Ideal_m - Nadir_m)} \right] \right\} \tag{3.11a}$$

$$U(w_k, S) = \min_{S_i \in S} \{U(w_k, S_i)\} \tag{3.11b}$$

$$U(w) = \frac{1}{nw}\sum_{k=1}^{nw} U(w_k, S) \quad (3.11c)$$

Here, w_k is the set of equidistant weights, and nw is the number of weights. $U(w_k, S_i)$ is the weighted Techebycheff utility function for a particular individual S_i in the set S. R indicators are defined based on the utility functions in equations 3.11a-c. R1 indicator is defined as the fraction of total weights where one Pareto-optimal front gives better utility function value compared to the other (equations 3.11a-b). It does not consider the extent by which a Pareto front is better/worse than other. On the other hand, R2 indicator gives the difference between average values of utility function obtained for two different Pareto-optimal fronts using complete set of weights (see equations 3.11a-c).

(v) Epsilon indicators (Zitzler et al., 2003) comprise additive and multiplicative versions. Both these indicators exist in unary and binary form. The binary multiplicative epsilon indicator gives the minimum factor by which each solution point in a Pareto-optimal front can be multiplied such that resulting transformed set is still weakly dominated by another Pareto-optimal front. Similarly, binary additive epsilon gives the minimum factor by which each solution point in a Pareto-optimal front can be added such that the resulting transformed set is still weakly dominated by another Pareto-optimal front. Equations 3.12a-c below define additive epsilon (binary) indicator.

$$\varepsilon_{s1,s2} = \max_{1\le m\le M}\{ s1^m - s2^m\} \quad (3.12a)$$

$$\varepsilon_{s2} = \min_{s1 \in S1}\{\varepsilon_{s1,\ s2}\} \quad (3.12b)$$

$$I_{\varepsilon+}(S1, S2) = \max_{s2\in S2}\{\varepsilon_{s2}\} \quad (3.12c)$$

In these, S1 and S2 are the two different Pareto-optimal fronts, and s1 and s2 are individuals from the sets S1 and S2, respectively. For example, S1 can be the true Pareto-optimal front and S2 can be the non-dominated solutions obtained.

3.4.2 Modified Performance Metrics and Their Evaluation

Pareto-optimal solutions for real world applications are not known in advance, and so calculation of performance metrics for search termination should not use true Pareto-optimal front. In this work, only the non-dominated solutions obtained as search progresses are used to calculate the values of different performance metrics for monitoring the search progress. Objectives are normalized using the extreme values of each objective in the current (and previous) generation, and then all performance metrics are calculated for normalized values of objective functions.

GD is calculated between the non-dominated solutions obtained in the current generation and those in the previous generation, using equation 3.7 where non-dominated solutions obtained in the current generation are considered as the true Pareto-optimal front. In the calculation of SP, extreme solutions of the true Pareto-optimal front cannot be used. So, SP is calculated for the non-dominated solutions in the current generation using the following equations.

$$SP = \frac{\sum_{i=1}^{N} |d_i - \bar{d}|}{N\bar{d}} \qquad (3.13a)$$

$$d_i = \min \| S_i - S_j \|^2 \text{ with respect to } j = 1, 2, \ldots, N \text{ (except } j = i) \qquad (3.13b)$$

Here, N and S are the number and set of non-dominated solutions in the current generation respectively. The remaining symbols are same as those in equations 3.8a-b.

HVR, R2 and additive-ε indicators are calculated using the non-dominated solutions obtained in the current and previous generations. Set of non-dominated solutions obtained in the current generation is considered as the true Pareto-optimal front. Nadir point vector of {1, 1} is used in HVR calculation for two objectives (Deb, 2001). In R2 indicator calculation, values of ideal and nadir points are {0, 0, …, 0} and {1.1, 1.1, ..., 1.1} respectively; these values are taken from source code of R indicators available at http://dbk.ch.umist.ac.uk/knowles/ under the GNU.

Table 3.1 summarizes the number of arithmetic operations (square, multiplication, division, square root and subtraction) involved in different modified performance metrics for monitoring search progress. M and N are the number of objective functions and non-dominated solutions obtained, respectively. nw is the number of equidistant weights, which is required in R2 calculation.

Table 3.1: Arithmetic operations involved in the calculation of performance metrics

Performance metric	Arithmetic operations involved			
	("square" and "×")	("/")	("√")	("-")
GD	$MN^2 + N$	-	N^2	MN^2
SP	MN(N-1)	-	N^2	MN(N-1) + N
HVR (2-D)	2(N-1)	-	-	4(N-1)
R2	2MN(nw)	2MN(nw)	-	2MN(nw)
ε_+	-	-	-	MN^2

Smallest Euclidean distance calculation between one solution from the non-dominated set obtained in the previous/last generation and another solution in the current generation, requires MN subtraction, MN square and N square root operations. In GD calculation, N Euclidean distances are calculated for all non-dominated solutions obtained in the last generation. Additional N squares are required in GD calculation (see equation 3.7) using N minimum Euclidean distances obtained for all

non-dominated solutions in the last generation. In SP calculation, Euclidean distance between each individual and its nearest individual is calculated in the latest generation; it requires M(N-1), M(N-1) and N subtraction, square and square root operations, respectively. The task of finding of minimum Euclidean distance for all non-dominated solutions in the current generation needs N times the number of mathematical operations required by one individual (i.e., M(N-1) subtractions, M(N-1) squares and N square roots). Additional N subtractions are used to obtain the difference between the obtained minimum Euclidean distance for each individual and average Euclidean distance. Table 3.1 does not include one or two mathematical operations used in calculation of a metric; for example, GD calculation requires one final division by N, which is negligible.

In HV calculation for two objectives, (N-1) rectangles are formed by one nadir point and N non-dominated solutions. In total, 2(N-1) subtraction and (N-1) multiplication operations are required to obtain the HV value. So, HVR between non-dominated solutions in the current and previous generations uses twice the number of subtractions and multiplications used in HV calculation. In R2 indicator, utility function calculation for N non-dominated solutions on a particular weight requires MN multiplications, MN subtractions and MN division. Here, subtractions used in ($Ideal_m$ - $S_{i,m}$) and ($Ideal_m$ - $Nadir_m$) are not considered as ideal vector is {0, 0,…,0}. Final R2 calculation for two objectives uses 2×nw times the above number of mathematical operations. In ε_+ indicator, each non-dominated solution in the last generation is compared with all the non-dominated solutions obtained in the current generation (S1). This comparison is done for all objectives, which takes MN subtractions. There are maximum N non-dominated solutions in set S2; hence, total subtractions to obtain ε_+ indicator value are MN^2. In summary, HVR is the least

computationally expensive performance metrics, whereas GD and SP require nearly same computational effort. R2 indicator is the most computationally expensive indicator, which requires more than twice compared to GD and SP. Computations required by ε_+ indicator are in between GD/SP and R2 indicator.

3.4.3 Selection of Modified Performance Metrics for Search Termination

Performance of modified performance metrics is assessed on five test functions (ZDT3, ZDT4, CF1, CF4, and CF6). These test functions have different characteristics, e.g., continuity and discontinuity of objectives, modality and convexity of search space, and constraints. These and other test functions used in this study are summarized in Table 3.2. ZDT3 and ZDT4 are unconstrained test functions proposed by (Zitzler et al., 2000), while CF1-9 are constrained test functions used in CEC-2009 (Zhang et al., 2009). Variations in the modified performance metrics with generations in solving the selected test functions by I-MODE are shown in Figure 3.3. Non-dominated solutions obtained using I-MODE at numerous generations are reviewed manually to identify the number of generations required for converging to the true Pareto-optimal front, and these are found to be nearly 90, 300, 130, 250 and 160 generations for ZDT3, ZDT4, CF1, CF4 and CF6 functions, respectively; these generations are marked in Figure 3.3 with dotted vertical lines.

In the selection of suitable performance metrics for termination criterion, computational complexity, physical significance, and variation with generations have to be considered. Based on the computational effort required, HVR and ε_+ indicator are most suitable metrics, followed by GD and SP (Table 3.1). Generally, value of nw is larger than N, and so use of R2 indicator for termination makes the algorithm slow. In HV calculation for two objectives, different areas are approximated as rectangles.

HV calculation for more number of objectives requires more approximations, and it is also difficult to obtain nadir point for a problem with more than two objectives. Value of ε_+ indicator will be zero, if some of the solutions are common in both the Pareto-optimal fronts obtained in the current and previous generations. For termination, a good indicator should vary initially before convergence, and then should stabilize at a constant value after convergence so that the global search can be terminated at the right time (i.e., not too early to avoid approximate solutions or too late for computational efficiency).

GD considers average improvement in the non-dominated solutions obtained in the current generation compared to the non-dominated solutions obtained in the previous generation. Its values after the generations required for convergence are relatively small for the test functions (see Figure 3.3a). Initially, SP varies much for all the test functions (Figure 3.3b), later it fluctuates in a narrow range due to change in the number of solutions in the best Pareto-optimal front in different generations. In particular, SP seems to be a suitable performance metric for monitoring the progress of multi-modal test function (i.e., ZDT4). Rudenko and Schoenauer (2004) mentioned that distribution of solutions on the Pareto-optimal front improves after the search reaches the true Pareto-optimal front. For all the 5 test functions, HVR stabilizes well before the generations required for convergence to the global solutions. This indicates that use of HVR for termination may lead to poor results. Similar to GD, some fluctuation can be seen in HVR for CF4 test function even after 250 generations.

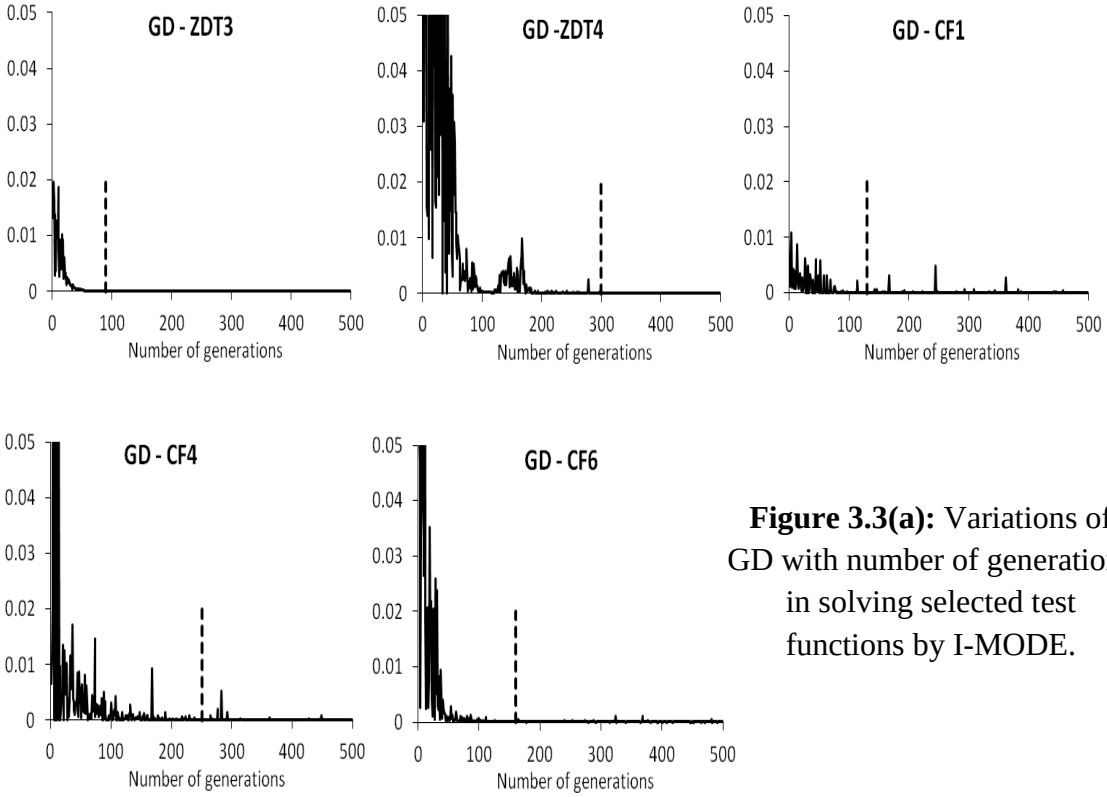


Figure 3.3(a): Variations of GD with number of generations in solving selected test functions by I-MODE.

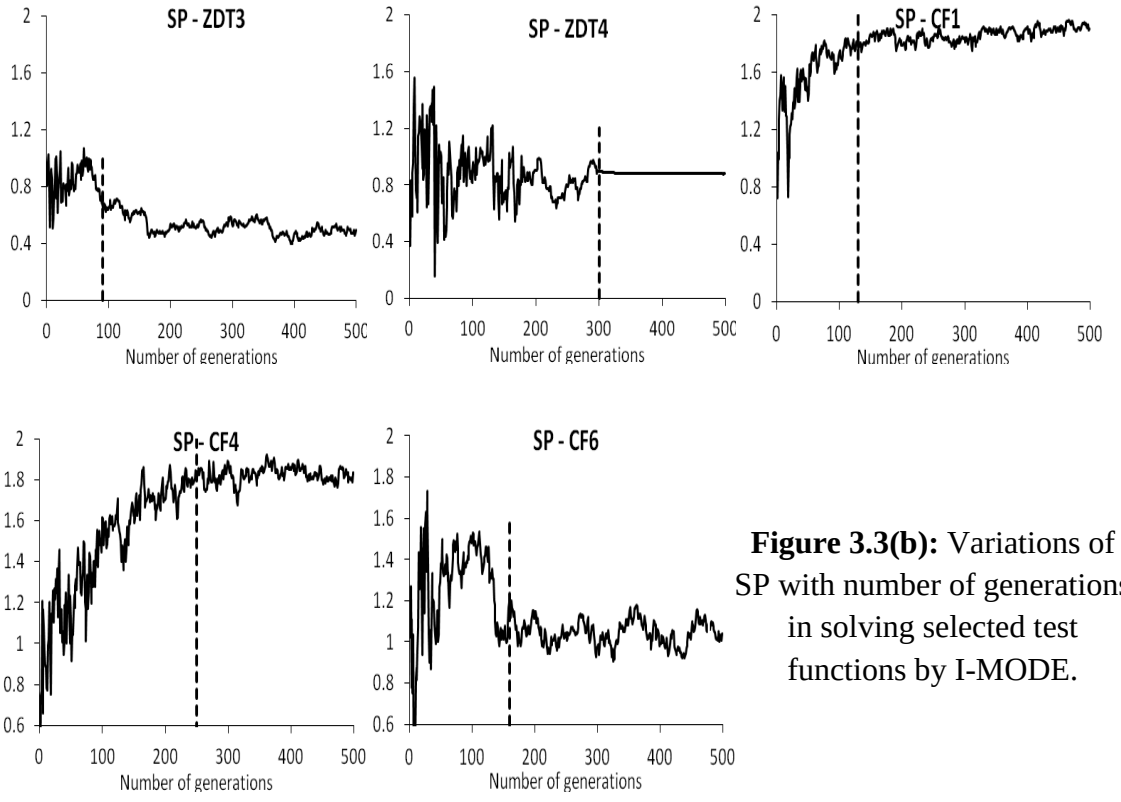


Figure 3.3(b): Variations of SP with number of generations in solving selected test functions by I-MODE.

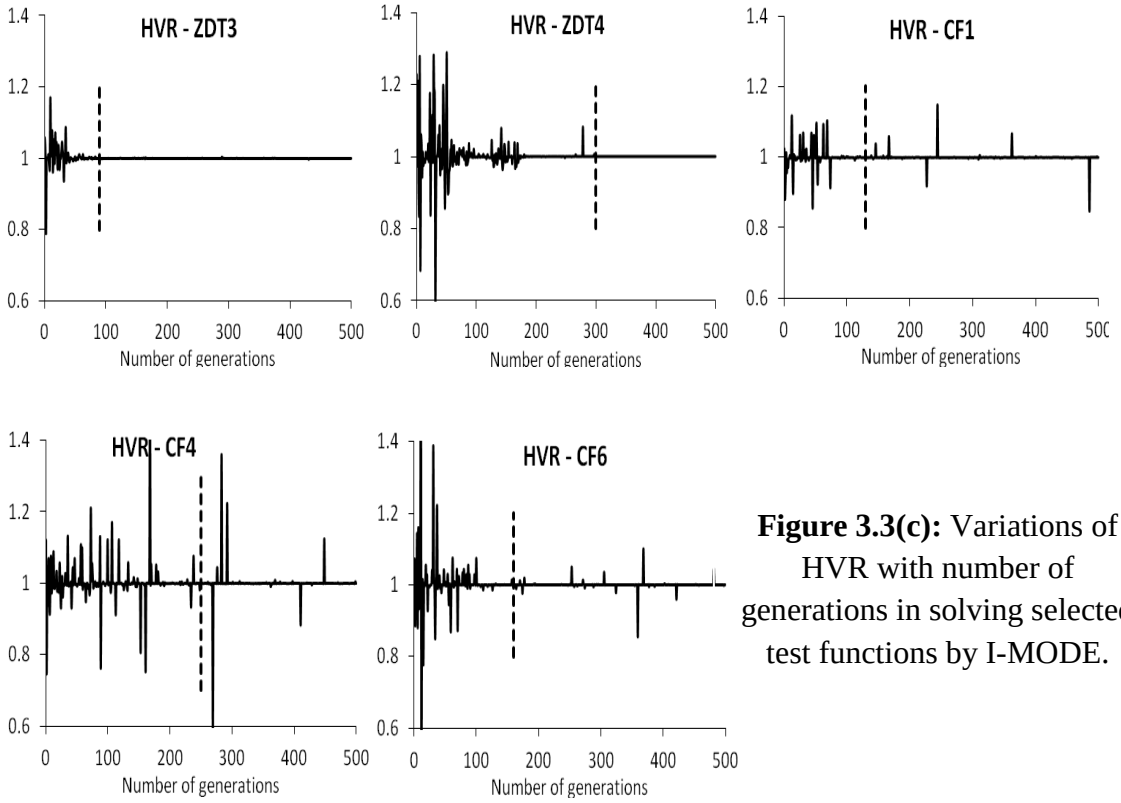


Figure 3.3(c): Variations of HVR with number of generations in solving selected test functions by I-MODE.

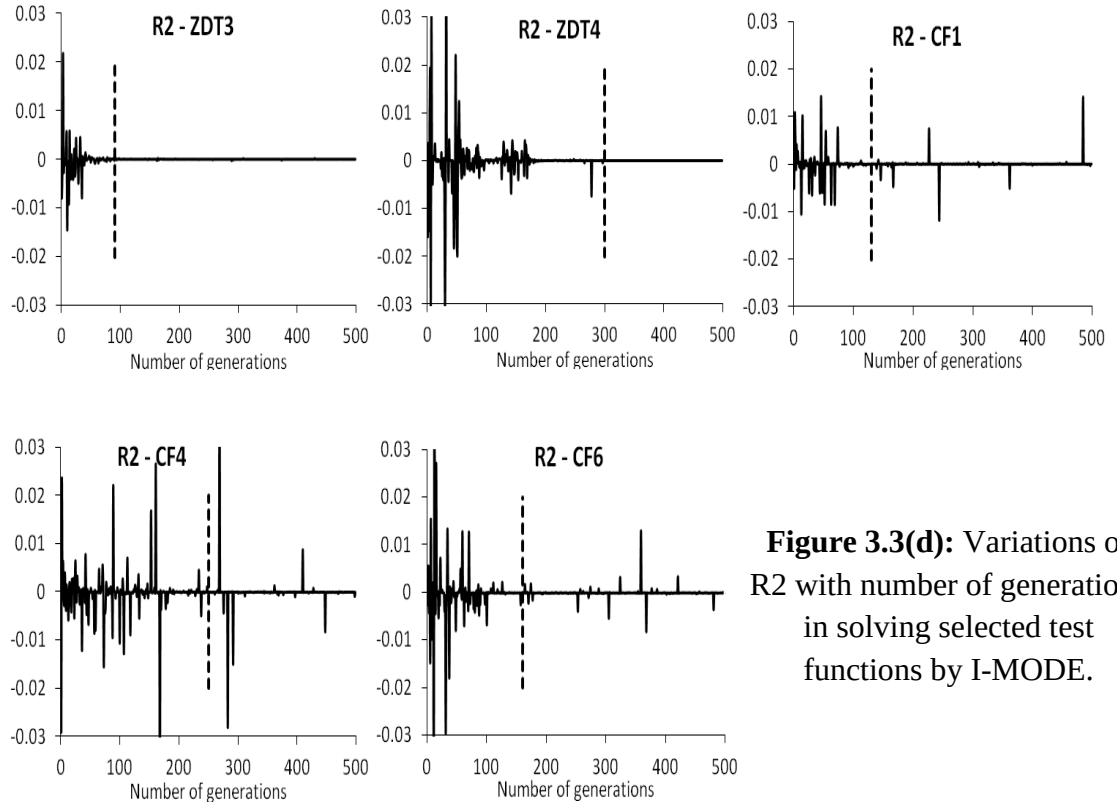


Figure 3.3(d): Variations of R2 with number of generations in solving selected test functions by I-MODE.

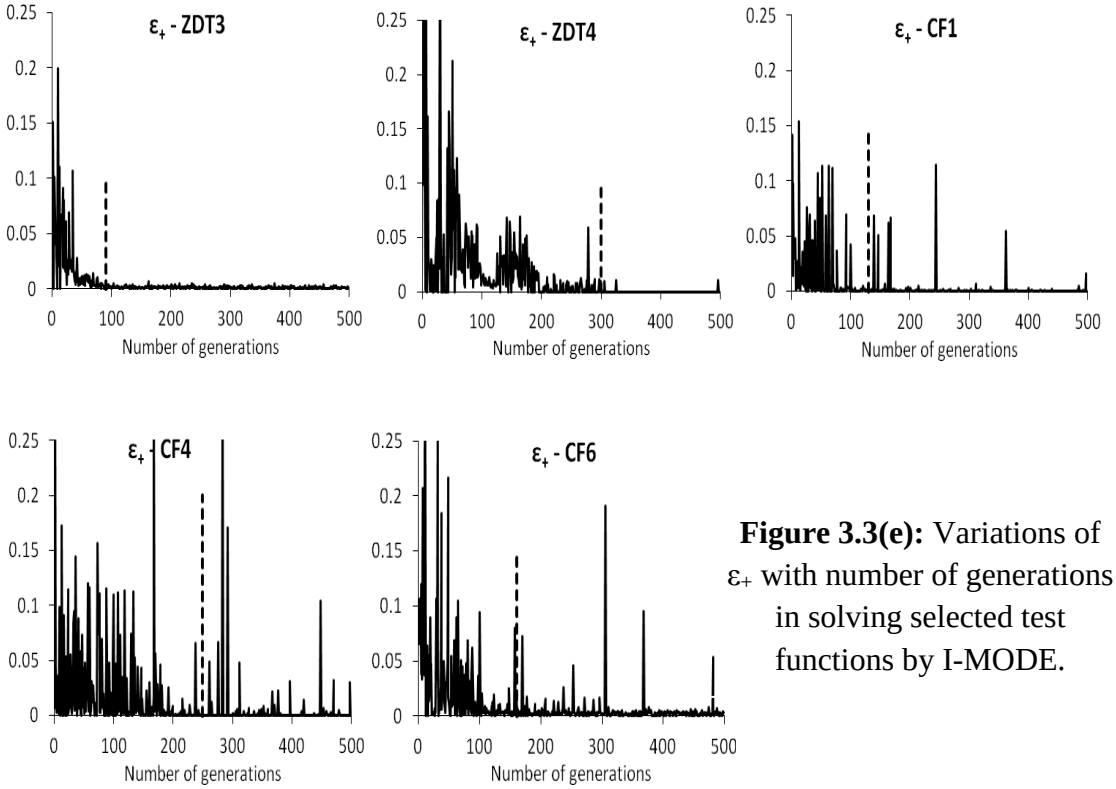


Figure 3.3(e): Variations of ε_+ with number of generations in solving selected test functions by I-MODE.

In the case of ZDT4, CF1 and CF4 test functions, R2 indicator points convergence early, around generations 190, 90 and 190 respectively, compared to the actual ones (Figure 3.3d). Finally, variation in ε_+ indicator is fluctuating for constrained functions even after the required generations to reach the global solutions (Figure 3.3e). Further, ε_+ indicator represents minimum weakly dominance of all non-dominated solutions whereas GD considers average improvement between the obtained Pareto-optimal fronts from two consecutive generations. Hence, ε_+ indicator is more sensitive compared to GD.

Both GD and SP have moderate computational complexity, and collectively they consider two important qualities of the obtained Pareto-optimal front, i.e., convergence of the search and distribution of non-dominated solutions along the obtained Pareto-optimal front. Further, GD has relatively small values, and SP fluctuates within narrow range after the required number of generations for

convergence to the optimal solutions. On the other hand, use of HVR and R2 for termination may lead to pre-mature stopping of the algorithm and hence poorer results. The ε_+ indicator value is zero if the best Pareto-optimal fronts obtained in two consecutive generations have one or more common solutions, and it is also more sensitive. Hence, GD and SP are selected for the development of search termination criterion.

Table 3.2: Characteristics of test functions used in this study (Zitzler et al., 2000; Zhang et al., 2009)

Test function	No. of decision variables	No. of constraints	No of objective functions
ZDT3	10	0	2
ZDT4	10	0	2
CF1	10	1	2
CF2	10	1	2
CF3	10	1	2
CF4	10	1	2
CF5	10	1	2
CF6	10	2	2
CF7	10	2	2
CF8	10	1	3
CF9	10	1	3

* http://www.cs.cinvestav.mx/~emoobook/
http://dces.essex.ac.uk/staff/zhang/MOEAcompetition/cec09testproblem0904.pdf

3.5 Search Termination Criterion (TC) using GD and SP Metrics

GD and SP values obtained in the recent generations are statistically checked for their variations; for this, χ^2-test is performed to check the variance of a sample (obtained in recent generations) compared to the specified tolerance. Global search is terminated if the variance of the performance metric is below some specified value,

and this condition should be fulfilled for both GD and SP individually at the same time. GD and SP values in the latest λ generations are used to apply χ^2-test, similar to Wagner et al. (2009).

$$\text{Chi} = \frac{\text{Variance}[\text{PM}, \text{PM}_2, \ldots, \text{PM}_\lambda](\lambda - 1)}{\delta_{\text{PM}}^2} \quad (3.14a)$$

$$p(\text{PM}) = \chi^2(\text{Chi}, \lambda\text{-}1) \quad (3.14b)$$

Here, PM is the performance metric (i.e., GD or SP), and δ_{PM} is the specified tolerance for standard deviation of PM. p is the probability that χ^2-test is supporting the hypothesis that variance of PM is lower than the specified tolerance (i.e., δ_{PM}^2). If this probability is more than 99% for GD and SP individually, then global search is terminated. To avoid indefinite looping, termination criterion based on the MNGs is also used.

A value of 10 is used for λ, which means that GD and SP values obtained in the last 10 (i.e., current and 9 previous) generations are used for χ^2-test. GD and SP are calculated from 2nd generation onwards, and first value of probability can be obtained in 11th generation. Initially, 0.0001 and 0.05 are used for δ_{GD} and δ_{SP}; later, a detailed study is conducted to obtain suitable value of termination parameters. Similarly, taboo radius of 0.01 is used; effect of taboo radius is studied at the end of this study.

Figure 3.4 presents the probability of supporting χ^2-test hypothesis for GD and SP individually by different test functions; in this figure, GT is the number of generations for p(PM) to cross 0.99 for the first time. It can be seen from Figure 3.4 that GD and SP do not improve consistently; sometimes their improvements become very slow (probability, p reaches near to 1) and then some improvement occurs in the next few generations (p reaches near to 0). The proposed termination criterion terminates

search at 99, 293, 134, 249 and 154 generations for ZDT3, ZDT4, CF1, CF4 and CF6 test problems, respectively. These are comparable to the number of generations required for converging to the true Pareto-optimal front stated earlier (i.e., nearly 90, 300, 130, 250 and 160 generations for ZDT3, ZDT4, CF1, CF4 and CF6, respectively). This indicates that the proposed termination criterion is successful.

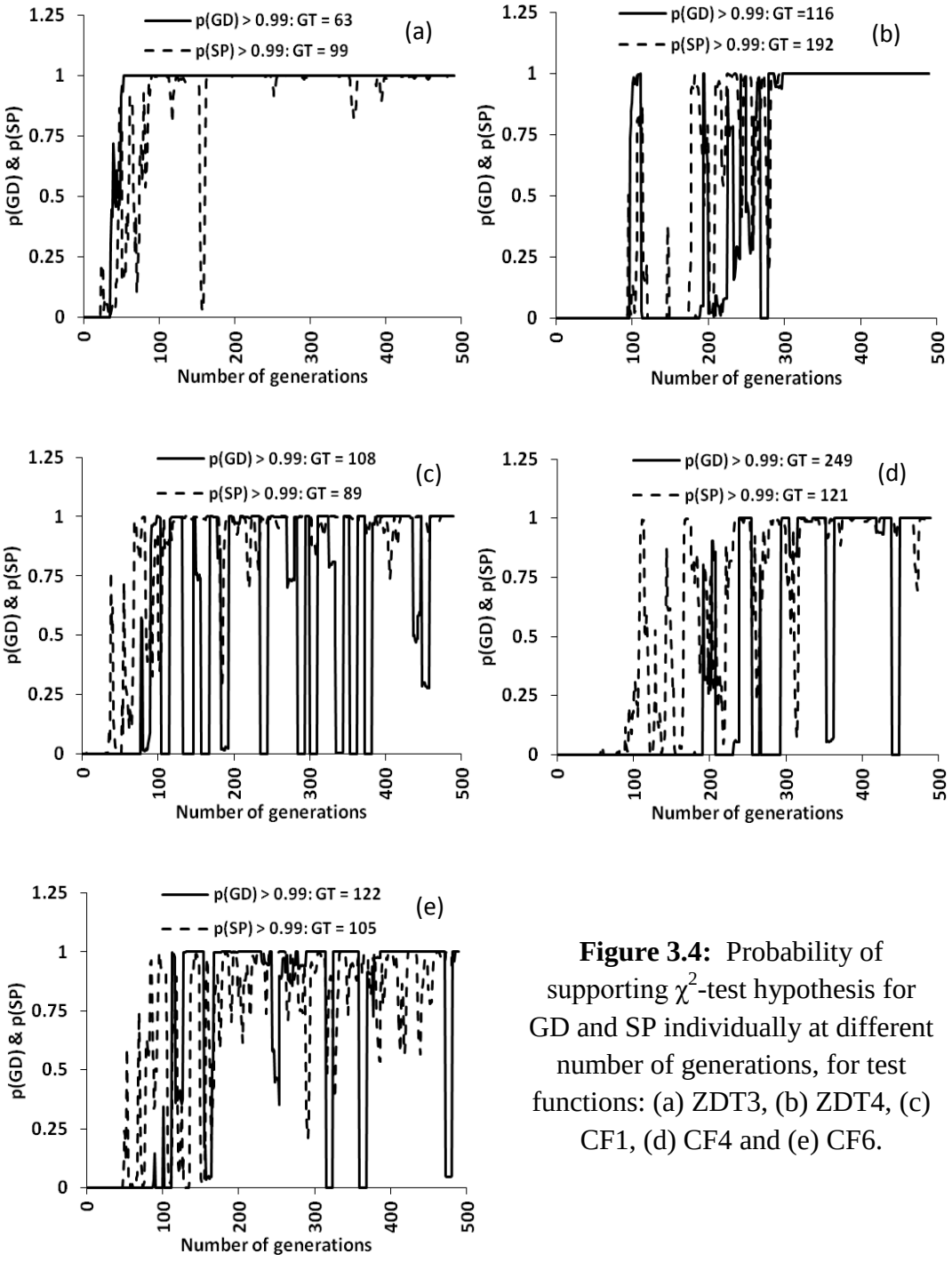


Figure 3.4: Probability of supporting χ^2-test hypothesis for GD and SP individually at different number of generations, for test functions: (a) ZDT3, (b) ZDT4, (c) CF1, (d) CF4 and (e) CF6.

3.6 I-MODE Algorithm

Figure 3.5 presents the flowchart of I-MODE algorithm. As mentioned earlier, I-MODE has three important strategies/parts: adaptation of DETL for multiple objectives, self-adaptation of parameters, and improvement based termination criterion, discussed in Sections 2, 3 and 5 respectively. Self-adaption part of I-MODE is not included in the flowchart to avoid additional complexity. Search termination criterion can be seen inside the dashed box in Figure 3.5.

3.7 Effect of Termination Parameters on I-Mode Performance

I-MODE algorithm with the proposed termination criterion is tested on 9 constrained test functions (Zhang et al., 2009) (see Table 3.2). Seven of these are bi-objective problems, while the remaining two are tri-objective problems. Size of TL is fixed at half of N, and taboo radius (TR) of 0.01 is chosen. Small values of both termination parameters can give better quality solution at the expense of large NFE. Sixteen combinations of values of δ_{GD} and δ_{SP} in the termination criterion are studied (see Tables 3 & 4); they cover all combinations of four values of δ_{GD} (i.e., 1e-5, 1e-4, 3e-4, 5e-4) and four values of δ_{SP} (i.e., 0.001, 0.01, 0.05, 0.1). Performance of I-MODE is compared with the performance of the best algorithm for CEC-2009 problems, namely, DMOEA-DD (Liu et al., 2009). Inverse generational distance (IGD^t; Zhang et al., 2009) is used as the performance metric in this comparison. Similar to GD^t, it is calculated between the true Pareto-optimal front and non-dominated solutions obtained in the objective function space, as follows.

$$IGD^t = \frac{1}{NT}\sum_{i=1}^{NT} d_i \qquad (3.15)$$

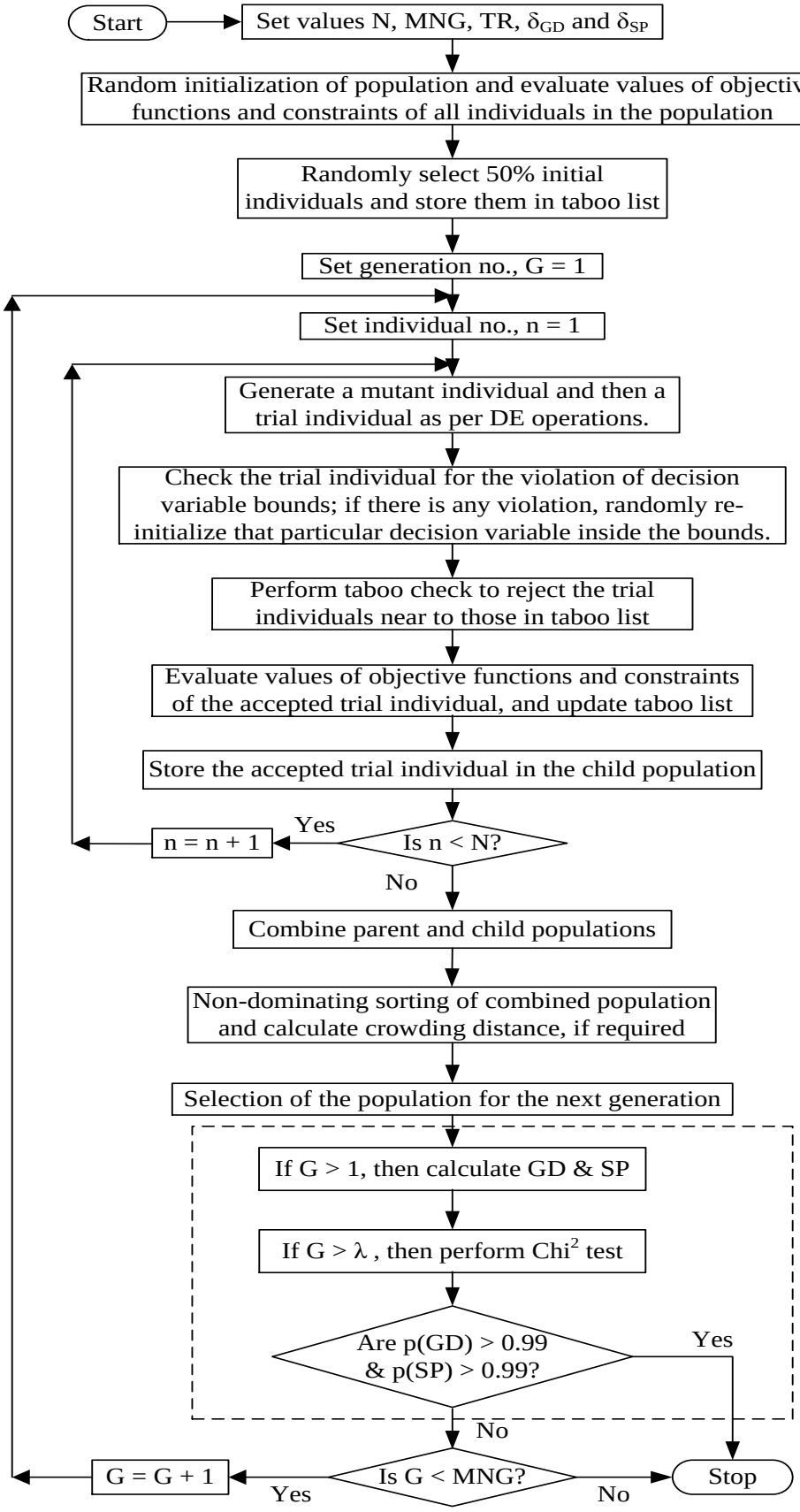


Figure 3.5: Flowchart of I-MODE algorithm.

Here, NT is the number of solutions in the true Pareto-optimal front, and d_i is the Euclidean distance of each solution in the true Pareto-optimal front to its nearest solution in the non-dominated set obtained. Objective function values of the true Pareto-optimal fronts for all constrained test functions are taken from: http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm (access date: April 2011).

A maximum of 300,000 function evaluations are used by DMOEA-DD algorithm on all constrained test functions, and the reported IGD^t values using DMOEA-DD are based on maximum of 100 and 150 non-dominated individuals for two- and three-objective problems respectively (Liu et al., 2009). In this work, population size and maximum number of generations are 200 and 1000 respectively, for all test functions. From the final optimal solutions obtained, 100 and 150 least crowded non-dominated individuals are used to calculate IGD^t for two- and three-objective problems respectively, for comparison of I-MODE with DMOEA-DD. I-MODE algorithm is terminated using the developed termination criterion.

For a set of δ_{GD} and δ_{SP} values, if I-MODE algorithm is terminated before it reaches the MNGs and the obtained IGD^t value is better than its reported mean value from DMOEA-DD, then it is considered to be a successful run. Since random numbers are employed in I-MODE, 30 independent runs, each time starting with a different random number seed, are performed on each test function. Number of successful runs (NSR), mean IGD^t ($\mu_{IGD}{}^t$) and mean number of function evaluations (μ_{NFE}) values for successful runs in these 30 runs for each problem is reported in Tables 3 and 4 for two- and three-objective problems respectively. Taboo radius, TR = 0.01 is used for these results.

Table 3.3: Effect of termination parameter values on I-MODE performance for two objective test functions

		δ_{GD} = 1e-5				δ_{GD} = 1e-4				δ_{GD} = 3e-4				δ_{GD} = 5e-4			
	δ_{SP} =	0.001	0.01	0.05	0. 1	0.001	0.01	0.05	0. 1	0.001	0.01	0.05	0. 1	0.001	0.01	0.05	0.1
CF-1	NSR	0	20	14	13	0	22	8	9	0	22	10	10	0	22	13	8
	$\mu_{IGD}{}^{t}$	0.00000	0.00295	0.00529	0.00544	0.00000	0.00330	0.00679	0.00536	0.00000	0.00354	0.00650	0.00670	0.00000	0.00354	0.00717	0.00691
	μ_{NFE}	0	148446	92395	91740	0	134083	32965	28890	0	134340	23030	22770	0	134340	22304	20318
CF-2	NSR	0	10	30	30	0	16	30	30	0	17	30	29	0	17	30	29
	$\mu_{IGD}{}^{t}$	0.00000	0.00031	0.00059	0.00064	0.00000	0.00050	0.00053	0.00049	0.00000	0.00054	0.00054	0.00070	0.00000	0.00054	0.00059	0.00062
	μ_{NFE}	0	143147	54292	49310	0	127149	38553	30709	0	121449	33255	23594	0	121449	32373	21842
CF-3	NSR	5	7	13	21	5	9	16	23	6	10	23	27	6	10	25	27
	$\mu_{IGD}{}^{t}$	0.00317	0.00315	0.00344	0.00395	0.00317	0.00317	0.00343	0.00403	0.00322	0.00320	0.00352	0.00389	0.00322	0.00320	0.00350	0.00364
	μ_{NFE}	63748	64852	65697	62797	63748	67418	65057	61999	66534	68714	64668	60237	66534	68714	63391	56760
CF-4	NSR	0	11	30	30	0	17	30	30	0	19	30	29	0	19	30	29
	$\mu_{IGD}{}^{t}$	0.00000	0.00045	0.00043	0.00052	0.00000	0.00046	0.00037	0.00056	0.00000	0.00037	0.00048	0.00057	0.00000	0.00037	0.00058	0.00056
	μ_{NFE}	0	131320	68008	64132	0	122203	45863	39577	0	120835	41305	34735	0	120835	37533	31187
CF-5	NSR	5	7	29	30	5	10	30	30	5	12	30	30	5	14	30	29
	$\mu_{IGD}{}^{t}$	0.00176	0.00158	0.00128	0.00139	0.00176	0.00158	0.00183	0.00151	0.00176	0.00159	0.00206	0.00149	0.00176	0.00157	0.00203	0.00152
	μ_{NFE}	67021	79708	77226	65691	67021	91147	68116	54774	67021	83625	60144	46623	67021	89097	54944	45131
CF-6	NSR	0	0	14	25	0	3	30	30	0	3	30	30	0	3	30	30
	$\mu_{IGD}{}^{t}$	0.00000	0.00000	0.00158	0.00175	0.00000	0.00169	0.00193	0.00196	0.00000	0.00169	0.00180	0.00180	0.00000	0.00169	0.00176	0.00186
	μ_{NFE}	0	0	108638	90844	0	134310	32979	24191	0	134310	25042	18671	0	134310	23864	16435
CF-7	NSR	3	16	29	29	3	21	30	30	3	21	30	30	3	22	30	30
	$\mu_{IGD}{}^{t}$	0.00083	0.00110	0.00116	0.00117	0.00083	0.00139	0.00108	0.00129	0.00083	0.00134	0.00119	0.00144	0.00083	0.00128	0.00129	0.00126
	μ_{NFE}	47899	86700	72030	64372	47899	92655	61257	49409	47899	92029	56863	42241	47899	91851	51543	41261
Total	PSR	6.2	33.8	75.7	84.8	6.2	46.7	82.9	86.7	6.7	49.5	87.1	88.1	6.7	51.0	89.5	86.7
	Avg_{NFE}	59556	109029	76898	69841	59556	109852	49256	41364	60485	107900	43472	35553	60485	108657	40850	33276

Table 3.3 contains percentage of successful runs (PSR) and average of μ_{NFE} (Avg_{NFE}) for successful runs, for all two-objective problems tested. In other words, PSR is the percentage of successful runs for all problems where obtained IGD^t value using I-MODE is better than the reported mean IGD^t value for DMOEA-DD. A set of values for the termination parameters is better if it gives larger PSR and requires smaller average of Avg_{NFE}.

Optimal values of termination parameters should give high value of PSR using fewer NFE, without compromising the quality of non-dominated solutions obtained. Small value of δ_{SP} (= 0.001) gives very low value (~ 6) of PSR for all δ_{GD} values (namely,1e-5, 1e-4, 3e-4, 5e-4) tested (Table 3.3); high δ_{GD} value cannot improve the PSR, which means that variance in the obtained values of SP in different generations is more than 0.001. Further, small value of δ_{GD} (= 1e-5) in conjunction with high value of δ_{SP} can give reasonably good PSR (75.7 and 84.8 for δ_{SP} of 0.05 and 0.1 respectively). It can be seen from Table 3.3 that δ_{SP} in the range of 0.05 to 0.1 can give high PSR of 75 to 90 for different values of δ_{GD}; small δ_{GD} value gives small values of $\mu_{IGD}{}^t$ and PSR at the expense of larger Avg_{NFE}.

For three-objective constrained test functions, six sets of termination parameters are able to give some successful runs (see Table 3.4); remaining ten sets of termination parameters are not satisfying the χ^2-test statistics in any of the 30 runs (i.e., NSR = 0). Hence, these are not shown in Table 3.4 for brevity. As expected, termination parameters with larger magnitude are more successful due to 3-D objective space. Sometimes, I-MODE terminates successfully before it reaches the MNGs. Successful termination of I-MODE algorithm saves additional NFEs used by DMOEA-DD (which has MNGs as the termination criterion). So, I-MODE is efficient to locate and reach the global Pareto-optimal front. Suitable δ_{GD} and δ_{SP} values can be

decided based on the required quality and efficiency of solution. Some optimization problems are multi-modal in nature; hence slightly conservative values of termination parameters will be good for a variety of optimization problems. Hence, δ_{GD} and δ_{SP} values of 3e-4 and 0.1 are appropriate for both two- and three-objective constrained problems.

Table 3.4: Effect of termination parameter values on I-MODE performance for tri-objective constrained test functions

		δ_{GD} = 1e-4		δ_{GD} = 3e-4		δ_{GD} = 5e-4	
	δ_{SP} =	0.05	0. 1	0.05	0. 1	0.05	0.1
CF-8	NSR	1	1	21	28	30	30
	$\mu_{IGD}{}^{t}$	0.00009	0.00009	0.00058	0.00051	0.00054	0.00060
	μ_{NFE}	187996	187996	74533	62632	52548	36792
CF-9	NSR	0	3	30	30	30	30
	$\mu_{IGD}{}^{t}$	0.00000	0.00014	0.00011	0.00015	0.00011	0.00014
	μ_{NFE}	0	123482	56569	37943	36330	19975

3.8 Effect of Taboo Radius on I-Mode Performance

A high value of TR is useful for exploring the search space thoroughly to locate the global optimum efficiently and reliably. On the other hand, a low value of TR is required to reach the Pareto-optimal solutions precisely. In this section, the effect of four values of TR (i.e., 0.0, 0.01, 0.05 and 0.1) on the performance of I-MODE is studied. Size of taboo list is fixed at half the population size (= N/2), same as in the previous section. PSR and Avg_{NFE} for solving bi-objective constrained test functions by I-MODE using different values of TR and termination parameters are presented in Table 3.5. These results are based on 30 runs. Performance of I-MODE with δ_{SP} of 0.001 and different values of δ_{GD} is not presented in Table 3.5 as it gives very small PSR (see Table 3.3).

Effect of different values of termination parameters on the performance of I-MODE with TR of 0.01 has been explored in the previous section (see Tables 3 and 4); remaining three values of TR (i.e., 0.0, 0.05, and 0.1) along with different values of δ_{GD} and δ_{SP} have similar performance (see Table 3.5). It can be seen that I-MODE algorithm with larger TR gives higher PSR in many cases, which proves the better exploration of search space. Also, I-MODE generally took fewer NFEs (i.e., Avg_{NFE}) as TR increases in the range tested. Hence, TR between 0.01 and 0.05 is suitable for the problems studied.

Table 3.5: Effect of taboo radius on PSR and AVG_{NFE} (based on successful runs) for bi-objective constrained test functions

		TR = 0.0		TR = 0.01		TR = 0.05		TR = 0.1	
δ_{GD}	δ_{SP}	PSR	Avg_{NFE}	PSR	Avg_{NFE}	PSR	Avg_{NFE}	PSR	Avg_{NFE}
1e-5	0.01	23.3	126482	33.8	109029	51.0	93860	68.1	80170
	0.05	74.8	78787	75.7	76898	90.5	63192	91.4	52270
	0. 1	79.0	72090	84.8	69841	91.9	58418	90.5	49751
1e-4	0.01	36.2	133540	46.7	109852	59.5	87020	76.7	71844
	0.05	84.8	48906	82.9	49256	89.0	43457	86.7	37117
	0. 1	82.4	41681	86.7	41364	87.6	36785	86.2	31984
3e-4	0.01	37.6	132559	49.5	107900	63.3	86177	78.1	70550
	0.05	84.3	45127	87.1	43472	89.0	38845	85.7	32916
	0. 1	80.0	36444	88.1	35553	87.6	31339	84.8	26514
5e-4	0.01	38.1	131670	51.0	108657	64.8	85693	78.6	70643
	0.05	83.3	42898	89.5	40850	89.5	37056	85.2	31086
	0. 1	81.0	33805	86.7	33276	86.7	28731	83.8	24664

Table 3.6 presents values of μ_{IGD}^t and σ_{IGD}^t after MNGs (= 1000) using I-MODE algorithm with TR of 0.01, based on 30 runs with different random seeds. NFEs for different problems for 1000 generations, given in Table 3.6, are less than 200,000 expected for a population of 200 due to the use of taboo list and check in I-MODE.

Values of μ_{IGD}^t and σ_{IGD}^t by DMOEA-DD (Liu et al., 2009) are also given in Table 3.6 for comparison; these results are using fixed NFEs of 300,000. It is clear from Table 3.6 that I-MODE gives smaller μ_{IGD}^t values, and also uses fewer function evaluations than DMOEA-DD for all the problems tested.

To illustrate the importance and benefit of termination criteria, μ_{IGD}^t, σ_{IGD}^t, and μ_{NFE} using I-MODE algorithm with the selected termination parameters (δ_{GD} = 3e-4 and δ_{SP} = 0.1) are also presented in Table 3.6. These termination parameter values are effective to stop the search in all 30 runs on all 9 problems except for CF-3 and CF-8 for which the search was stopped in 27 and 28 runs respectively. Generally, μ_{IGD}^t values using the proposed termination criterion are slightly worse compared to those obtained with MNGs = 1000, but they are satisfactory for engineering applications and require significantly fewer NFEs (Table 3.6). Hence, the inclusion of the proposed termination criterion in the search algorithm makes it computationally very efficient and yet provides satisfactory results.

In addition to IGD^t values, equation 3.7 is used to calculate GD^t values. Table 3.6 presents μ_{GD}^t and σ_{GD}^t for different test problems. Notably, values of μ_{IGD}^t and μ_{GD}^t are very much different for three-objective test functions although they are comparable for five bi-objective problems. Some of the non-dominated solutions obtained can be ignored in IGD^t calculation due to availability of other nearby non-dominated solutions in the obtained Pareto-optimal front while all non-dominated solutions have to be considered in GD^t calculation. This is probably the reason for differences in values of μ_{IGD}^t and μ_{GD}^t.

Table 3.6: Mean-IGD^t and Sigma-IGD^t values (over 30 runs) using I-MODE and comparison with DMOEA-DD

	DMOEA-DD		I-MODE Using TR = 0.01							
			MNGs = 1000			Termination Criterion (δ_{GD} = 3e-4 & δ_{SP} = 0.1)			MNGs = 1000	
	NFE = 300,000									
	μ_{IGD}^t	σ_{IGD}^t	μ_{IGD}^t	σ_{IGD}^t	μ_{NFE}	μ_{IGD}^t	σ_{IGD}^t	μ_{NFE}	μ_{GD}^t	σ_{GD}^t
CF-1	0.0113	0.00276	0.0076	0.00848	183902	0.0384	0.03483	21409	0.0055	0.0024
CF-2	0.002	0.00045	0.0004	0.00049	194336	0.00078	0.00061	23579	0.0054	0.0074
CF-3	0.0563	0.00757	0.0037	0.00126	83554	0.0039	0.00155	60237	0.0694	0.0454
CF-4	0.007	0.00146	0.0005	0.00054	173847	0.0009	0.00187	34649	0.0010	0.0007
CF-5	0.0158	0.00666	0.0016	0.00085	116862	0.0015	0.00076	46623	0.0138	0.016
CF-6	0.0150	0.00646	0.0016	0.00049	189432	0.0018	0.00051	18671	0.0007	0.0005
CF-7	0.0191	0.00612	0.0013	0.00139	117282	0.0014	0.00157	42241	0.0099	0.0133
CF-8	0.1434	0.02142	0.0005	0.00087	196615	0.0005	0.00138	62632	0.2371	0.2701
CF-9	0.1621	0.03162	0.0001	0.00007	196456	0.0002	0.00007	37943	0.0605	0.0240

3.9 General Discussion

The developed I-MODE algorithm is suitable for application problems. In this work, algorithm parameters are self-adapted to avoid their tuning for different problems. Further, I-MODE has termination criterion for making a timely decision on search termination. Recommended values (δ_{GD} = 0.0003, δ_{SP} = 0.1, based on this study) of termination parameters can be used for a new optimization problem. Similar to other stochastic methods, I-MODE can solve problems with moderate number (about 30) of decision variables. If the optimization problem has equality constraints, these can be converted into inequality constraints. But, it would be good to use sequential solution approach for better performance of algorithm. Small taboo radius value can be used for an optimization problem with fewer decision variables, whereas large taboo radius value can improve exploration capabilities of the algorithm for

optimization problems with large number of decision variables. I-MODE algorithm can easily be used for black box problems.

3.10 Conclusions

In this chapter, I-MDOE has been developed, to solve MOO problems, by adapting computationally efficient DETL. Five performance metrics for MOO are studied for their variations with generations, and it is found that GD and SP can quantify the improvement in the Pareto-optimal front efficiently and consistently. Hence, a termination criterion for stopping the search algorithm based on GD and SP in the latest generations is developed and tested on constrained test functions. It does not use information about the true Pareto-optimal front, which makes it suitable for real world applications. A detailed study has been conducted for different combinations of termination parameters, and suitable values for these parameters are suggested for constrained optimization problems. Use of taboo list/check improves reliability and efficiency for larger values of taboo radius, but it may reduce exploitation of search space. Accordingly, obtained IGD^t values using I-MODE with a larger taboo radius are slightly worse than those obtained with a smaller value of taboo radius, but the former are still satisfactory for engineering applications. Compared to the results by DMOEA-DD (Liu et al., 2009), I-MODE gives smaller IGD^t values, and also uses fewer function evaluations for the problems tested. Further, use of the proposed termination criterion in I-MODE reduces function evaluations even more.

Chapter 4

Use of Termination Criterion with Other Algorithms

4.1 Introduction

In this chapter, effectiveness of the proposed termination criterion is tested with other algorithms. The developed search termination criterion, based on the improvement in the Pareto-optimal front, has been used to check convergence of NGSA-II with the selected jumping gene adaptations. Additionally, a hybrid method may give both high reliability and computational efficiency; hence stochastic global search is integrated with deterministic local search, using termination criterion.

The elitist non-dominated sorting genetic algorithm (NSGA-II) has been used to optimize many process design and operation problems for two or more objectives. In order to improve its performance, jumping gene concept from natural genetics has been incorporated in NSGA-II. Several jumping gene adaptations have been proposed and used to solve mathematical and application problems in different studies. In this chapter, four jumping gene adaptations are selected and comprehensively evaluated on a number of two-objective unconstrained and constrained test functions. Three quality metrics, namely, generational distance, spread and inverse generational distance are employed to evaluate the distribution and convergence of the obtained Pareto-optimal front.

Stochastic optimization algorithms can explore the search space comprehensively, but they cannot locate global optimum precisely although solutions obtained by stochastic search are satisfactory for engineering applications. In the last generations of search, these algorithms become very slow due to the similarity of information content in most of the individuals. On the other hand, deterministic methods are more

efficient in finding the precise optimum, particularly if a very good initial estimate is given. Hence, they can be used for local refinement of each Pareto-optimal point obtained after stochastic global search. This refinement can be applied on all or some selected solutions from the Pareto-optimal front. Several deterministic methods (e.g., PDM, NBI and NNC) can be potentially used as local search after stochastic global search. These methods are reviewed in Chapter 2 of this thesis. NNC method can be used for problems with discontinuous Pareto-optimal front; it does not assign any weights to different objectives, rather it includes additional inequality constraints in the problem formulation (Messac et al., 2003). Hence, in this work, NNC method is chosen and used to refine the non-dominated solutions obtained by the global search.

The rest of this chapter is organized as follows. The next section of this chapter briefly discusses different variants of JG adaptations and their applications. It also provides details on constraint handling and program implementation including values of algorithm parameters. Finally, performance of selected JG adaptations on many test functions is compared in the same section. Section 4.3 describes NNC method with termination criterion, and presents performance of hybrid search on several test functions. Finally, useful findings of this work are summarized in the last section of this chapter.

4.2 Jumping Gene Adaptations of NSGA-II[3]

Deb et al. (2002) have developed the elitist non-dominated sorting genetic algorithm (NSGA-II) for solving multi-objective optimization (MOO) problems, which has found many applications in Chemical Engineering. In order to improve the performance of the binary-coded NSGA-II algorithm, Kasat and Gupta (2003) have

[3] This section is based on the book chapter: Sharma, S., Nabavi, S. R. and Rangaiah, G. P. (2013), Performance comparison of jumping gene adaptations of elitist non-dominated sorting genetic algorithm, MOO in Chemical Engineering: Developments and Applications, Wiley

included the jumping gene (JG) operator in it. Following this, several variants of JG adaptations have been developed, and applied to solve a number of application problems. Guria et al. (2005a) have developed one variant of JG adaptation, referred as mJG, for problems having the optimal solutions near to decision variable bounds. Bhat et al. (2006) have proposed aJG variant, which was later used with NSGA-II in Bhat (2007). Agarwal and Gupta (2008a) have suggested two new variants of JG adaptations, namely, sJG and saJG, and studied them with binary-coded NSGA-II. NSGA-II-saJG can only be applied if number of binaries used for representing each decision variable is same, whereas NSGA-II-sJG completely replaces part of the chromosome associated with a particular decision variable and so can be used even if the number of binaries used is not same for different variables. Also, Agarwal and Gupta (2008a) compared four variants of JG adaptations (namely, JG, aJG, sJG and saJG) on three unconstrained test functions. Set convergence ratio, spacing and maximum spread are used as performance indicators, and it was found that performance of NSGA-II-aJG, NSGA-II-sJG and NSGA-II-saJG is comparable, whereas NSGA-II-JG is outperformed by the other three.

Ramteke and Gupta (2009a) have discussed and evaluated five variants of JG adaptations, namely, NSGA-II-JG/mJG/aJG/saJG/sJG on three unconstrained test functions. Recently, two more variants of JG adaptations, namely, Alt-NSGA-II-aJG (Ramteke and Gupta, 2009b) and biogenetic-NSGA-II-aJG (Ramteke and Gupta, 2009c) were proposed. Alt-NSGA-II-aJG mimics biological altruism from honey bee to solve MOO problems. In biogenetic-NSGA-II-aJG, information/solution from an earlier optimization problem is used to solve the modified/new optimization problem. This strategy can be used with other JG variants also, and is relevant for modifications

in industrial optimization problems, such as increase in number of objectives, decision variables and/or ranges of decision variables.

In order to speed up the convergence, Ripon et al. (2007) and Furtuna et al. (2011) applied the concept of jumping gene in real coded NSGA-II. In the work of Ripon et al. (RJGGA; 2007), part of the chromosome (i.e., transposon) is copied/cut and pasted into the same or different chromosome. Its performance has been compared with seven MOO algorithms on five test functions, using set convergence metric, spacing, spread and hyper volume as performance metrics. Mostly, RJGGA performed better than other algorithms, in terms of diversity of non-dominated solutions and convergence to the known Pareto-optimal front. Furtuna et al. (2011) adapted the JG proposed by Kasat and Gupta (2003) for the real coded NSGA-II. Additionally, JG and aJG variants (of binary coded NSGA-II) are also used with multi-objective simulated annealing (Sankararao and Gupta, 2006, 2007a and 2007b). So, jumping gene concept has potential for use with other MOO algorithms.

4.2.1 Use of JG Adaptations to Solve Application Problems

Several researchers have successfully used one or more variants of JG adaptations to solve different application problems; these applications are summarized in Table 4.1. Mathematical functions tested in these studies are also included in this table. Although many applications have been studied using JG adaptations, their evaluation using mathematical functions is limited, as can be seen in Table 4.1.

4.2.2 Selection of JG Adaptations for Comparison

There are a number of JG adaptations proposed and applied to Chemical Engineering problems in the past decade, and some of these are compared on a

limited number of problems (Agarwal and Gupta, 2008a; Ramteke and Gupta, 2009a). However, there has been no comprehensive and systematic evaluation of these adaptations.

Further, NSGA-II has been popular for solving application problems. Hence, this chapter analyzes and compares the performance of four variants of JG adaptations, namely, NSGA-II-aJG, NSGA-II-saJG, NSGA-II-sJG and Alt-NSGA-II-aJG, for bi-objective optimization problems. In this comparison, NSGA-II-mJG and biogenetic-NSGA-II-aJG are not considered since the former's applicability is for a specific type of optimization problems and the latter is similar to NSGA-II-aJG except for the difference in the approach which can be used with other adaptations/algorithms as well. A detailed flow-chart of NSGA-II with JG adaptation for MOO is given in Figure 4.1. More details on NSGA-II can be found in Deb (2001).

A multi-objective optimization for M different objectives: f_1, f_2, …, f_M, can be mathematically stated as follows.

$$\text{Min.} \qquad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots f_M(\mathbf{x})\} \tag{4.1a}$$

$$\text{Subject to} \qquad \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \text{ and } \mathbf{g}(\mathbf{x}) \leq \mathbf{b} \tag{4.1b}$$

Here, $\mathbf{x}$ is the vector of decision variables between lower (i.e., $\mathbf{x}^L$) and upper (i.e, $\mathbf{x}^U$) bounds; $\mathbf{g}$ is the set of inequality constraints where $\mathbf{b}$ is the vector of constants. If an optimization problem has equality constraints, then those can be converted into inequality constraints by relaxation.

Penalty function and feasibility criterion are the two popular approaches for handling constraints within evolutionary algorithms. In this work, penalty function approach is used to handle inequality constraints. In this approach, objective functions are penalized (i.e., modified) by adding a penalty term to each of the original objective functions, as follows.

Table 4.1: Use of different JG adaptations to solve application problems

Application	Test functions	Algorithm(s)	Reference(s)
Fluidized-bed catalytic cracker	Schaffer, ZDT4	NSGA-II and NSGA-II-JG	Kasat and Gupta (2003)
Flotation circuits	-	NSGA-II (binary & real coded) and NSGA-II-mJG	Guria et al. (2005a & 2006)
Reverse osmosis desalination units	-	NSGA-II, NSGA-II-JG and aJG	Guria et al. (2005b)
Industrial steam reformer	-	MOSA, MOSA-JG, MOSA-aJG, NSGA-II, NSGA-II-JG, NSGA-II-aJG	Sankararao and Gupta (2006)
Low density poly-ethylene reactor	CONSTR, SRN, TNK, WATER	NSGA-II, NSGA-II-JG and aJG	Agrawal et al. (2006 & 2007)
Fuel oil blending	-	NSGA-II, NSGA-II-JG and aJG	Khosla et al. (2007)
Pressure swing adsorbers for air separation	-	MOSA-aJG	Sankararao and Gupta (2007a)
Fluidized-bed catalytic cracker	ZDT4, two more test functions	NSGA-II, NSGA-II-JG, NSGA-II-aJG, MOSA-JG, MOSA-aJG	Sankararao and Gupta (2007b)
Shell and tube heat exchangers	ZDT2, ZDT3, ZDT4	NSGA-II-JG, aJG, sJG and saJG	Agarwal and Gupta (2008a & 2008b)
Industrial phthalic anhydride (PA) reactor	-	NSGA-II-aJG and guided NSGA-II-aJG	Bhat and Gupta (2008)
Nylon-6 semi batch reactor	-	NSGA-II, NSGA-II-aJG, MOSA-aJG	Mitra et al. (1998), Ramteke and Gupta (2008)
Industrial PA reactor system, and simulation of cancer	ZDT2, ZDT3, ZDT4	Alt-NSGA-II-aJG	Ramteke and Gupta (2009b)
Industrial PA reactor system; nylon-6 polymerization	Two modified test functions	NSGA-II-aJG, B-NSGA-II-aJG	Ramteke and Gupta (2009c)
Liquefied petroleum gas (LPG) cracker	-	NSGA-II-aJG	Nabavi et al. (2009)
Synthesis of polymeric nano-particles	-	NSGA-II (real coded), NSGA-RJG	Furtuna et al. (2011)
Fixed bed maleic anhydride reactor	-	NSGA-II-aJG, Alt-NSGA-II-aJG	Chaudhari and Gupta (2012)

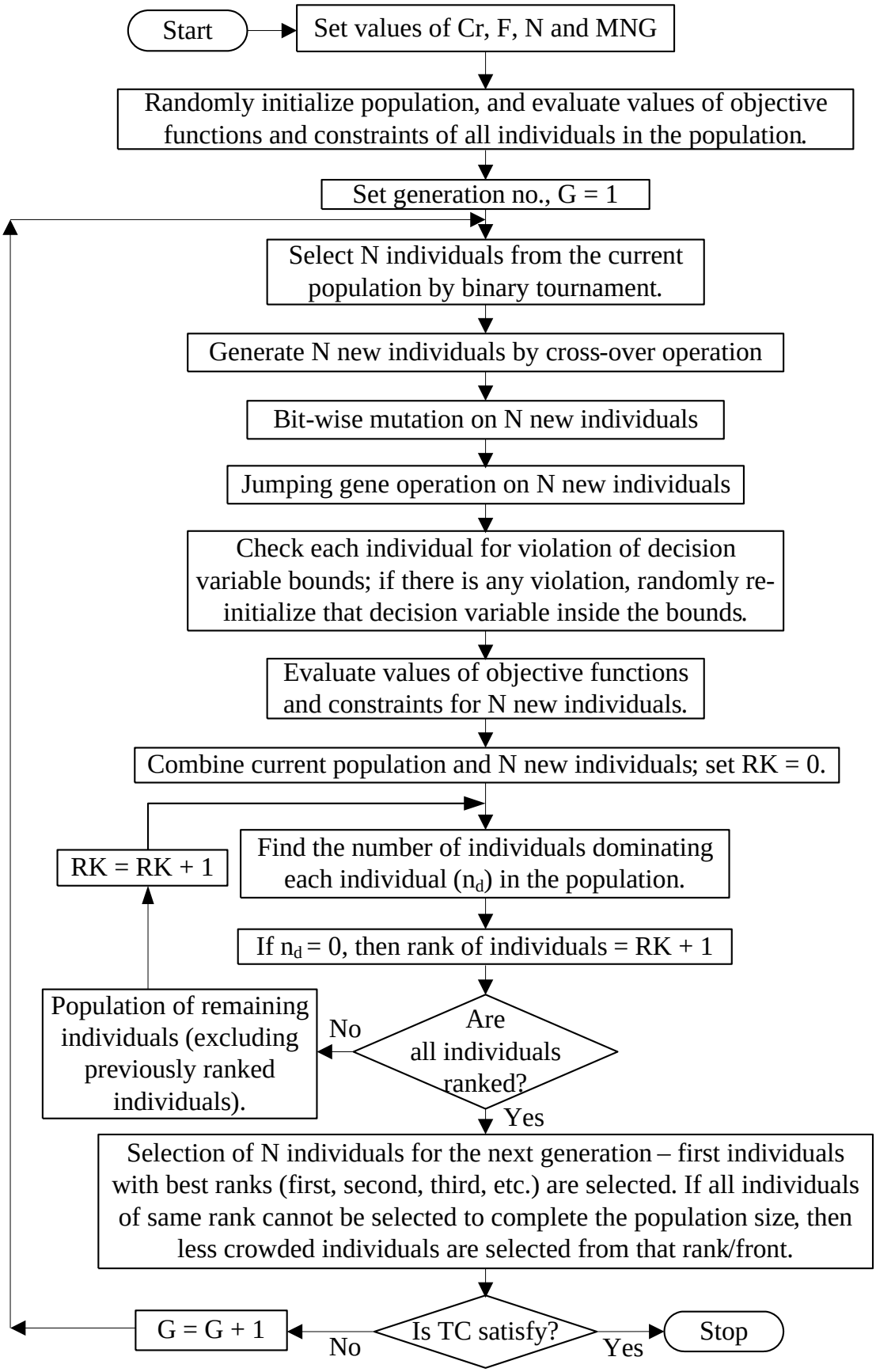


Figure 4.1: A detailed flow-chart of NSGA-II with JG adaptation for MOO

$$F_m(\mathbf{x}) = f_m(\mathbf{x}) + \sum_{j=1}^{n_i} R_j \max\left[0, G_j(\mathbf{x})\right] \quad (4.2)$$

Here, F_m and f_m are m^{th} modified and original objective function respectively, $G_j(\mathbf{x})$ is j^{th} inequality constraint (defined in equation 4.3 below), R_j is the user-defined penalty parameter for j^{th} inequality constraint and n_i is the number of inequality constraints. In order to use a single penalty parameter for all inequality constraints, they are normalized using the following transformation.

$$G_j(\mathbf{x}) \equiv g_j(\mathbf{x})/b_j - 1 \leq 0 \quad (4.3)$$

Many application problems have constraints; hence, performance of the above four JG adaptations is compared on four constrained and five unconstrained test functions. The unconstrained test functions: ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 have different characteristics like continuous or discontinuous objective functions, multi-modality and convexity of search space. ZDT1 and ZDT2 test functions have convex and non-convex Pareto-optimal front respectively. ZDT3 has several non-continuous convex parts in the Pareto-optimal front. ZDT4 is multi-modal in nature and has 99 local optimal fronts (Sindhya et al., 2011), and ZDT6 problem has non-uniform density of solutions. The constrained test functions: OSY, CONSTR, SRN, and TNK are considered for testing since many applications involve constraints. Main details of unconstrained and constrained test functions are given in Table 4.2. Search termination at the right time improves efficiency of the algorithm; hence, the termination criterion, developed in Chapter 3, is also tested with the four selected JG adaptations of NSGA-II. Furthermore, several accepted and/or applied performance metrics (GD^t, SP^t and IGD^t, as defined in Chapter 3) are used for performance comparison. As in the earlier studies, which proposed these adaptations, binary coding is used for representing variables.

Table 4.2 : Test functions studied in this work; DVs - decision variables (Deb et al., 2001; Coello Coello et al., 2007)

	No of DVs	Range of DVs	Min. $f_1(x)$	Min. $f_2(x, g(x))$	g(x)
ZDT1	30	$x_i \in [0, 1]$	x_1	$g(\mathbf{x})\left[1-\sqrt{\frac{f_1}{g(x)}}\right]$	$1+\frac{9}{n-1}\sum_{i=2}^{n} x_i$
ZDT2	30	$x_i \in [0, 1]$	x_1	$g(\mathbf{x})\left[1-\left(\frac{f_1}{g(x)}\right)^2\right]$	$1+\frac{9}{n-1}\sum_{i=2}^{n} x_i$
ZDT3	30	$x_i \in [0, 1]$	x_1	$g(\mathbf{x})\left[1-\sqrt{\frac{f_1}{g(x)}}-\frac{f_1}{g(x)}\sin(10\pi f_1)\right]$	$1+\frac{9}{n-1}\sum_{i=2}^{n} x_i$
ZDT4	10	$x_1 \in [0, 1]$, $x_i \in [-5, 5]$	x_1	$g(\mathbf{x})\left[1-\sqrt{\frac{f_1}{g(x)}}\right]$	$1+10(n-1)+\sum_{i=2}^{n}(x_i^2-10\cos(4\pi x_i))$
ZDT6	10	$x_i \in [0, 1]$	$1-\exp(-4x_1)\times \sin^6(6\pi x_1)$	$g(\mathbf{x})\left[1-\left(\frac{f_1}{g(x)}\right)^2\right]$	$1+9\left[\frac{\sum_{i=2}^{n} x_i}{9}\right]^{0.25}$

	No of DVs	Range of DVs	Min. $f_1(x)$	Min. $f_2(x)$
KUR	3	$x_i \in [-5, 5]$	$\sum_{i=1}^{n-1}\left(-10\exp\left[-0.2\sqrt{x_i^2+x_{i+1}^2}\right]\right)$	$\sum_{i=1}^{n}(\lvert x_i\rvert^2+5\sin(x_i^3))$
FON	8	$x_i \in [-2, 2]$	$1-\exp\left(-\sum_{i=1}^{8}\left[x_i-1/\sqrt{8}\right]^2\right)$	$1-\exp\left(-\sum_{i=1}^{8}\left[x_i+1/\sqrt{8}\right]^2\right)$

	Range of DVs	Objectives (Min.)	Constraints
OSY	$(x_1, x_2, x_6)\in$ [0, 10], $(x_3, x_5)\in$ [1, 5], $x_4 \in$ [0, 6],	$f_1(\mathbf{x}) = -[25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2+(x_4-4)^2+(x_5-1)^2]$	$x_1+x_2-2\geq 0$
			$-x_1-x_2+6\geq 0$
			$x_1-x_2+2\geq 0$
			$-x_1+3x_2+2\geq 0$
		$f_2(\mathbf{x}) = x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_6^2$	$-(x_3-3)^2-x_4+4\geq 0$
			$(x_5-3)^2+x_6-4\geq 0$
CONS -TR	$x_1 \in [0.1, 1]$, $x_2 \in [0, 5]$	$f_1(\mathbf{x}) = x_1$	$9x_1+x_2-6\geq 0$
		$f_2(\mathbf{x}) = (1+x_2)/x_1$	$9x_1-x_2-1\geq 0$
SRN	$(x_1, x_2) \in$ [-20, 20]	$f_1(\mathbf{x}) = (x_1-2)^2+(x_2-1)^2+2$	$-x_1^2-x_2^2+225\geq 0$
		$f_2(\mathbf{x}) = 9x_1-(x_2-1)^2$	$-x_1+3x_2-10\geq 0$
TNK	$(x_1, x_2) \in$ [0, π]	$f_1(\mathbf{x}) = x_1$	$x_1^2+x_2^2-1-0.1\cos(16\arctan(x_1/x_2))\geq 0$
		$f_2(\mathbf{x}) = x_2$	$-(x_1-0.5)^2-(x_2-0.5)^2+0.5\geq 0$

FORTRAN programs for NSGA-II-aJG and Alt-NSGA-II-aJG have been taken from http://www.iitk.ac.in/che/skg.htm (access date: October 2011), and then

modified for NSGA-II-sJG and NSGA-II-saJG. All these programs are amended to include GD and SP calculations and to implement the χ^2-test for termination, as discussed in Chapter 3. GD^t, SP^t and IGD^t calculations are also implemented to compare the obtained Pareto-optimal front with the known Pareto-optimal front. Parameters used in the termination criterion are: $\lambda = 10$, $\delta_{GD} = 0.0002$ and $\delta_{SP} = 0.03$. A fixed and large value of the penalty parameter ($R = 10^9$) is used for all constrained problems. In this study, values of parameters in these algorithms used are chosen based on the values used and/or recommendations in Agarwal and Gupta (2008a) and Ramteke and Gupta (2009b), and these are given in Table 4.3.

Table 4.3: Values of parameters in JG adaptations of NSGA-II used in this study

Parameter	**aJG**	**saJG**	**sJG**	**Alt-aJG**
N (population size)	200	200	200	200
MNG (maximum number of generations)	1500	1500	1500	1500
p_c (crossover probability)	0.9	0.9	0.9	0.9
p_m (mutation probability)	0.001	0.001	0.001	0.001
p_{JG} (JG probability)	0.5	0.5	0.5	0.5
f_b (arbitrary number used in aJG operator)	25/ 10*	-	-	25/ 10*
l_{string} (no. of bits for each decision variable)	30	30	30	30

***** For constrained problems

4.2.3 Performance Comparison on Unconstrained Test Functions

Values of GD^t and IGD^t vary significantly with generations and also for different algorithms, whereas SP^t does not change much. Hence, for ease of comparison among different algorithms, GD^t and IGD^t values are normalized using certain GD^t_{max} and IGD^t_{max}. GD^t and IGD^t values are larger in the beginning of search, and decrease slowly with the progress of search. Further, there can be some fluctuations in GD^t and

IGD^t values at the start of search due to change in the number of non-dominated solutions in the best front obtained. However, GD^t and IGD^t vary smoothly after some generations (e.g., 100). Hence, maximum values of GD^t and IGD^t obtained after 100 generations using four different algorithms are considered as GD^t_{max} and IGD^t_{max}. Table 4.4 presents GD^t_{max} and IGD^t_{max} for different unconstrained and constrained test problems. As mentioned earlier, GD^t, SP^t and IGD^t are calculated for original values of objectives, and hence significant variation can be observed from problem to problem.

Table 4.4: Maximum values of GD^t and IGD^t obtained after 100 generations, using four different algorithms

PM	**ZDT1**	**ZDT2**	**ZDT3**	**ZDT4**	**ZDT6**	**OSY**	**CONSTR**	**SRN**	**TNK**
GD^t_{max}	0.0046	0.0043	0.0038	0.0118	0.0343	1.5156	0.0008	0.0173	0.0008
IGD^t_{max}	0.1155	0.1468	0.1133	0.2631	0.7386	9.1184	0.0341	0.5498	0.0212

Table 4.5 presents GD^t/GD^t_{max}, SP^t and IGD^t/IGD^t_{max} for unconstrained test problems using four selected JG adaptations. Here, search is stopped using the termination criterion discussed in Chapter 3; termination generation (GT) of each algorithm for each problems is also given in Table 4.5. These values of performance metrics are average of 10 runs with different random number seed value (i.e., 0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95), for each problem with each algorithm. Note that random number seed value affects the series of random numbers generated, which in turn can affect performance of stochastic algorithms. Same set of random seed values and procedure are employed for testing JG adaptations on constrained problems. The best values obtained for a problem by different JG adaptations are identified in bold in Table 4.5 and subsequent table.

Table 4.5: GD^t/GD^t_{max}, SP^t and IGD^t/IGD^t_{max} for unconstrained test functions obtained by four JG adaptations; these values are average of 10 runs with random number seeds

	Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Total
GD^t/GD^t_{max}	NSGA-II-aJG	0.598	0.860	0.675	0.361	0.275	2.768
	NSGA-II-saJG	0.707	0.866	0.778	0.601	0.244	3.196
	NSGA-II-sJG	0.604	0.720	0.651	0.439	0.215	2.629
	Alt-NSGA-II-aJG	**0.030**	**0.020**	**0.131**	**0.029**	**0.003**	**0.214**
SP^t	NSGA-II-aJG	0.829	1.079	0.978	0.797	1.028	4.711
	NSGA-II-saJG	0.855	1.107	0.963	0.885	1.019	4.828
	NSGA-II-sJG	**0.802**	1.061	**0.904**	0.814	**1.002**	**4.584**
	Alt-NSGA-II-aJG	0.970	**0.983**	1.177	**0.771**	1.004	4.906
IGD^t/IGD^t_{max}	NSGA-II-aJG	0.592	0.733	0.738	0.467	0.287	2.817
	NSGA-II-saJG	0.637	0.763	0.777	0.564	0.280	3.021
	NSGA-II-sJG	0.599	0.627	0.720	0.446	0.223	2.616
	Alt-NSGA-II-aJG	**0.026**	**0.026**	**0.099**	**0.031**	**0.002**	**0.183**
GT	NSGA-II-aJG	165	136	163	**202**	466	1132
	NSGA-II-saJG	**156**	**135**	163	297	514	1265
	NSGA-II-sJG	181	159	195	251	522	1308
	Alt-NSGA-II-aJG	407	139	**116**	227	**222**	**1111**

It can be seen in Table 4.5 that Alt-NSGA-II-aJG gives smaller values of GD^t/GD^t_{max} and IGD^t/IGD^t_{max} for the unconstrained functions tested, compared to other three JG adaptations. NSGA-II-sJG and Alt-NSGA-II-aJG gives better values of SP^t on different unconstrained test problems. Alt-NSGA-II-aJG algorithm performs well on ZDT3 and ZDT6 problems, based on the closeness of the non-dominated

solutions obtained to the known Pareto-optimal front (i.e., smaller GD^t/GD^t_{max} and IGD^t/IGD^t_{max}), and also it takes smallest GT (i.e., 116 and 222). For ZDT2 and ZDT4 problems, Alt-NSGA-II-aJG is superior to the other three adaptations, based on all three performance metrics; here, generations used (i.e., 139 and 227) are also comparable to the smallest GT for these problems (i.e., 135 and 202). For ZDT1 problem, Alt-NSGA-II-aJG gives significantly smaller values of GD^t/GD^t_{max} and IGD^t/IGD^t_{max} but it takes larger number of generations (i.e., 407 compared to the smallest GT of 156). NSGA-II-aJG is the second best algorithm for solving ZDT1 problem, which gives smaller GD^t/GD^t_{max}, smaller IGD^t/IGD^t_{max} and comparable SP^t than those obtained by NSGA-II-saJG and NSGA-II-sJG; here, the required number of generations (i.e., 165) is also comparable to those used by NSGA-II-saJG and NSGA-II-sJG algorithms (i.e., 156 and 181 respectively).

Total values of GD^t/GD^t_{max}, SP^t, IGD^t/IGD^t_{max} and GT by four JG adaptations on unconstrained functions tested, are shown in the last column of Table 4.5. Overall, Alt-NSGA-II-aJG is the best, based on GD^t/GD^t_{max}, IGD^t/IGD^t_{max} and GT, among the adaptations tested. NSGA-II-sJG is better than other adaptations based on SP^t; Alt-NSGA-II-aJG gives smallest SP^t for ZDT2 and ZDT4 but relatively larger SP^t for ZDT1 and ZDT3 test functions. NSGA-II-saJG algorithm performs worse than others tested, based on all performance metrics. Figure 4.2 shows non-dominated solutions obtained by Alt-NSGA-II-aJG (best adaptation) and NSGA-II-saJG (worst adaptation) for ZDT3 and ZDT4 test functions. For ZDT3 function, the non-dominated solutions obtained by both these adaptations are closer to the known Pareto-optimal front. The non-dominated solutions obtained by NSGA-II-saJG for ZDT4 function are away from the known Pareto-optimal front, indicating premature convergence to a local Pareto-optimal front (as ZDT4 has 100 distinct Pareto-optimal fronts). However,

global Pareto-optimal front of ZDT4 problem can be found by NSGA-II-saJG using larger number of generations, which means this algorithm is able to escape from the local optimal front after some generations of stagnation. On the other hand, Alt-NSGA-II-aJG gives converged solutions closer to the global Pareto-optimal front in fewer generations.

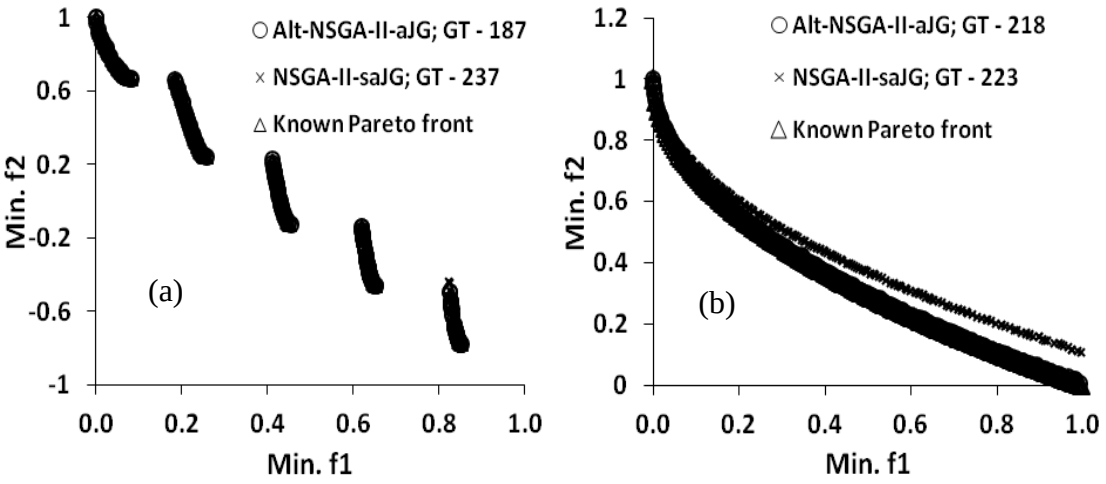


Figure 4.2: Non-dominated solutions obtained by Alt-NSGA-II-aJG and NSGA-II-saJG algorithms using random seed of 0.05: (a) ZDT3 and (b) ZDT4

4.2.4 Performance Comparison on Constrained Test Functions

Table 4.6 shows values of GD^t/GD^t_{max}, SP^t, IGD^t/IGD^t_{max} and GT for constrained test problems by four JG adaptations using the termination criterion. As mentioned earlier, these results are average of 10 runs with different random seed values. Alt-NSGA-II-aJG gives smallest GD^t/GD^t_{max} values for OSY, CONSTR and TNK problems, and close to the smallest value of GD^t/GD^t_{max} for SRN problem. It requires the smallest GT for OSY, CONSTR and SRN problems, and its GT (= 115) for TNK problem is comparable to the best GT value of 105.

Table 4.6: GD^t/GD^t_{max}, SP^t and IGD^t/IGD^t_{max} for constrained test functions obtained by four JG adaptations; these values are average of 10 runs, each with a different random number seed value

	Algorithm	OSY	CONSTR	SRN	TNK	Total
GD^t/GD^t_{max}	NSGA-II-aJG	0.547	0.946	**0.930**	0.556	2.979
	NSGA-II-saJG	1.129	0.974	1.108	0.917	4.128
	NSGA-II-sJG	1.248	1.070	1.014	0.627	3.960
	Alt-NSGA-II-aJG	**0.233**	**0.927**	0.941	**0.168**	**2.269**
SP^t	NSGA-II-aJG	0.921	0.988	**0.466**	1.001	3.375
	NSGA-II-saJG	0.888	1.009	0.500	1.021	3.418
	NSGA-II-sJG	**0.872**	**0.975**	0.484	1.034	**3.366**
	Alt-NSGA-II-aJG	0.938	0.987	0.515	**0.971**	3.410
IGD^t/IGD^t_{max}	NSGA-II-aJG	**0.528**	**0.943**	0.976	**0.545**	**2.991**
	NSGA-II-saJG	0.824	1.061	1.001	0.706	3.593
	NSGA-II-sJG	0.754	0.920	**0.964**	0.595	3.233
	Alt-NSGA-II-aJG	0.583	0.987	0.965	0.998	3.533
GT	NSGA-II-aJG	305	198	317	137	957
	NSGA-II-saJG	223	145	290	120	778
	NSGA-II-sJG	182	161	340	**105**	788
	Alt-NSGA-II-aJG	**173**	**95**	**106**	114	**488**

All JG adaptations of NSGA-II are comparable based on SP^t for different problems tested; variation in total value of SP^t for different JG adaptations is less than 2 percent. NSGA-II-aJG gives smallest values of IGD^t/IGD^t_{max} for OSY, CONSTR and TNK problems, and close to the smallest value of IGD^t/IGD^t_{max} for SRN problem. Alt-NSGA-II-aJG is computationally more efficient than the other adaptations, while

number of generations used by NSGA-II-aJG is significantly more than the best GT required for solving these problems. Total of GD^t/GD^t_{max}, SP^t, IGD^t/IGD^t_{max} and GT by four JG adaptations on constrained functions tested are in the last column of Table 4.6. Overall, Alt-NSGA-II-aJG and NSGA-II-aJG are the best based on GD^t/GD^t_{max} and IGD^t/IGD^t_{max} respectively. The former is computationally efficient too. NSGA-II-saJG performs worse than other adaptations, based on all performance metrics. Figure 4.3 shows the non-dominated solutions obtained by Alt-NSGA-II-aJG and NSGA-II-saJG for OSY and TNK test functions.

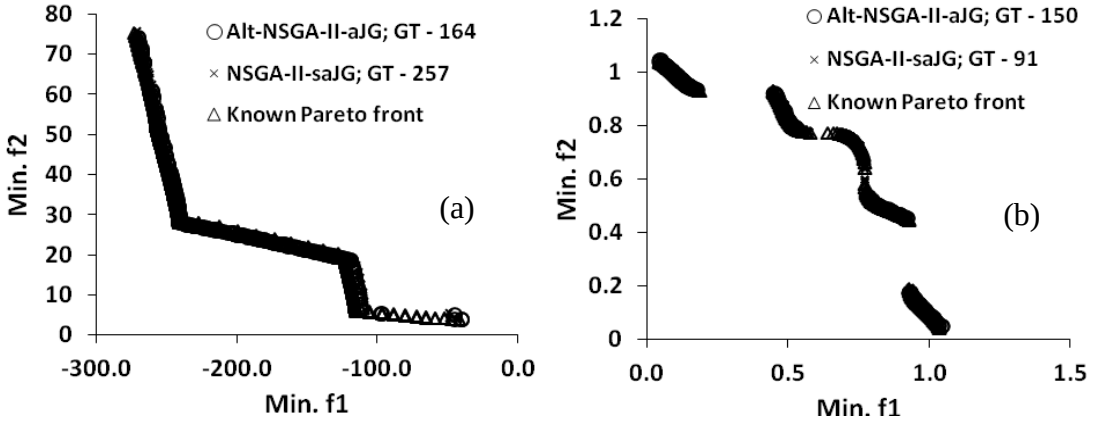


Figure 4.3: Non-dominated solutions obtained by Alt-NSGA-II-aJG and NSGA-II-saJG algorithms using random seed of 0.05: (a) OSY and (b) TNK

The obtained Pareto fronts are visually compared with the known Pareto fronts at different generations, and it is found that NSGA-II-saJG algorithm takes about 225 and 100 generations respectively for OSY and TNK test functions, to reach closer to the known solutions. Alt-NSGA-II-aJG takes about 150 generations for OSY test function, whereas it requires about 300 generations for TNK test function. The search becomes stagnant between 100 to 300 generations, and it can find the middle part of the Pareto front around 300 generation. As solution is not improving for many generations, so search is terminated at 150 generation as per the developed termination criterion.

4.3 Normalized Normal Constraint Method with TC

The non-dominated solutions obtained by stochastic global search can be improved by deterministic search method. The search space, near to the global solution, can be assumed convex and unimodal in nature. In this study, NNC method is used as the local search. Integrated multi-objective differential evolution (I-MODE) algorithm is used as global search, which has an improvement based termination criterion, taboo list to avoid the revisit of search space, and self-adaptation of algorithm parameters. The hybrid approach (I-MODE + NNC) is tested on nine bi-objective test functions.

NNC formulates the multi-objective optimization (MOO) problem as a single objective optimization (SOO) problem based on linear mapping of objectives. In this method for two objectives (see Figure 4.4), each objective is normalized in the range of 0 to 1 using its minimum and maximum values obtained by SOO. F^A and F^B are the anchor points obtained by successively minimizing the first and second objectives respectively (see Figure 4.4), whereas Utopia line (UL) is defined by connecting both anchor points. NNC incorporates one additional inequality constraint in the problem formulation for bi-objective optimization problem (defined in equation 4.1). Now, F^g is one non-dominated solution obtained by stochastic global search (i.e., I-MODE) near the Pareto-optimal front. NNC formulation for this solution can be written as follows:

$$\text{Min.} \quad f_2(\mathbf{x}) \tag{4.4a}$$

$$\text{Subject to} \quad \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U,\ \mathbf{g}(\mathbf{x}) \leq \mathbf{b} \text{ and } UL(F - F^g)^T \leq 0 \tag{4.4b}$$

Here F is the optimum solution obtained for F^g using NNC method. $UL(F - F^g)^T$ is the additional inequality in the problem formulation (with respect to the original problem

in equation 4.1) which divides the objective search space into feasible and infeasible search spaces (see Figure 4.4). NNC method improves the non-dominated solutions in the direction of Pareto-optimal front, whereas ε-constraint method improves the obtained solution for one of the objective function.

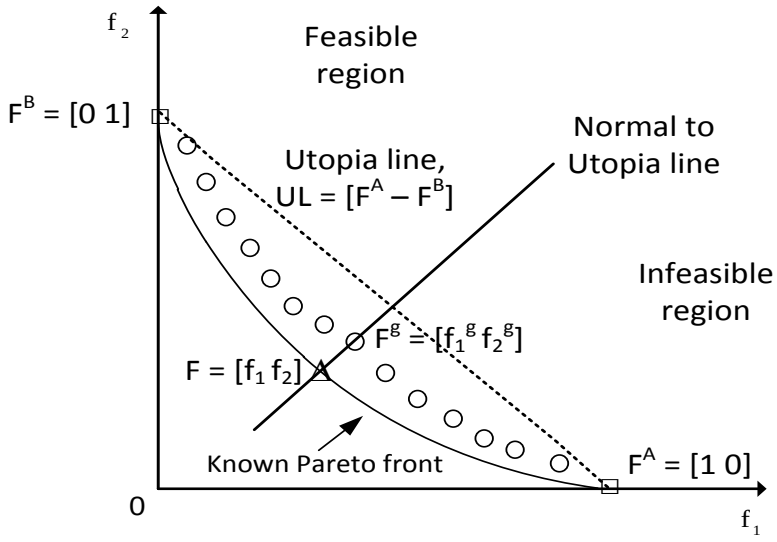


Figure 4.4: Graphical representation of NNC for two objectives

The newly formulated constrained SOO problem (equation 4.4) can be solved using an efficient SOO method. As mentioned earlier, NNC method is unlikely to find the global Pareto-optimal front for non-convex and multi-modal problems. Hence, a stochastic global search (i.e., I-MODE) before NNC method is used to escape from the local Pareto-optimal fronts. Decision variable vector, corresponds to the non-dominated solution obtained by stochastic global search, is used as initial point for single objective deterministic search method (e.g., simplex, generalized reduced gradient, etc.). In the proposed hybrid MOO, stochastic global search is terminated when its progress becomes significantly slow, and each Pareto-optimal solution obtained is refined using NNC. NNC method can be applied on MOO problems with more than two objectives; a general description of NNC method for MOO problems can be found in Messac et al. (2003).

4.3.1 Test Functions and Algorithm Parameters

The hybrid algorithm has been tested on 9 bi-objective test functions; of these, six are unconstrained problems and remaining three are bi-objective constrained problems. These test functions are summarized in Table 4.2 (Deb et al., 2001; Coello Coello et al., 2007). KUR and FON have discontinuous and non-convex Pareto-optimal fronts, respectively. Characteristics of the remaining test problems are discussed in the previous section. Here, ZDT1, ZDT2 and ZDT3 test problems are tested for 10 number of decision variables. Values of algorithm parameters used for these test functions are summarized in Tables 4.7. Cr and F values are self-adapted in I-MODE algorithm, whereas values of taboo list size (TLS), taboo radius (TR), λ, δ_{GD} and δ_{SP} are taken based on recommendation in Chapter 3. For ZDT4 problem, fixed values of Cr (= 0.3), F (= 0.5) and MNG (= 300) are used.

Table 4.7: Values of I-MODE parameters for all the test functions

Parameter	**Value**	**Parameter**	**Value**
Population size, N	100	δ_{GD}	0.0003
Maximum number of generations, MNG	200	δ_{SP}	0.1
Taboo list size, TLS	N/2	λ	10
Taboo radius, TR	0.01		

In this work, generational distance (GD^t) is used to evaluate the performance of hybrid algorithm on the selected test functions. GD^t is used to evaluate the closeness of the obtained Pareto-optimal front to the known Pareto-optimal front (defined in Chapter 3). Objective functions values of known Pareto-optimal fronts for different test functions are taken from http://www.cs.cinvestav.mx/~emoobook.

4.3.2 Performance Evaluation on Test Functions

NNC is employed to quickly improve the non-dominated solutions obtained by the I-MODE algorithm. The I-MODE algorithm handles the inequality constraint using feasibility approach (Deb et al., 2002). As discussed in previous section, NNC formulates the MOO problem as a SOO problem which can be solved using an efficient SOO method. In this work, the newly formulated SOO problem is solved using Excel Solver, which employs generalized reduced gradient (GRG2) method for solving nonlinear optimization problems. Number of iterations used by Solver are not available to the user, but these can be obtained manually. It is not possible to get the number of iterations used by all the solution points of the obtained Pareto-optimal front, in each run for all the test functions. Hence, two extreme points and three equally spaced intermediate points in one run for each test functions are used to estimate the NFEs used by local search. GRG2 uses first partial derivatives of each function (objective function + constraints) with respect to each variable, and these are computed by finite difference approximation (User's Guide for GRG2 Optimization Library, Windward Technologies). Central difference requires two function evaluations per partial derivative while forward difference requires only one. In this study, forward difference approximation has been used. Table 4.8 present mean termination generation (GT) and mean NFE used by I-MODE algorithm (over 30 runs). It can be seen that use of taboo list reduces NFE used (NFE without use of taboo list = GT×N). The I-MODE algorithm has varying exploration capabilities in different runs, which results in reasonably high standard deviation of NFEs. The estimated NFEs, to refine one solution of the obtained Pareto-optimal front are also reported in Table 4.8 for all the test functions.

Table 4.8: GT and NFE used by I-MODE search for successful runs, and estimated average NFE used by NNC to refine one solution obtained using I-MODE

Test function	**μ_{GT} used by I-MODE**	**NFE used by I-MODE**		**Estimated average NFE used by NNC** = DV×Iterations
		Mean	**Std**	
ZDT1	78	7295	734	107
ZDT2	91	7773	902	86
ZDT3	75	7072	858	86
ZDT4	223	18221	2052	200
CONSTR	42	3169	664	6
SRN	35	2170	887	6
TNK	54	3220	1043	11
KUR	51	2992	543	34
FON	55	5335	1290	69

Table 4.9 presents mean SR (success rate) and mean GD^t values obtained using I-MODE and hybrid (I-MODE + NNC) algorithms. SR is the percentage of runs in which hybrid search reached the global optimum. NNC search has been used to refine the solutions obtained by I-MODE algorithm, and it is able to improve the mean GD^t values in all the test functions. NNC is also able to improve the solution obtained for optimization problem with discontinuous Pareto-optimal front (e.g., ZDT3, KUR). In most of the cases, NNC method is able to reach the nearby Pareto-optimal front, but it is not able to escape from the local Pareto-optimal front. In case of ZDT4 (multi-modal) test function, many times I-MODE is not able to find the correct Pareto-optimal front, and NNC refined the non-dominated solutions obtained (by I-MODE) to the near-by local Pareto-optimal front. So, SR are same for both I-MODE and hybrid algorithms.

Table 4.9: μ_{GD}^t and SR using I-MODE and hybrid (I-MODE + NNC) algorithms for different test functions. μ_{GD}^t using MOSADE, NSGA-II-RC, SPEA2 and MOPSO are taken from Wang et al. (2010).

Test function	SR (I-MODE)	I-MODE	I-MODE + NNC	MOSADE	NSGA-II -RC	SPEA2	MOPSO
		μ_{GD}^t					
ZDT1	100	1.50E-03	**3.05E-04**	1.25E-03	1.34E-03	8.61E-03	1.86E-01
ZDT2	100	1.77E-03	**2.53E-04**	9.81E-04	9.81E-04	2.48E-02	5.24E-01
ZDT3	100	1.27E-03	**1.01E-03**	2.16E-03	2.48E-03	9.72E-03	4.34E-01
ZDT4*	70	2.13E-03	**1.08E-03**	1.20E-03	5.16E-02	9.25E-01	--
CONSTR	100	1.46E-03	**1.22E-03**	4.81E-03	5.13E-03	4.82E-03	4.54E-03
SRN*	63	5.75E-02	7.18E-03	**2.00E-03**	3.71E-03	2.11E-03	2.76E-03
TNK	100	1.55E-03	**4.73E-04**	3.74E-03	4.05E-03	3.82E-03	5.09E-03
KUR	100	6.37E-03	**2.60E-03**	2.44E-02	2.90E-03	7.16E-01	3.01E-02
FON	100	1.81E-03	1.25E-03	**1.24E-03**	2.57E-03	1.86E-03	2.14E-02

* GD^t values are based on successful runs, -- cannot converge to the Pareto-optimal front

Recently, Wang et al. (2010) proposed a multi-objective self adaptive differential evolution with elitist archive and crowding entropy-based diversity measure (MOSADE), and compared its performance with the NSGA-II-RC, SPEA2 and MOPSO algorithms on several test functions. For the comparison purpose, Table 4.9 also presents the obtained values of mean GD^t for MOSADE and reported values of GD^t for NSGA-II-RC, SPEA2, and MOPSO in Wang et al. (2010). 25,000 NFEs were used by MOSADE for each test function studied. Performance of I-MODE is comparable to MOSADE (see Table 4.9), and also it takes less number of function evaluations (see Table 4.8). The GD^t values obtained using hybrid (I-MODE + NNC) search, are better than those reported for other algorithms (Wang et al., 2010), for

many of the test functions (except FON and SRN). The hybrid search uses fewer NFEs compared to MOSADE for different test functions using GD^t as performance indicator; this shows that hybrid search is able to reach the known Pareto-optimal front faster in comparison to other existing algorithms.

In Figure 4.5, box plots are used to show the improvement in GD^t values due to local search. Here, GD^t values obtained only in successful runs are considered. NNC not only improves the closeness between the obtained Pareto-optimal front and known Pareto-optimal front, but also reduces the variation in GD^t values obtained in different runs.

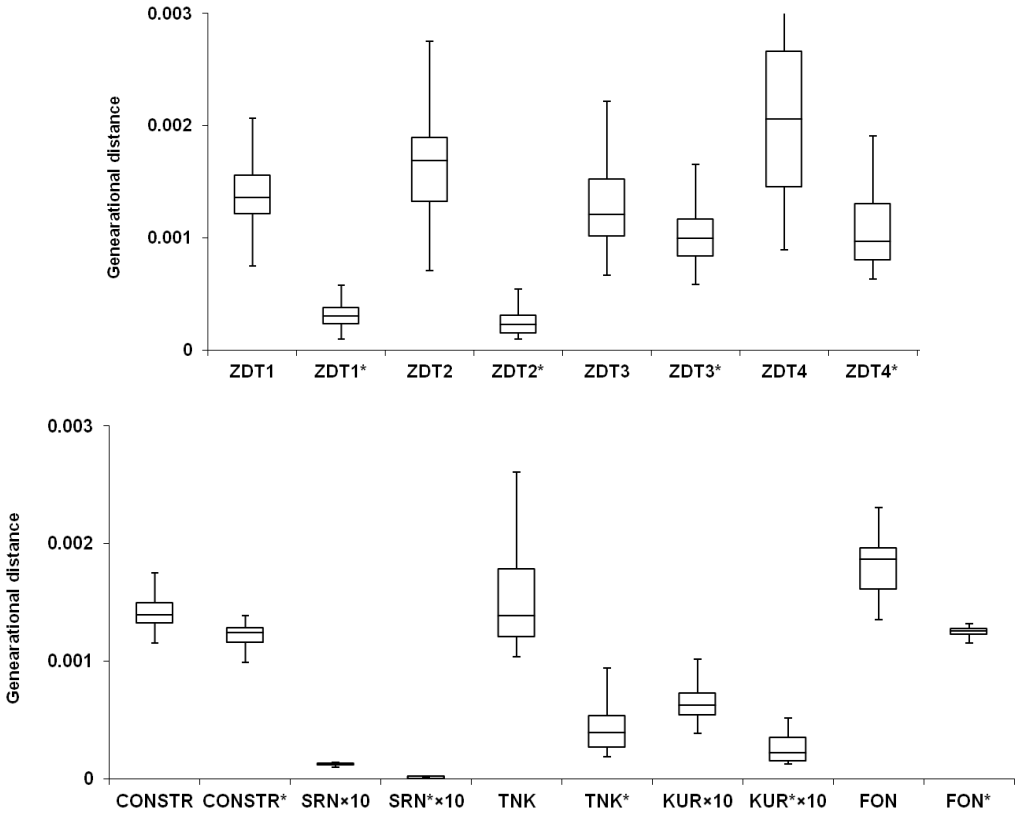


Figure 4.5: Box plot for different test functions using global (I-MODE) and hybrid searches (I-MODE + NNC, which is indicated by *)

Table 4.10 compared the mean GD^t values obtained after termination generation (i.e., GT) and MNG using both I-MODE and hybrid algorithms. It can be seen that mean GD^t using I-MODE algorithm improves between GT and MNG, but this

improvement is relatively small. NNC significantly improves the non-dominated solutions obtained after I-MODE search, after GT and MNG. The difference, between mean GD^t values obtained using hybrid search after GT and MNG, is not much. This small variation is probably due to difference in the distribution of non-dominated solutions along the Pareto-optimal front. Hence, termination criterion can save significant amount of computation, and NNC can precisely reach the known solution.

Table 4.10: μ_{GD}^t using I-MODE and hybrid (I-MODE + NNC) algorithms after GT and MNG

Test function	μ_{GD}^t After GT		μ_{GD}^t After MNG	
	I-MODE	I-MODE + NNC	I-MODE	I-MODE + NNC
ZDT1	1.50E-03	3.05E-04	2.51E-04	2.24E-04
ZDT2	1.77E-03	2.53E-04	1.91E-04	9.73E-05
ZDT3	1.27E-03	1.01E-03	8.55E-04	8.24E-04
ZDT4	2.13E-03	1.08E-03	4.57E-04	4.03E-04
CONSTR	1.46E-03	1.22E-03	1.25E-03	1.23E-03
SRN	5.75E-02	7.18E-03	5.24E-02	7.95E-03
TNK	1.55E-03	4.73E-04	7.08E-04	6.14E-04
KUR	6.37E-03	2.60E-03	2.84E-03	1.36E-03
FON	1.81E-03	1.25E-03	1.29E-03	1.19E-03

The mean GD^t values in Table 4.10 are based on the successful runs. SR of I-MODE on SRN and ZDT4 problems are respectively 63 and 70, after GT . In case of SRN problem, SR has reached to 73 after MNG, but it does not improve for ZDT4 test function. Differential evolution (DE) has inherent characteristic of exploiting the search space at the later stage of the search. Once the search traps in the local

optimum, DE improves the solutions closer to the local optimum. Taboo list is useful in exploring the search space in the early stage, but it only improves the solution distribution at the end of search. Further, SR of I-MODE on ZDT4 test function drops to 60, for zero value of taboo radius.

4.4 Conclusions

In this section, performance of four jumping gene adaptations of NSGA-II is analyzed on bi-objective test problems. This analysis considers quality of non-dominated solutions (i.e., convergence to the known Pareto-optimal front measured by GD^t and IGD^t, and distribution of non-dominated solutions measured by SP^t) and also computational efficiency measured by the number of generations for satisfying the termination criterion (GT). Optimization results confirm that the described termination criterion is able to terminate the search at the right time, and so it can avoid unnecessary computations. Overall, Alt-NSGA-II-aJG is better than the other three JG adaptations for both unconstrained and constrained problems. Since Alt-NSGA-II-aJG is better than NSGA-II-aJG, other operators such as sJG and saJG can be combined with altruism approach in order to improve their performance. Normalized normal constraint method is used to improve the closeness between non-dominated solutions obtained by stochastic and known Pareto-optimal front, in quick time. The termination criterion is able to switch the search from I-MODE to NNC at the right generation.

The proposed hybrid optimization approach is robust and computationally efficient, which makes it suitable for application problems including real time optimization applications. If required, selected number of non-dominated solutions can be refined using local search. For this, Pareto front ranking methods (e.g., net

flow method, Thibault, 2009) can be used to select some individuals for the refinement purpose. As deterministic method is used to refine the non-dominated solutions obtained by global search, gradient calculation/estimation is essential for hybrid search.

Chapter 5

Evaluation of Developed Termination Criterion on Chemical Engineering Application Problems

5.1 Introduction

Real world applications often have multiple performance criteria, and stochastic optimization techniques are used to generate the set of non-dominated solutions. These solutions are not known in advance for new optimization problems, and so decision making on search termination is difficult. In the literature, maximum number of generations (MNG) has been commonly used as the termination criterion in stochastic optimization algorithms. For optimal use of computational resources, termination of stochastic search at the right generation is necessary. An improvement based termination criterion has been developed in Chapter 3. In this, multi-objective differential evolution is combined with termination criterion and taboo list. The resulting algorithm is called I-MODE (integrated multi-objective differential evolution). The developed termination criterion calculates the improvement in the Pareto-optimal front in two consecutive generations, and then statistically checks the improvement in a number of latest generations. This improvement based termination criterion has been tested on mathematical test problems in Chapter 3.

In this chapter, I-MODE algorithm with the termination criterion is evaluated on Chemical Engineering application problems, namely, alkylation, Williams-Otto and fermentation processes. The next section of this chapter describes these processes and their MOO problem formulation. Section 5.3 presents optimization results obtained

for the three application problems. Finally, Section 5.4 summarizes findings of this chapter.

5.2 Chemical Processes used for Testing of Termination Criterion

Testing of the developed termination criterion is performed on the application problems studied in the literature. The known Pareto-optimal front is not used in deciding the search termination, but it is used only to compare the non-dominated solutions obtained after search termination as per the developed termination criterion. In this study, three process applications, namely, alkylation, Williams-Otto and fermentation processes are selected. Alkylation process for single objective has been studied by many researchers; Luus (1978) and Rangaiah (2009b) have studied this process for multiple objectives. Williams-Otto process, a typical chemical process used in numerous studies, was optimized for multiple objectives by Lee et al. (2009). Recently, fermentation process was also studied for multiple objectives (Wang and Lin, 2010).

5.2.1 Alkylation Process

Products of alkylation process are mixed with petroleum refining products to enhance their octane number. A simple schematic of alkylation process is shown in Figure 5.1. In this process, olefins react with isobutene in the presence of acid inside a reactor. Acid present in the reaction mixture settles by gravity, and is removed for regeneration. The other outlet stream from the reactor goes to a fractionator where isobutene is recovered from the product for recycling back to the reactor.

Sauer et al. (1964) formulated single objective optimization problem for alkylation process, which has 10 decision variables, 7 equality constraints and one

objective function (i.e., profit). Recently, Rangaiah (2009b) has optimized alkylation process for multiple objectives using ε-constraint method; two different bi-objective optimization problems for alkylation process are studied, and these are described below.

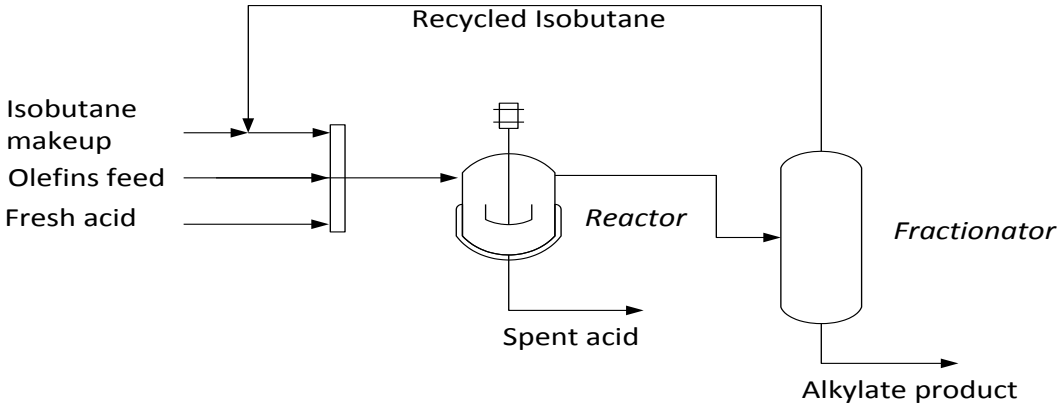


Figure 5.1: A schematic diagram of alkylation process

Objectives	i) Max. profit and max. octane number (x_7)		(5.1a)
	ii) Max. profit and min isobutane recycling (x_2)		(5.1b)
Subject to	Olefin feed, barrels/day	$0 \le x_1 \le 2{,}000$	(5.1c)
	Isobutane recycle, barrels/day	$0 \le x_2 \le 16{,}000$	(5.1d)
	Acid addition rate, 10^3×pounds/day	$0 \le x_3 \le 120$	(5.1e)
	Alkylate production rate, barrels/day	$0 \le x_4 \le 5{,}000$	(5.1f)
	Isobutane feed, barrels/day	$0 \le x_5 \le 2{,}000$	(5.1g)
	Spent acid strength, wt.%	$85 \le x_6 \le 93$	(5.1h)
	Octane number	$90 \le x_7 \le 95$	(5.1i)
	Isobutane to olefins ratio	$3 \le x_8 \le 12$	(5.1j)
	Acid dilution factor	$1.2 \le x_9 \le 4$	(5.1k)
	F-4 performance number	$145 \le x_{10} \le 162$	(5.1l)

Here, profit = $0.063x_4x_7 - 5.04x_1 - 0.035x_2 - 10x_3 - 3.36x_5$ [\$/day] (5.1m)

$$x_2 = x_1x_8 - x_5 \quad (5.1n)$$

$$x_3 = 0.001x_4x_6x_9/(98- x_6) \quad (5.1o)$$

$$x_4 = x_1(1.12 + 0.13167x_8 - 0.0066667x_8^2) \quad (5.1p)$$

$$x_5 = 1.22x_4 - x_1 \quad (5.1q)$$

$$x_6 = 89 + (x_7 - 86.35 - 1.098x_8 + 0.038x_8^2)/0.325 \quad (5.1r)$$

$$x_9 = 35.82 - 0.222x_{10} \quad (5.1s)$$

$$x_{10} = -133 + 3x_7 \quad (5.1t)$$

The two objectives in equation 5.1a are used in one bi-objective problem, and the two objectives in equation 5.1b are for another bi-objective problem.

5.2.2 Williams-Otto Process

Figure 5.2 shows a schematic of Williams-Otto process, which has both reaction and separation sections. Reactants A and B enter the reactor, and produce intermediate product C. Reactant A reacts with intermediate C to produce the main product P and byproduct E. Further, main product also reacts with the intermediate product C, and produces waste product G. The reaction mixture is cooled in a cooler, and then waste product G is removed from the cooled reaction mixture using decanter. After this, the remaining mixture enters a distillation column, where product P is recovered as distillate. Additionally, the bottom product is recycled back to the reactor, and a small amount of bottom product is purged to remove the byproduct E from the process.

Lee et al. (2009) have presented and solved two bi-objective optimization problems for Williams-Otto process, which are presented below.

Objectives i) Max. NPW and max. PBT (profit before tax) (5.2a)

ii) Max. NPW and min PBP (payback period) (5.2b)

Subject to	Volume of reactor, m^3	$0.85 \leq V \leq 10$	(5.2c)
	Temperature of reactor, K	$322 \leq T \leq 378$	(5.2d)
	Purge fraction	$0 \leq \eta \leq 0.99$	(5.2e)
	Mass flow rate of stream B, kg/h	$10{,}000 \leq q_2^B \leq 15{,}000$	(5.2f)
	Six process model equations (5.2h-m)		(5.2g)

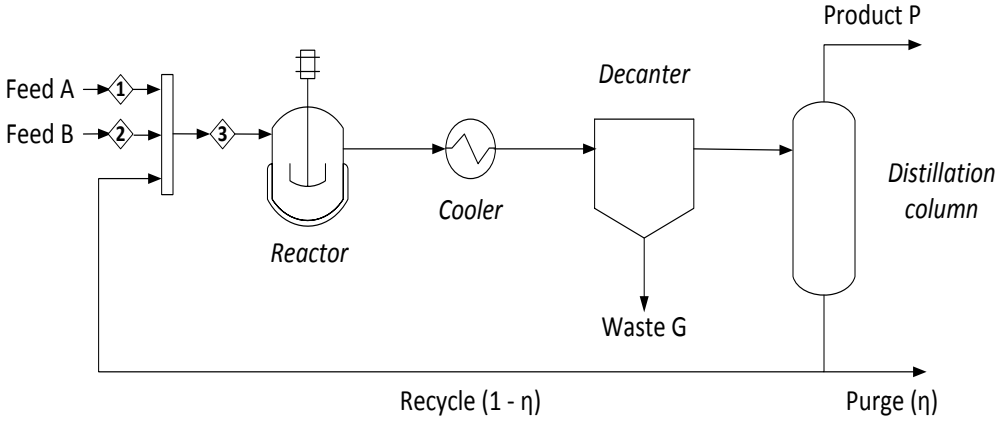


Figure 5.2: A schematic diagram of William-Otto process

Lee et al. (2009) provided details on the calculation of objective functions. Williams-Otto process has 6 non-linear equations and 10 variables; so there are four degrees of freedom or decision variables (i.e., V, T, η and q_2^B) for optimization. The 6 model equations are:

$$q_1^A + q_3^A(1-\eta) - \frac{k_1 q_3^A q_3^B V\rho}{(q_3)^2} - q_3^A = 0 \qquad (5.2h)$$

$$q_2^B + q_3^B(1-\eta) - \frac{(k_1 q_3^A + k_2 q_3^C) q_3^B V\rho}{(q_3)^2} - q_3^B = 0 \qquad (5.2i)$$

$$q_3^C(1-\eta) + \frac{[2k_1 q_3^A q_3^B - 2k_2 q_3^B q_3^C - k_3 q_3^C(2160 + 0.1q_3^E)]V\rho}{(q_3)^2} - q_3^C = 0 \qquad (5.2j)$$

$$q_3^E(1-\eta) + \frac{2k_2 q_3^B q_3^C V\rho}{(q_3)^2} - q_3^E = 0 \qquad (5.2k)$$

$$\frac{q_3^E(1-\eta)}{10} + \frac{[k_2 q_3^B - 0.5k_3(2160 + 0.1q_3^E)]q_3^C V\rho}{(q_3)^2} - (2160 + 0.1q_3^E) = 0 \qquad (5.2l)$$

$$\frac{[1.5k_3(2160 + 0.1q_3^E)]q_3^C V\rho}{(q_3)^2} - q_3^G = 0 \qquad (5.2m)$$

Here, $q_3 = q_3^A + q_3^B + q_3^C + 1.1q_3^E + q_3^G + 2160$, q_i^j is the mass flow rate of j^{th} component in i^{th} stream (see Figure 5.2), ρ is the density of reaction mixture, and k_1, k_2 and k_3 are rate constants; expression of different rate constants can be obtained from Lee et al. (2009). The above 6 model equations (5.2h-m) are solved for q_1^A, q_3^A, q_3^B, q_3^C, q_3^E and q_3^G using Solver in Excel.

5.2.3 Three-stage Fermentation Process Integrated with Cell Recycling

Wang and Lin (2010) have studied a three-stage continuous fermentation process integrated with cell recycling, where each stage has a fermentor and a cell separator to separate the cell mass and recycle it back to the fermentor. Schematic diagram of this fermentation process is shown in Figure 5.3. Equations 5.3a, 5.3b and 5.3c are the steady state material balances for cell mass, glucose and ethanol respectively, around the k^{th} stage of continuous fermentation process. The kinetic model is given by equations 5.3d and 5.3e.

$$D\,[b_{k-1}x_{k-1} - b_k x_k] + \mu_k x_k = 0 \qquad (5.3a)$$

$$D\left[s_{f,k} + s_{k-1} - s_k\right] - \frac{q_{p,k}}{Y_{p/s}} x_k = 0 \qquad (5.3b)$$

$$D[p_{k-1} - p_k] + q_{p,k}x_k = 0 \qquad (5.3c)$$

$$\mu_k = \left(\frac{\mu_m s_k}{K_s + s_k + s_k^2/K_{sI}}\right)\left(\frac{K_p}{K_p + p_k + p_k^2/K_{pI}}\right) \qquad (5.3d)$$

$$q_{p,k} = \left(\frac{\nu_m s_k}{K_s' + s_k + s_k^2/K_{sI}'}\right)\left(\frac{K_p'}{K_p' + p_k + p_k^2/K_{pI}'}\right) \qquad (5.3e)$$

Here, D (= F_1/V) is the dilution rate, F_1 is the feed flow rate to the first stage, and V is the volume of each fermentor. x_k, s_k and p_k are respectively cell mass, glucose and ethanol concentrations in the k^{th} stage. b_k is the bleed ratio for k^{th} stage. $s_{f,k}$ is glucose concentration in the feed to k^{th} stage; since feed is entering only into the first stage, $s_{f,2}$ = 0 and $s_{f,3}$ = 0. Further, for the first stage, b_{k-1}, x_{k-1}, s_{k-1} and p_{k-1} are all zero. Kinetic parameter values used in equations 5.3a-e are listed in Table 5.1.

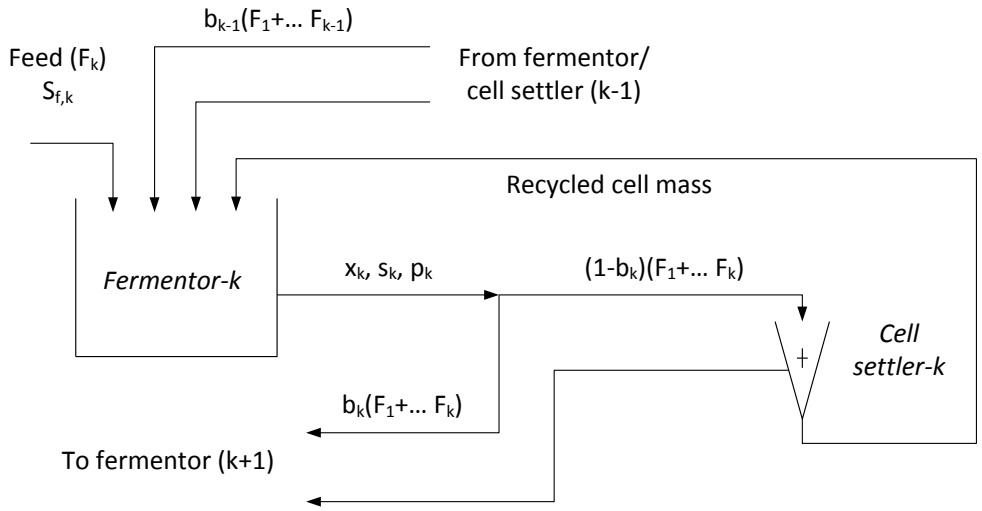


Figure 5.3: A schematic diagram of k^{th} stage of continuous fermentation process integrated with cell recycling (and glucose as feed)

For optimizing the fermentation process, ethanol productivity and glucose conversion are used as two objectives, which ensure efficient utilization of production capacity and glucose respectively. The MOO problem for the 3-stage continuous fermentation process integrated with cell recycling is summarized in Table 5.2. Decision variables for this optimization are dilution rate (D), glucose concentration in feed ($s_{f,1}$) and cell mass recycling for different stages (i.e., bleed ratios, b_1, b_2 and b_3). Physical constraints are positive values of productivity and glucose conversion for each stage. Residual glucose after the third stage and total glucose supplied per unit volume of all fermentors in the feed are additional constraints in the optimization problem (Wang and Lin, 2010).

Table 5.1: Kinetic parameters and their values for the continuous fermentation process integrated with cell recycling (Wang and Sheu, 2000)

Kinetic parameter	Estimated value	Kinetic parameter	Estimated value
μ_m	0.9819	K_p	27.9036
ν_m	2.3507	K'_p	252.306
K_s	2.3349	K_{pI}	41.2979
K'_s	7.3097	K'_{pI}	15.2430
K_{sI}	213.5899	$Y_{p/s}$	0.4721
K'_{sI}	5759.105		

Table 5.2: MOO problem formulation for the three-stage continuous fermentation process integrated with cell recycling; k = 1, 2 and 3

Objective functions	Max. ethanol productivity, [kg/(m^3.h)]	$\frac{D}{3} p_3$
	Max. overall glucose conversion	$1 - \frac{s_3}{s_{f,1}}$
Decision variables	Dilution rate, [1/h]	$3.5 \leq D \leq 4$
	Glucose concentration in feed, [kg/m^3]	$60 \leq s_{f,1} \leq 65$
	Bleed ratio for each stage	$0.1 \leq b_1, b_2, b_3 \leq 0.2$
Constraints	Productivity for each stage, [kg/(m^3.h)]	$D[p_k - p_{k-1}] \geq 0$
	Glucose conversion for each stage	$1 - \frac{s_k}{s_{f_k + s_{k-1}}} \geq 0$
	Residual glucose after 3rd stage, [kg/m^3]	$0.1 \leq s_3 \leq 0.5$
	Total glucose supplied per unit volume of all fermentors, kg/(m^3.h)]	$10 \leq s_T = \frac{D s_{f,1}}{3} \leq 180$
	Model for the process (Equations 5.3a-e for each stage)	

5.3 Optimization Results

In this study, I-MODE algorithm is used to obtain the trade-off solutions for the above three application problems. I-MODE algorithm handles inequality constraints using feasibility approach (Deb et al., 2002). Both Williams-Otto and fermentation processes have equality constraint arising from mass balance equations. Here, sequential solution of optimization problem and equality constraints is performed. For this, optimization algorithm provides a vector of decision variables, and then process model equations (i.e., equality constraints) are solved to obtain values of state variables. Both decision and state variables are used to calculate objectives and constraints. In this work, Solver tool in Excel is used for solving model equations by minimizing sum of squares. Figure 5.4 presents a simple flowchart of optimization method with sequential solution of equality constraints.

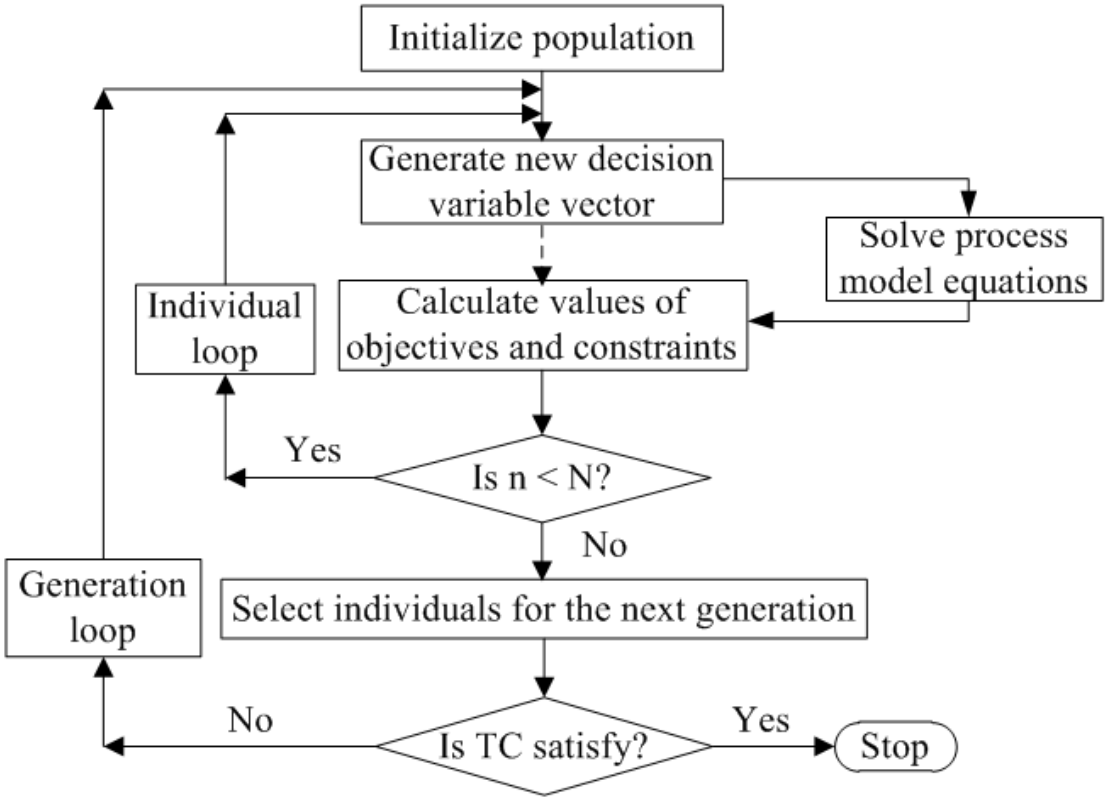


Figure 5.4: A simple flowchart for optimization algorithm with sequential solution of process model

In the testing of termination criterion, values of termination parameters and taboo radius are taken based on the recommendation in Chapter 3; these values are reported in Table 5.3. Initially, all the problems are solved for maximum number of generation (MNG) to obtain the known Pareto front; later, it is used for comparison purpose. In order to ensure global optimality of the known Pareto fronts for different problems, these are visually compared with the reported Pareto fronts. All application problems are solved using I-MODE algorithm, and the non-dominated solutions obtained after termination generations (i.e., GT) as per the termination criterion, are compared against the known Pareto-optimal fronts.

Table 5.3: Algorithm parameters used for different application problems

Application	**To obtain known Pareto**		**I-MODE**	**Termination parameters**		
	MNG	**N**	**N**	δ_{GD}	δ_{SP}	**TR**
Alkylation	500	100	100	0.0003	0.1	0.01
Williams-Otto	200	100	100	0.0003	0.1	0.01
Fermentation	200	100	100	0.0003	0.1	0.01

5.3.1 Alkylation Process

Figure 5.5(a) shows the non-dominated solutions obtained for simultaneous maximization of profit and minimization of isobutane recycle. In this case, the known Pareto front is obtained after running the I-MODE algorithm for 500 generations. It can be seen that non-dominated solutions obtained, after search termination based on the termination criterion, are closer to the global solution (i.e., known Pareto front) and also well distributed along the Pareto-optimal front. Here, generation of termination is 158, which is significantly less than MNG (= 500). Hence, termination criterion is able to terminate the search at the appropriate generation, and it can avoid

running the search for unnecessary generations where only small improvement in solution is possible. The non-dominated solutions obtained for maximization of both octane number and profit, are shown in Figure 5.5(b). A visual comparison between the non-dominated solutions obtained and known Pareto front indicates that termination criterion is successful. Similar to the previous case, the improvement based termination criterion is able to avoid the unnecessary computations. In both the cases, the non-dominated solutions obtained are close enough to the optimal solutions for their industrial acceptance (Figure 5.5).

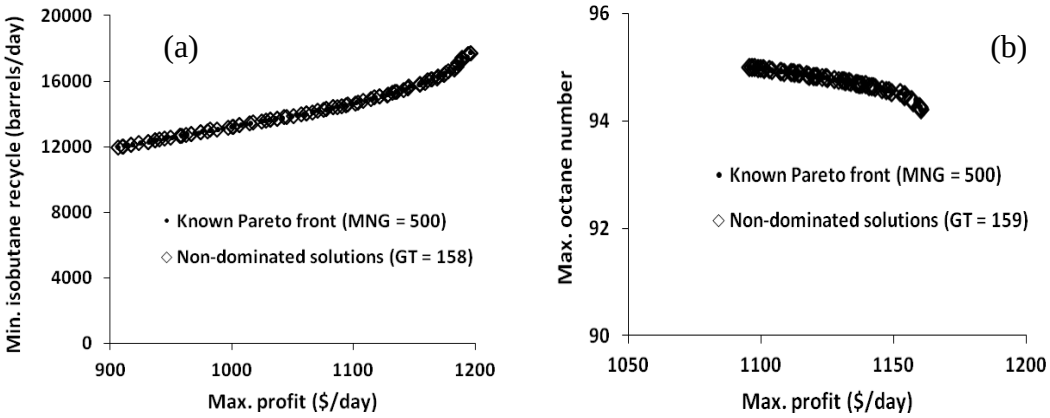


Figure 5.5: Non-dominated solutions obtained for alkylation process: (a) max. profit and min. recycle isobutane, and (b) simultaneous max. both profit and octane number

5.3.2 Williams-Otto Process

Figure 5.6(a) shows the non-dominated solutions obtained for simultaneous maximization of NPW and PBT, whereas non-dominated solutions obtained for maximization of NPW and minimization of PBP are shown in Figure 5.6(b). Initially, both these optimization problems are solved for 200 generations to obtain the known Pareto-optimal fronts. The known Pareto fronts (obtained after MNG) are compared with the reported Pareto fronts in the literature (not shown here), and these are found to be acceptable. Finally, the known Pareto-optimal fronts are used for comparison purpose.

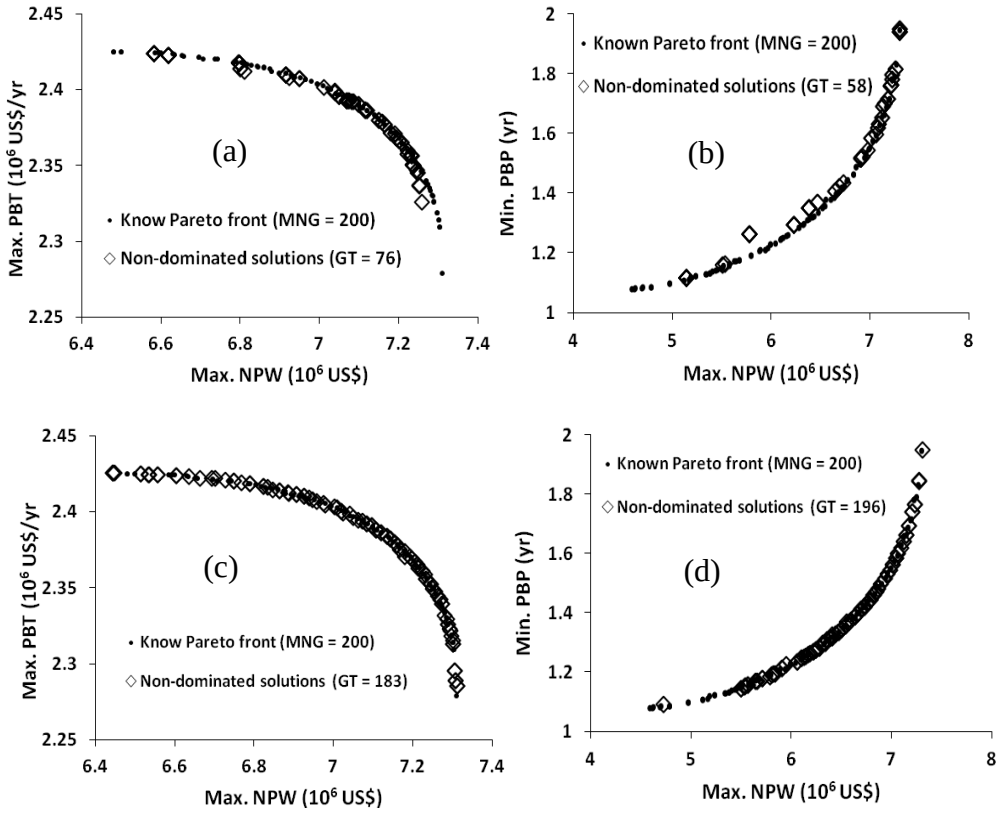


Figure 5.6: Non-dominated solutions obtained for Williams-Otto process: (a), (c) simultaneous maximization of NPW and PBT, and (b), (d) maximization of NPW and minimization of PBP

Although, in both the cases, the non-dominated solutions have converged near to the known Pareto fronts, distribution of the non-dominated solutions along the Pareto-optimal fronts is not good (Figures 5.6a-b). The non-dominated solutions obtained may be acceptable, or these can be considered for further refinement. In this work, the used termination parameters are based on the recommendation for two- and three-objective optimization problems. Hence, smaller values of both termination parameters (δ_{GD} = 0.0002 and δ_{SP} = 0.05) are also tried, and the non-dominated solutions obtained are presented in Figures 5.6(c) and 5.6(d). Now, the non-dominated solutions obtained are covering most of the parts of the known Pareto-optimal fronts, but the required number of generations has increased significantly (with smaller values of termination parameters).

5.3.3 Three-stage Fermentation Process Integrated with Cell Recycling

In the case of three-stage fermentation process integrated with cell recycling, ethanol productivity and xylose conversion are maximized simultaneously. Figure 5.7 shows non-dominated solutions obtained for this fermentation process. Here, the termination criterion is successful in stopping the search at the right generation. The non-dominated solutions obtained are close to the known solutions, and are also well distributed along the Pareto-optimal front. Here, the termination criterion saved lot of computation time (GT = 50). Thus, it avoids the unnecessary computations with little compromise on the solution quality.

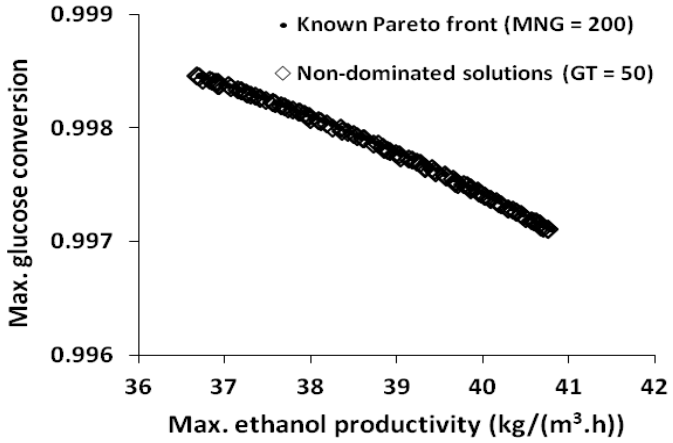


Figure 5.7: Non-dominated solutions obtained for simultaneous maximization of both ethanol productivity and xylose conversion

5.4 Conclusions

In this chapter, I-MODE algorithm is used to optimize three Chemical Engineering applications. The improvement based termination criterion is able to terminate the search at the right generations, for the selected applications. The obtained Pareto-optimal fronts after termination generations are closer to their known solutions. Hence, termination criterion is able avoid unnecessary computations with no or small compromise on the quality of obtained solutions.

Chapter 6

Improved Constraint Handling Technique for Multi-objective Optimization Problems[4]

6.1 Introduction

Application problems often have constraints besides bounds on decision variables; these constraints arise from design equations (such as mass and energy balances), equipment limitations (such as size) and operation requirements (such as temperature limit for safe operation). Mathematical form of a constrained MOO optimization problem is as follows:

$$\text{Min.} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots f_M(\mathbf{x})\} \tag{6.1a}$$

$$\text{Subject to} \quad \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U, \mathbf{h}(\mathbf{x}) = \mathbf{0} \text{ and } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \tag{6.1b}$$

Here f_1, f_2, …, f_M are M number of objective functions; $\mathbf{x}$ is the vector of n decision variables; $\mathbf{x}^L$ and $\mathbf{x}^U$ are respectively vectors of lower and upper bounds on decision variables; and $\mathbf{h}$ and $\mathbf{g}$ are the set of n_e equality and n_i inequality constraints respectively.

Many algorithms have been proposed to solve MOO problems; examples of these algorithms are the elitist non-dominated sorting genetic algorithm (NSGA-II; Deb et al., 2002), strength Pareto evolutionary algorithm (SPEA2; Zitzler et al., 2001), multi-objective particle swarm optimization (MOPSO; Coello Coello and Salazar Lechuga, 2002) and multi-objective differential evolution (MODE). Originally, MOO

[4] This chapter is based on the book chapter: Sharma, S. and Rangaiah, G. P. (2013), Improved constraints handling technique for multi-objective optimization with application to two fermentation processes, Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.

algorithms were developed and studied for solving unconstrained optimization problems (i.e., with bounds on decision variables but without any inequality or equality constraints). Later, to solve constrained MOO problems, several constraint handling techniques were developed and incorporated in the MOO algorithms.

The next section of this chapter briefly reviews different constraint handling approaches, followed by their applications in Chemical Engineering in Section 6.3. Section 6.4 describes ACRFA for constrained SOO problems, and Section 6.5 presents modified ACRFA for constrained MOO problems. Section 6.6 describes MODE with ACRFA. In Section 6.7, performance of ACRFA is compared with the classical feasibility approach on two test functions. MODE with modified ACRFA is used for MOO of two fermentation processes in Section 6.8. Finally, concluding remarks are made at the end of this chapter.

6.2 Constraint Handling Approaches

Coello Coello (2002) summarized constraint handling methods utilized in evolutionary algorithms under five main categories: (i) penalty function approach, (ii) separation of constraints and objectives, (iii) special representation, (iv) repair algorithms, and (v) hybrid methods. Penalty function approach penalizes objective functions (e.g., it increases their values by adding penalty terms, in case of minimization of objectives) based on the extent of constraint violation; it is simple in concept and has been popular. However, the difficulty in using this approach is the selection of a suitable penalty factor value for different problems. If the penalty factor value is not appropriate, then the optimization algorithm may converge to either non-optimal feasible solution or infeasible solution. Penalty function approach is divided into several sub-categories (e.g., static, dynamic, adaptive, co-evolutionary, etc.)

based on the method of penalty factor handling. If the objective function value cannot be computed in the infeasible search space for some reason, then the penalty function approach cannot be used for solving such constrained optimization problems. For example, mathematical functions such as logarithms and/or square roots, present in the objectives, cannot be evaluated for negative values of their arguments. If values of objective functions cannot be calculated or process simulator does not converge for a particular set of decision variable values (i.e., potential solution), then worst value for each objective can be given. Finally, this solution is very unlikely to be selected for the subsequent generation.

Deb et al. (2002) proposed feasibility approach for handling inequality constraints, which considers the constraints and objectives separately. It selects a feasible solution over an infeasible solution during the selection step in the generations, so this limits the diversity of search. Constraint handling using special representation is used for particular types of optimization problems, whereas repair algorithms convert the infeasible individual into a feasible or less infeasible individual (Harada et al., 2007). Finally, in the hybrid approach, constraint handling is tied with some other optimization approach. For example, Van Le (1995) combined fuzzy logic with evolutionary programming to handle the constraints; here, constraints are replaced by fuzzy constraints which allow high tolerance for constraint violation. Of the five categories of constraint handling methods, penalty function and feasibility approaches have been popular for solving constrained MOO problems in Chemical Engineering application; see section 6.3 for more details.

Generally, feasibility approach is good for solving problems with inequality constraints due to their large feasible regions. Feasible search space is extremely small for equality constrained problems. The feasibility approach can handle equality

constraints via suitable transformation into inequality constraints, but this requires different values of tolerance limit for different constraints in the same problem and also for different problems. Takahama and Sakai (2006) proposed ε-constrained DE, where equality constraints are relaxed systematically. Zhang and Rangaiah (2012) proposed adaptive constraint relaxation with feasibility approach (ACRFA) for handling constraints in single objective optimization (SOO) problem. In this approach, individuals with total constraint violation less than certain limit are temporarily considered as feasible individuals during selection for the next generation. This violation limit is changed dynamically based on the performance of search. In this chapter, ACRFA proposed by Zhang and Rangaiah (2012), is modified for solving constrained MOO problems. It is implemented in the multi-objective differential evolution (MODE) algorithm and tested on two benchmark functions with equality and inequality constraints. Then, MODE with ACRFA is used to optimize two fermentation processes for two objectives; these applications involve many equality constraints arising from mass balances. The performance of ACRFA is compared with the feasibility approach alone, and discussed.

6.3 Constraint Handling Approaches in Chemical Engineering

Researchers have used different approaches for handling constraints in optimization problems. Selected studies in the past decade in Chemical Engineering involving constrained MOO problems and using stochastic algorithms with constraint handling approaches are briefly reviewed in this section.

Li et al. (2003) optimized the design of a styrene reactor, where penalty function approach is used for handling constraints; they used a larger value for penalty factor to locate the global optimum precisely. Yee et al. (2003) also used NSGA with

penalty function approach to optimize the styrene reactor. Mitra et al. (2004) handled constraints using feasibility approach to optimize a semi-batch epoxy polymerization process. In this study, feasibility approach is chosen for handling constraints as it does not involve any additional parameter. Tarafder et al. (2005) used NSGA-II with feasibility approach to optimize styrene manufacturing process for multiple objectives, and they found feasibility approach to be efficient and better than penalty function approach. Guria et al. (2005b) have used penalty function approach for handling constraints in the optimization of reverse osmosis process for multiple objectives. Sarkar and Modak (2005) used NSGA-II with feasibility approach for MOO of fed-batch bioreactors.

Agrawal et al. (2006) applied NSGA-II and its jumping gene adaptations with penalty function approach for optimal design of a low density polyethylene tubular reactor for multiple objectives. Later, Agrawal et al. (2007) used both penalty and feasibility approaches to handle constraints in the optimization of the same process, and found that feasibility approach performs slightly better than penalty function approach. Sand et al. (2008) have used penalty function approach for handling constraints in batch scheduling; penalty function approach is selected over repair algorithm as later approach may introduce bias into search. Ponsich et al. (2008) tried several constraint handling techniques with a genetic algorithm to optimize the design of a batch plant as an example; these include elimination of infeasible individuals (i.e., fitness of infeasible individual = 0, which prevents selection of an infeasible individual using roulette wheel), use of penalty term in the objective, relaxation of upper bounds for discrete variables, dominance-based tournament (similar to feasibility approach), and multi-objective strategy. Based on their results, Ponsich et al. (2008) concluded that elimination of infeasible individuals is most attractive when

objective function calculations require less computational effort, and dominance-based tournament is better if the process model calculations require large computational time. This is mainly due to the number of (objective) function evaluations required.

Mazumder et al. (2010) have used NSGA-II-aJG with penalty function approach to optimize design of a liquid-solid circulating bed for continuous protein recovery, for multiple objectives. Kundu et al. (2011) also have used penalty function approach to handle inequality constraints in the MOO optimization of a counter-current moving bed chromatographic reactor. From this brief review of the selected studies, it is clear that both penalty function and feasibility approaches have been used and popular for handling constraints in MOO of Chemical Engineering applications. Of these two, feasibility approach seems to be preferable because it does not involve any parameter and for potential computational efficiency.

6.4 Adaptive Constraint Relaxation and Feasibility Approach (ACRFA) for SOO

Real world optimization problems often involve both equality and inequality constraints. Although an equality constraint can be converted into an inequality constraint by a priori relaxation, feasible search space is very small in case of problems with equality constraints, compared to complete search space and also compared to feasible search space of problems with no equality constraints. Moreover, equality constraints in Chemical Engineering problems arise from mass balances, mole fraction summation and/or energy balances, with terms having a wide range of magnitudes. Such equality constraints require different magnitudes of relaxation for obtaining meaningful optimal solutions.

Zhang and Rangaiah (2012) introduced the concept of adaptive relaxation of constraints based on the number of feasible points obtained in each generation. First, values of objective functions and constraints are calculated for the initial population. Next, total absolute constraint violations (TACV) is calculated for each individual in the population, using:

$$\text{TACV} = \sum_{i=1}^{n_e} |h_i(\mathbf{x})| + \sum_{j}^{n_i} \max [0,\ g_j(\mathbf{x})] \qquad (6.2)$$

where h_i and g_j are the equality and inequality constraints respectively, and n_e and n_i are the number of equality and inequality constraints respectively. Median of TACV for all individuals in the initial population is chosen as the initial value for constraint relaxation (μ). Individuals are treated as temporarily feasible if their TACV is less than μ.

In the first generation, feasibility of each individual is decided using μ value from the initial population; i.e., the individual is considered feasible if its TACV is less than μ. After that, feasibility approach of Deb et al. (2002) is used to select the individuals for the subsequent generation. μ value is updated based on the number of feasible solutions obtained at the end of the first generation (see equation 6.3), which is used to decide the feasibility of individuals in the next generation.

$$\mu_{G+1} = \mu_G \left(1 - \frac{F_F}{N}\right) \qquad (6.3)$$

Here, F_F is the fraction of feasible individuals at the end of first generation. G and N are respectively the generation number and population size. The iterative procedure is repeated until the maximum number of generations.

6.5 ACRFA for MOO

In the case of SOO by differential evolution (DE), selection is made between target and trial individuals. In MOO by MODE, on the other hand, non-dominant sorting is employed where all target and trial individuals collectively contest for selection to the next generation. A trial individual can be temporarily feasible based on its TACV and μ, but, based on non-dominated sorting, it may not be selected for the subsequent generation. In any case, F_F can be obtained by checking the feasibility of individuals selected for subsequent generation. In the initial tests, μ value was updated using equation 6.3 in MODE, but μ was found to decrease very fast leading to many infeasible individuals in the population. In the case of SOO, a few feasible individuals are good enough to obtain the global solution. On the other hand, for MOO, larger number of feasible solutions is required to obtain the Pareto-optimal front with many non-dominated solutions. Hence, several other relaxation schemes (similar to equation 6.3) were tried but they all showed fast decrease in μ value.

Finally, a different strategy is adopted for dynamically updating μ value in ACRFA for MOO problems with constraints. μ value is chosen so as to make a certain percentage of individuals selected for the next generation as infeasible. After trying μ based on 10%, 25% and 50% infeasible individuals on several test problems, μ value corresponding to 25% infeasible individuals is found to be better. Since better individuals are selected for the next generation, μ value is expected to decrease continually; this is confirmed by results presented later.

6.6 Multi-objective Differential Evolution with ACRFA

MODE algorithm of Kukkonen and Lampinen (GDE3, 2007) is used for implementing and testing ACRFA (MODE-ACRFA) for solving constrained MOO

problems. Flowchart of MODE-ACRFA is shown in Figure 6.1. Population of N individuals is initialized randomly inside the bounds on decision variables. Values of objectives, constraints and TACV (according to equation 6.2) are calculated for each individual in the initial population. Then, initial value of μ is selected such that 25% of individuals in the initial population will be temporarily infeasible based on TACV.

In each generation, a trial individual/vector for each target individual in the current/initial population is generated by mutation and crossover. For this, DE/rand/1 mutation strategy and binomial crossover are applied. See Chapter 3 for more details on these mutation and crossover operations in classic differential evolution (Price et al., 2005). After crossover, the trial vector is tested for satisfaction of decision variable bounds; if a bound on any decision variable is violated, then it is randomly re-initialized within the bounds on that decision variable. Finally, values of objective functions, constraints and TACV of the trial individual are calculated. Thus, N trial individuals are generated and stored in the child population, which is later mixed with the parent population containing target individuals.

The combined population of 2N individuals undergoes non-dominated sorting followed by crowding distance calculation. If the MOO problem has no constraints, then N individuals are selected from the combined population based on the following definitions and steps.

a) Two individuals A and B are non-dominated to each other if A is better than B in at least one objective, and also B is better than A in at least one other objective. Thus, both these individuals are equally good. One individual is dominating another individual if it is better than the other in all objectives.

b) The number of individuals dominating each individual (n_d) is calculated. First rank is assigned to the non-dominated individuals with $n_d = 0$. This is shown as PF1 in Figure 6.2.

c) Then, non-dominated individuals in the remainder of the combined population (i.e., excluding those with first rank) are assigned second rank (shown as PF2 in Figure 6.2). This procedure is repeated until all individuals are ranked.

d) The first/best N individuals are selected as the population for the subsequent generation. For this, individuals are first selected based on the Pareto rank given in the above steps. When all the individuals of a Pareto front cannot be selected for the subsequent generation (e.g., PF3 in Figure 6.2), less crowded individuals (based on the crowding distance measure) are selected to complete the population size. Note that the crowding distance measures distribution of non-dominated solutions on the Pareto-optimal front by calculating Euclidean distance between two neighboring non-dominated solutions; see Deb (2001) for more details.

For constrained MOO problems, feasibility of all individuals in the combined population is decided using the current μ value. MODE-ACRFA algorithm selects N individuals for the subsequent generation from the combined population according to steps b-d above, but the following definition of constrained dominance is used in step a (according to feasibility approach of Deb et al., 2002). If any of the following conditions is true, then individual A is dominating individual B.

i) Both the individuals are feasible, and individual A dominates B (as per the usual dominance definition in step a above).

ii) Individual A is feasible and B is infeasible.

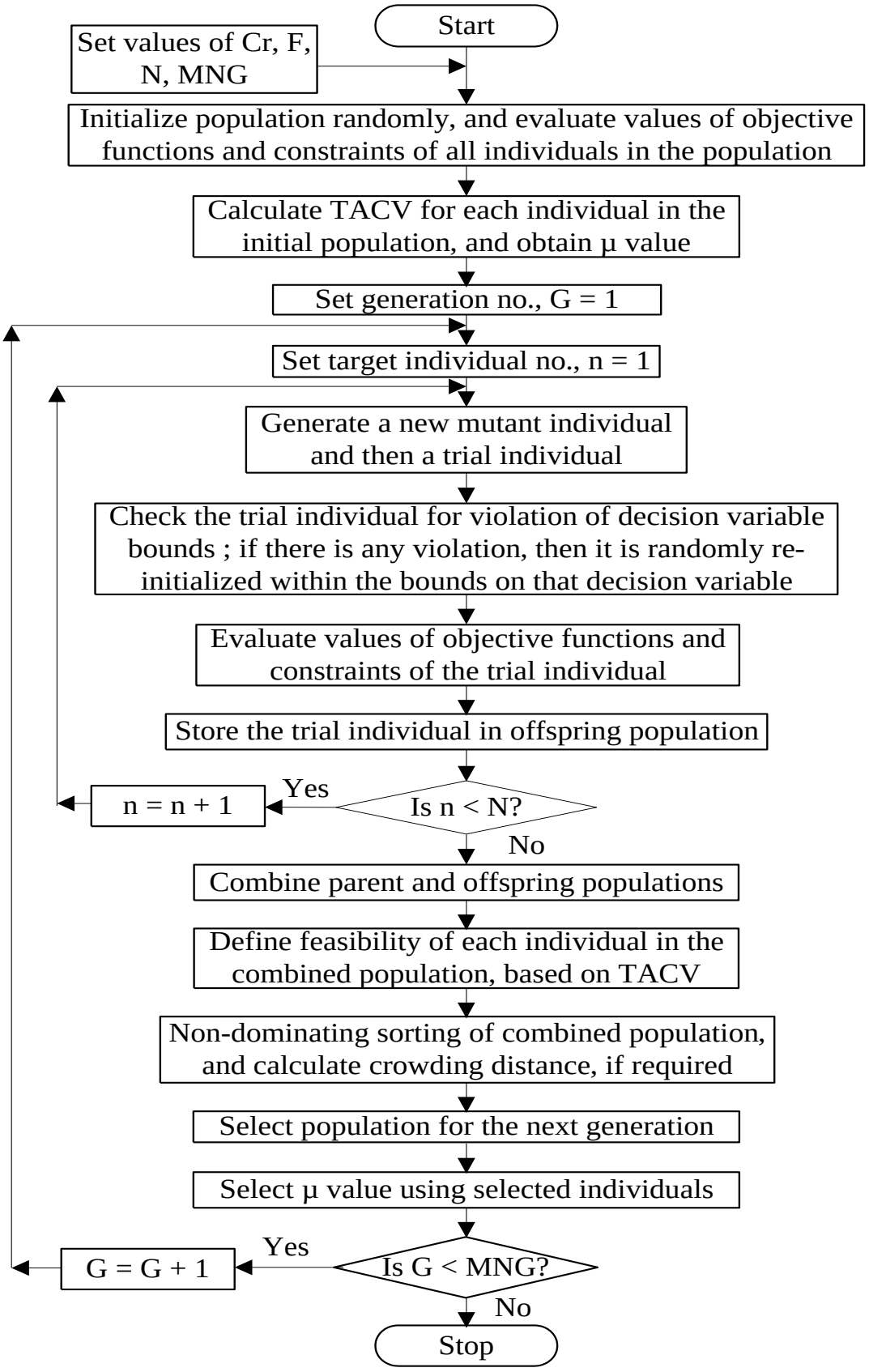


Figure 6.1: Flowchart for MODE-ACRFA algorithm

iii) Both the individuals are infeasible, but individual A has smaller number of violated constraints (and lesser TACV if both have the same number of violated constraints) compared to individual B.

TACV of the selected individuals for the next generation is used to update μ value, which is chosen such that 25% of selected individuals will be temporarily infeasible based on TACV. The new μ value is used to define the feasibility of the individuals in combined population in the next generation. The generations and stochastic search continue until the specified search termination criterion is met. Here, maximum number of generations (MNG) is the termination criterion (Figure 6.1), which is commonly used in stochastic algorithms.

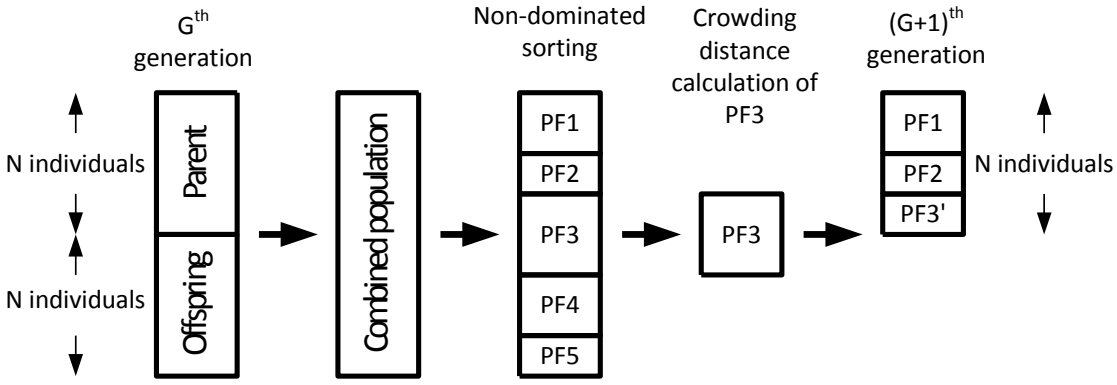


Figure 6.2: Selection of N individuals from the combined population of 2N individuals using Pareto dominance and crowding distance criteria

6.7 Testing of MODE-ACRFA on Developed Test Problems

There are many benchmark problems for testing MOO algorithms; these are with only bounds on decision variables (Zitzler et al., 2000) or with both bounds on variables and inequality constraints (Coello Coello et al., 2007). Interestingly, there seems to be no benchmark MOO problems with equality constraints. So, in this work, two inequality constrained MOO problems, namely, Viennet and Osyczka problems (Coello Coello et al., 2007) have been modified to equality constrained MOO

problems. For this, values of different inequality constraints corresponding to the complete Pareto-optimal front have been analyzed. If an inequality constraint is active or has nearly constant value, then it is converted to an equality constraint. The modified test problems are given in Table 6.1.

Table 6.1: Modified MOO test functions with equality constraints

Test problem	Decision variables	Objective functions (minimize)	Constraints
Modified Viennet	$-4 < x_1 < 4$ $-4 < x_2 < 4$	$f_1 = (x_1 - 2)^2/2 + (x_2 + 1)^2/13 + 3$ $f_2 = (x_1 + x_2 - 3)^2/175 + (2x_2 - x_1)^2/17 - 13$ $f_3 = (3x_1 - 2x_2 + 4)^2/8 + (x_1 - x_2 + 1)^2/27 + 15$	$4x_1 + x_2 - 4 = 0$ $-x_1 - 1 < 0$ $x_1 - x_2 - 2 < 0$
Modified Osyczka	$2 < x_1 < 7$ $5 < x_2 < 10$	$f_1 = x_1 + x_2^2$ $f_2 = x_1^2 + x_2$	$x_1 + x_2 - 12 = 0$ $-x_1^2 - 10x_1 + x_2^2 - 16x_2 + 80 < 0$

Performance of MODE-ACRFA is compared with that of MODE with feasibility approach alone (MODE-FA). For MODE-FA, each equality constraint is converted into an inequality constraint as follows.

$$h_i(\mathbf{x}) = 0 \tag{6.4a}$$

$$TL - |h_i(\mathbf{x})| \geq 0 \tag{6.4b}$$

Here, TL is the tolerance limit of constraint violation, which depends on the terms in the equality constraint (e.g., flow rates can be large whereas mole fractions are between zero and unity).

Generational distance, GD^t (as defined in Chapter 3) is used to compare the performance of MODE-FA and MODE-ACRFA. Algorithm parameters used in the performance comparison for test functions are: F = 0.8, Cr = 0.9, N = 100 and MNG = 500; values of F and Cr are based on the recommendation in the literature (e.g., Chen

et al., 2010), while population size of 100 is reasonable for small problems with a few decision variables and constraints. A TL value of 1.0e-6 is used for relaxing equality constraints in Table 6.1 into inequality constraints for MODE-FA. Figures 6.3(a) and 6.4(a) show the variation in GD^t with generations on modified Viennet and Osyczka problems, respectively, using MODE-FA and MODE-ACRFA. Performance of both constraint handling approaches is comparable on the modified Viennet problem. Initially, MODE-FA shows faster convergence on the modified Osyczka problem, but performance of both approaches is comparable after 200 generations (Figure 6.4a).

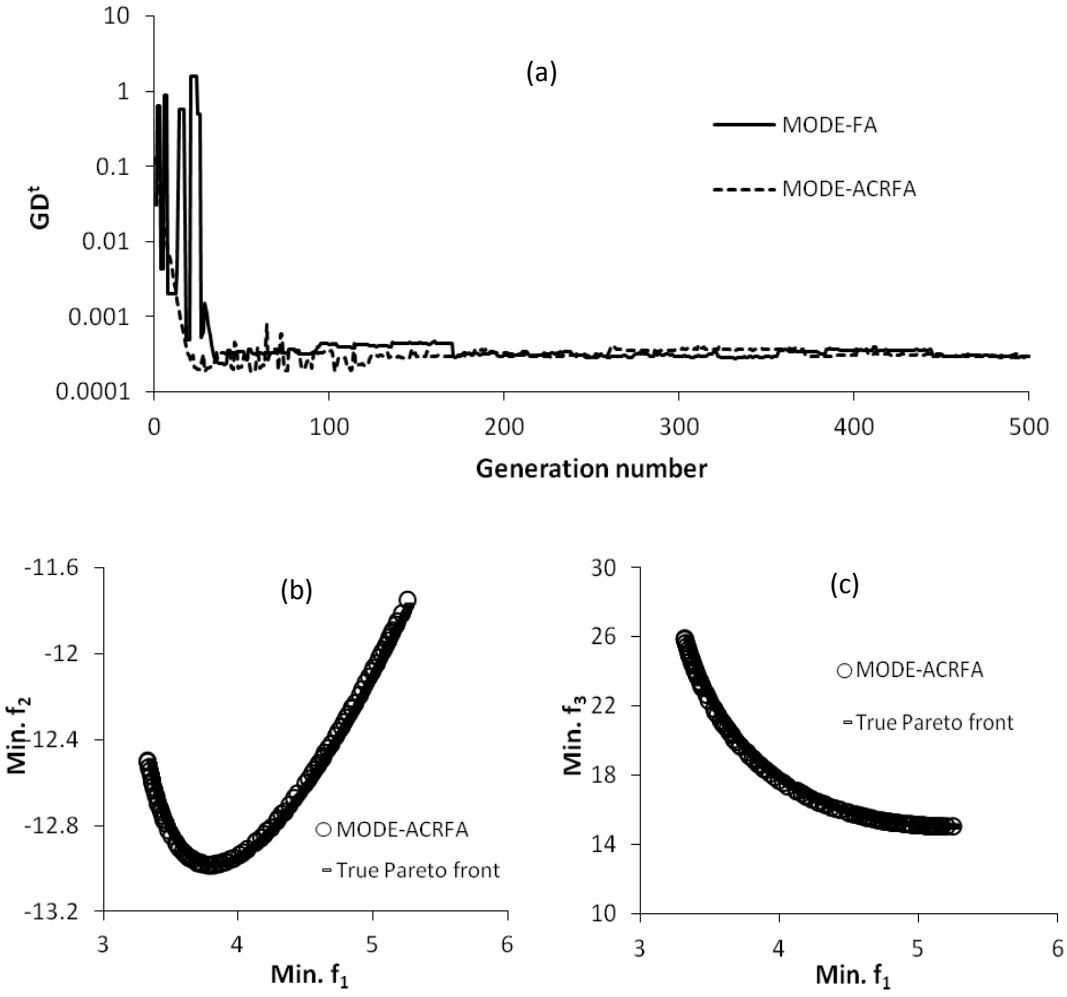


Figure 6.3: Performance of MODE-FA and MODE-ACRFA on modified Viennet problem

Moreover, the final Pareto-optimal fronts obtained for both the problems using ACRFA and FA are very close to the true/known Pareto-optimal fronts, as shown in Figures 6.3(b), 6.3(c) and 6.4(b). Figure 6.5 shows variation in μ with generations on modified Viennet and Osyczka problems using MODE-ACRFA; these follow the general trend of GD^t with generations in Figures 6.3(a) and 6.4(a). As expected, μ decreases with generations because better individuals (in terms of feasibility and objective values) are selected for the next generation.

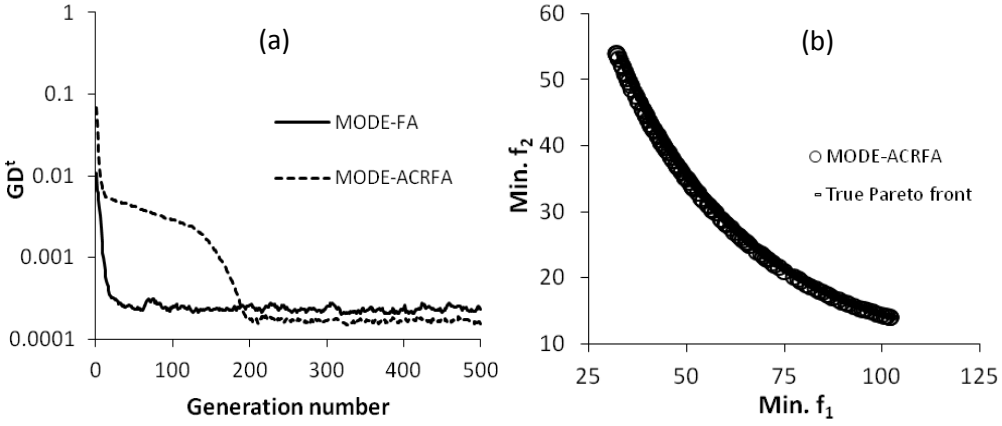


Figure 6.4: Performance of MODE-FA and MODE-ACRFA on modified Osyczka Problem

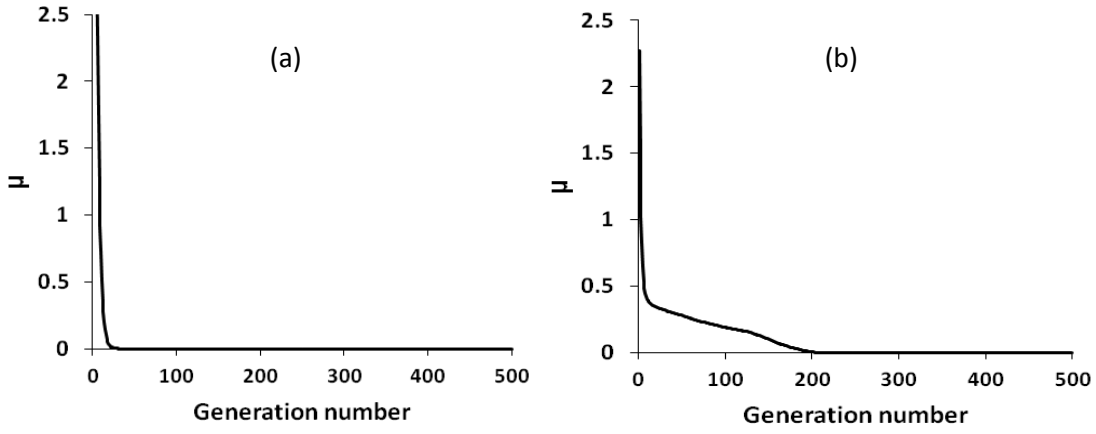


Figure 6.5: Variation in μ with generations in MODE-ACRFA on: (a) modified Viennet problem, and (b) modified Osyczka problem

6.8 Application of ACRFA on Fermentation Processes

Ethanol is widely used as a chemical and bio-fuel. Bio-ethanol production from sustainable feed-stocks is one of the possible alternatives to fossil fuel. Its production using first generation feed-stocks (e.g., glucose) is well established, while bio-ethanol production using second generation feed-stocks (e.g., starch and cellulose) is in the development phase. In this section, operation of a fermentation process integrated with cell recycling and a fermentation process integrated with cell recycling and inter-stage extraction is optimized for multiple objectives by both MODE-FA and MODE-ACRFA. Both these applications involve many equality constraints.

6.8.1 Three-stage Fermentation Process Integrated with Cell Recycling

Process model for this integrated fermentation process has been presented in Chapter 5. Glucose is used as feed-stock, and ethanol productivity and glucose conversion are the performance objectives. The MOO problem for the 3-stage continuous fermentation process integrated with cell recycling is summarized in Table 6.2. Decision variables for this optimization are dilution rate (D), glucose concentration in feed ($s_{f,1}$) and cell mass recycling for different stages (i.e., bleed ratios, b_1, b_2 and b_3). Physical constraints are same as those in Chapter 5. The model equations 5.3a-e for each stage are the equality constraints in the MOO problem. Of these, equations 5.3d-e can be substituted in equations 5.3a-c. Then, there will be 3 equality constraints for each stage or 9 equality constraints for the 3-stage fermentation process.

Wang and Lin (2010) have solved the MOO problem in Table 6.2 using fuzzy goal attainment method, which requires preference intervals for objectives and constraints. Finally, it is solved as a SOO problem using hybrid differential evolution

(HDE). Adaptive penalty function approach has been used for constraint handling in HDE. Same optimization approach (i.e., HDE with adaptive penalty function approach) is also used to solve the MOO problem of extractive fermentation process (Section 6.8.2). In this work, MOO problem has been solved by three different strategies, all using MODE. Each strategy differs in the handling of constraints, as described below. The present approach provides many Pareto-optimal solutions for better understanding and selection of one of them.

Table 6.2: MOO problem formulation for the three-stage continuous fermentation process integrated with cell recycling; k = 1, 2, 3

Objective functions	Max. ethanol productivity, [kg/(m^3.h)]	$\frac{D}{3}p_3$
	Max. overall glucose conversion	$1 - \frac{s_3}{s_{f,1}}$
Decision variables	Dilution rate, [1/h]	$3.5 \leq D \leq 4$
	Glucose concentration in feed, [kg/m^3]	$60 \leq s_{f,1} \leq 65$
	Bleed ratio for each stage	$0.1 \leq b_1 = b_2 = b_3 \leq 0.2$
Constraints	Same as defined in Table 5.2 (Chapter 5)	

(A) MODE-Solver and FA: In this approach, MODE algorithm has been used to generate the vector of 3 decision variables (i.e., D, S_f, and b_k). In order to calculate objectives and constraints, material balance equations 5.3a-c for each stage have to be solved; Solver tool in Excel is used to solve these equations (see Figure 6.6). Feasibility approach is used to handle inequality constraints in the optimization problem (total of 10 inequality constraints in Table 6.2 excluding the model equations). This strategy is used to illustrate the use of sequential solution of optimization problem and equality constraints (i.e., model equations), and to obtain the Pareto-optimal solutions for comparison.

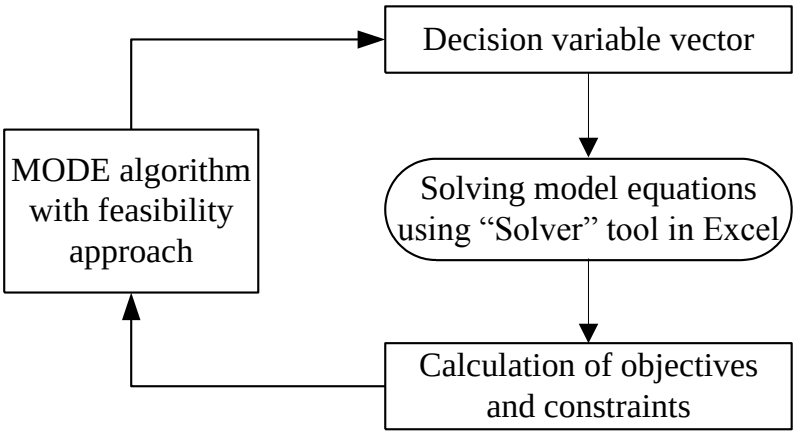


Figure 6.6: Flowchart for calculation of objective functions and constraints using Solver tool in Excel for solving model equations

(B) MODE-FA: In this approach, material balance equations 5.3a-c for each stage are converted into inequality constraints; conversion of each equality constraint to an inequality constraint is done based on equations 6.4a-b and using the same value of TL. Finally, the reformulated MOO problem has 19 inequality constraints (10 inequality constraints in Table 6.2, and 9 inequality constraints from material balances for each stage). For this and next strategy, additional decision variables are cell mass, glucose and ethanol concentrations for each stage. These variables with their bounds are presented in Table 6.3; non-dominated solutions obtained, using strategy A, are used to choose suitable bounds on the additional decision variables. Thus, number of decision variables in this and next strategy is 12.

(C) MODE-ACRFA: This approach can handle equality constraints without any conversion. It has 9 equality and 10 inequality constraints. Decision variables are the same as those in MODE-FA. Optimization problem and equality constraints are solved simultaneously in both MODE-FA and MODE-ACRFA. However, solution of equality constraints in MODE-FA is not exact due to relaxation by TL, and so its optimization results can differ from the other two strategies.

Table 6.3: Additional decision variables and their bounds for optimization strategies B and C (continuous fermentation)

Decision variable	x_1	x_2	x_3	s_1	s_2	s_3	p_1	p_2	p_3
Lower bound	40	80	90	10	0	0.1	10	20	20
Upper bound	70	110	110	30	10	0.5	30	40	40

MODE parameters used in this optimization study are given in Table 6.4. F value is tuned through preliminary experimentation, whereas Cr value is based on the recommendation in the literature (see Chen et al., 2010). The population size of 100 is used in solution strategy A, while population size in solution strategies B and C is 15 times sum of number of decision variables and constraints. MNG used for different strategies is based on preliminary experimentation.

Table 6.4: MODE algorithm parameter values used in MOO of fermentation processes

Algorithm parameter	**Strategy A: MODE-Solver with FA**	**Strategy B: MODE-FA**	**Strategy C: MODE-ACRFA**
F	0.9	0.9	0.9
Cr	0.9	0.9	0.9
N	100	465	465
MNG	100	5000	5000

Figure 6.7(a) shows Pareto-optimal front obtained for the 3-stage continuous fermentation process integrated with cell recycling using the strategy A. As expected, ethanol productivity is conflicting with glucose conversion. The obtained Pareto-optimal front is well distributed, and dilution rate is mainly contributing to the variations in the objective functions. Glucose concentration in the feed and bleed ratios are nearly constant, and they are near to their upper (i.e., 65) and lower (i.e., 0.1) bounds respectively. For brevity, bleed ratios are not shown in Figure 6.7. The

obtained Pareto-optimal front in Figure 6.7(a) is nearly linear in shape. In the operation optimization considered, production capacity is sufficiently large to convert glucose completely for any dilution rate in the range 3.5 to 4; hence, glucose concentration in the feed is always near to its upper limit (see Figure 6.7c). Further, increase in dilution rate increases ethanol productivity as larger amount of glucose enters into fermentor although glucose conversion decreases due to lower residence time. Both objectives are linearly dependent on dilution rate since the quantities (i.e., dilution rate, ethanol concentration in product stream, glucose concentration in feed and residual glucose) involved in the objectives are directly related to the dilution rate (see Table 6.2).

Figure 6.7(a) also shows the Pareto-optimal front obtained for the same continuous fermentation process using MODE-FA. These optimization results are obtained using TL of 3.0; MODE-FA did not give any feasible solution with TL of 1.0 or smaller. The non-dominated solutions obtained in Figure 6.7(a) have average absolute constraint violations (AACV) of 2.15. Here, variation in ethanol productivity is smaller compared to the Pareto-optimal front obtained using strategy A (Figure 6.7a). Figures 6.7(b) and 6.7(c) show respectively the variations in dilution rate and glucose concentration in the feed with ethanol productivity. Bleed ratio is constant near to 0.1, and so this is not shown in Figure 6.7. Variations in the remaining decision variables for strategies B and C are also not presented as these are not essential for performance comparison. Pareto-optimal front obtained for the 3-stage continuous fermentation process integrated with cell recycling using MODE-ACRFA is also shown in Figure 6.7(a). Here, both objectives are varying in narrow ranges compared to the other two; AACV for all the non-dominated solutions obtained by MODE-ACRFA is 0.013, which is much smaller than that by MODE-FA.

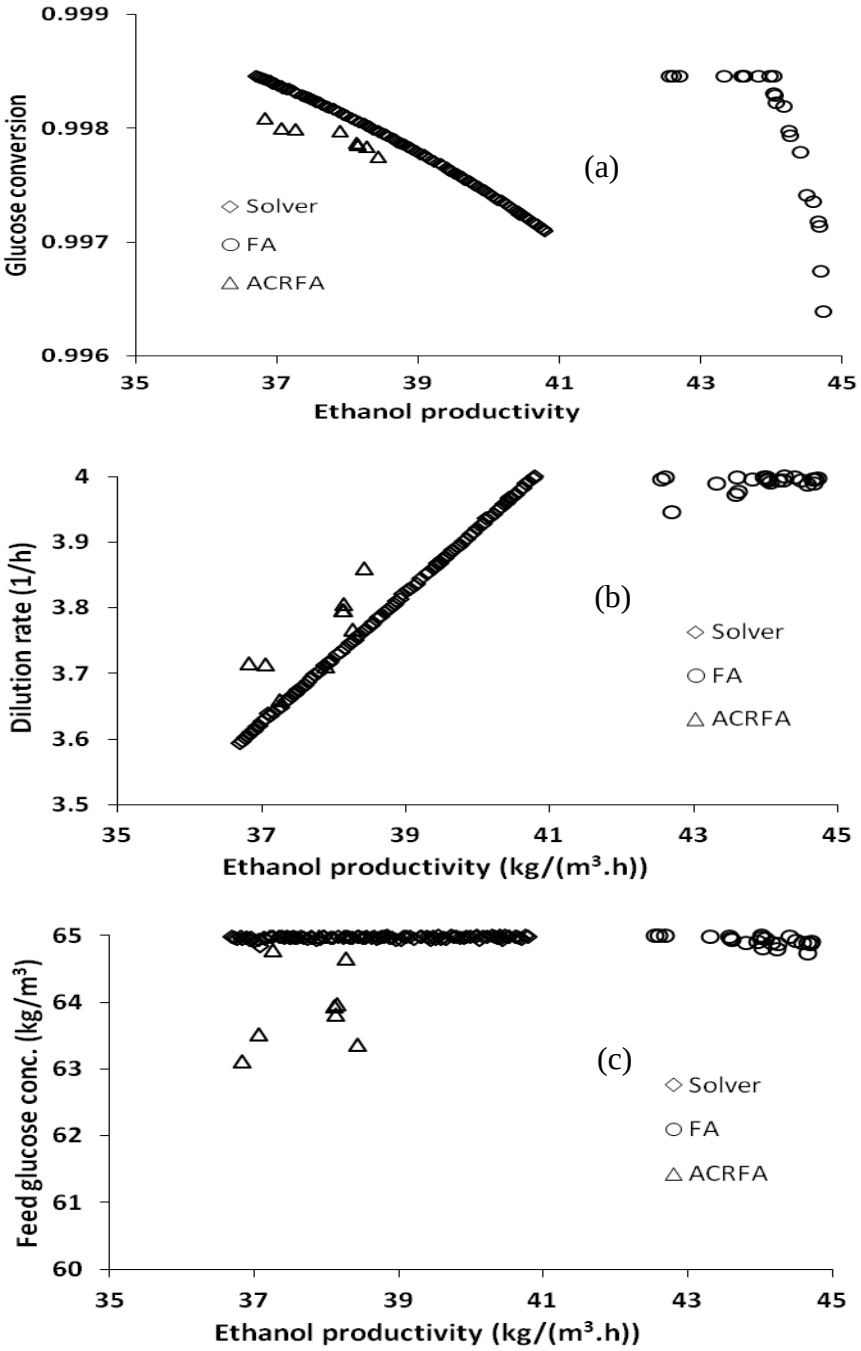


Figure 6.7: Selected optimization results for 3-stage continuous fermentation process integrated with cell recycling, using strategies A (Solver), B (FA), and C (ACRFA)

Figure 6.7(a) can also be used for comparing the non-dominated solutions obtained for the 3-stage continuous fermentation process integrated with cell recycling using three different optimization strategies. In this comparison, Pareto-optimal front obtained by MODE-Solver with FA can be considered as the correct front. It can be seen that non-dominated solutions obtained using MODE-FA are

significantly far from the correct Pareto-optimal front; also they have high value of AACV (= 2.15), and so they are not the optimal solutions satisfying all constraints. On the other hand, non-dominated solutions obtained by MODE-ACRFA are close to the correct Pareto-optimal front, and they have a much lower value of AACV (= 0.013), which is acceptable in engineering applications.

6.8.2 Three-stage Fermentation Process Integrated with Cell Recycling and Extraction

Chen and Wang (2010b) have studied a three-stage fermentation process integrated with cell recycling and inter-stage-extraction using a mixture of glucose and xylose as feed-stocks (referred to as extractive fermentation process). Figure 6.8 shows a schematic diagram of this fermentation process. Ethanol concentration inhibits conversion of glucose and xylose to ethanol in the fermentor, which results in lower ethanol productivity and yield. To avoid this, ethanol can be continuously removed from the fermentor, for example, using extraction. In the present study, three fermentors are placed in series, with feed entering into the first fermentor only. A part of mother liquor from a fermentor goes directly to the next fermentor while the remaining goes through a cell separator and an extractor. A cell separator is used after each fermentor to separate the cell mass and recycle it back to the fermentor, whereas an extractor is used to extract ethanol using an organic solvent. After extraction of ethanol, mother liquor goes to the next fermentor. Extractor is not necessary in the last/third stage of the fermentation process (see Figure 6.8).

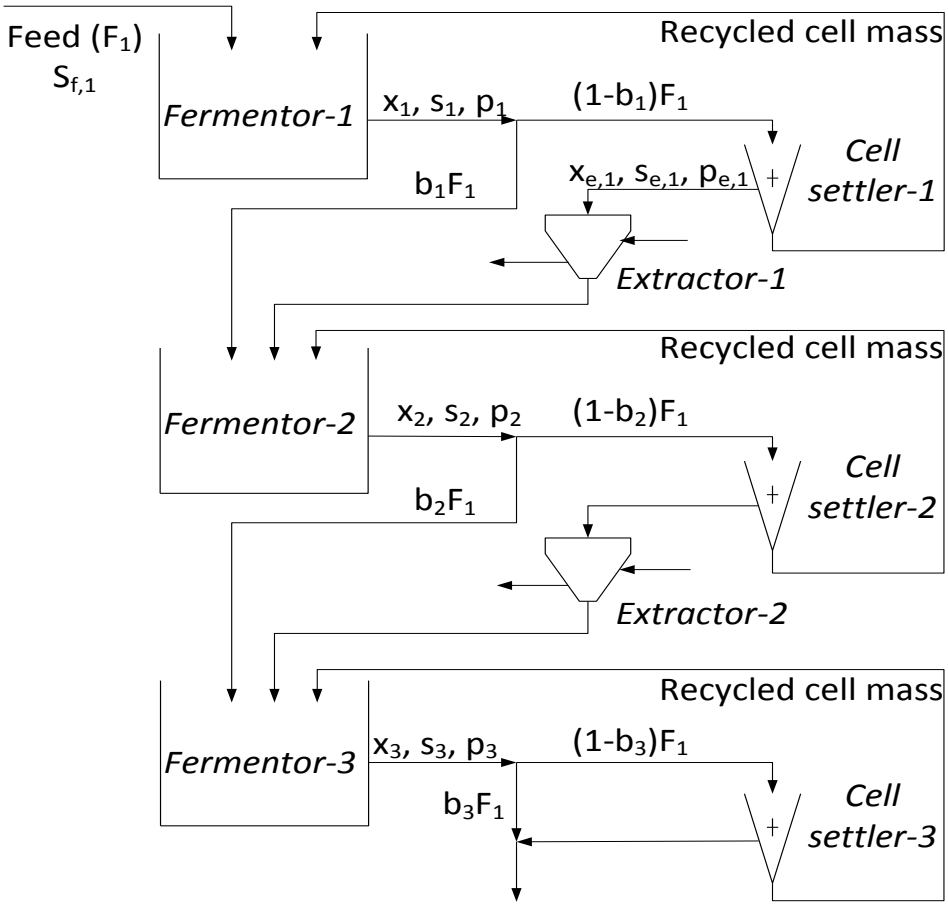


Figure 6.8: Schematic diagram of a three-stage fermentation process integrated with cell recycling and extraction

The mathematical model of the three-stage extractive fermentation process is taken from Chen and Wang (2010b). Equations 6.5a-d present steady-state mass balances for cell mass, glucose, xylose and ethanol around k^{th} stage, respectively.

$$D\left[b_{k-1} + (1-b_{k-1})\zeta_x\right]x_{k-1} - D\left[b_k + (1-b_k)\zeta_x\right]x_k + r_{x,k} = 0$$

$$D\lambda s_{f,k} + D\left[b_{k-1} + (1-b_{k-1})\zeta_s\right]s_{g,k-1} - D[b_k + (1-b_k)\zeta_s]s_{g,k} - r_{sg,k} = 0$$

$$D(1-\lambda)s_{f,k} + D\left[b_{k-1} + (1-b_{k-1})\zeta_s\right]s_{x,k-1} - D[b_k + (1-b_k)\zeta_s]s_{x,k} - r_{sx,k} = 0$$

$$D\left[b_{k-1} + (1-b_{k-1})\frac{\zeta_p}{1+E_{k-1}}\right]p_{k-1} - D\left[b_k + (1-b_k)\zeta_p\right]p_k + r_{p,k} = 0 \qquad (6.5a\text{-}d)$$

In the above equations, D is the dilution rate. x_k, $s_{g,k}$, $s_{x,k}$ and p_k are respectively cell mass, glucose, xylose and ethanol concentration (kg/m^3) in the k^{th} stage fermentor. b_k

is the bleed ratio for k^{th} stage, and $s_{f,k}$ is the substrate concentration in feed entering the k^{th} stage. Here, feed is entering only into the first fermentor, and so values of $s_{f,2}$ = 0 and $s_{f,3}$ = 0 for second and third stages respectively. λ is the mass fraction of glucose in substrate (and the remaining is xylose). ζ_x, ζ_s and ζ_p are cell discard factors (e.g., $x_{e,1}/x_1 = 0.01$), substrate condensed factors (e.g., $s_{e,1}/s_1 = 1.01$) and ethanol condensed factors (e.g., $p_{e,1}/p_1 = 1.01$) respectively (see Figure 6.8); these factors define relative concentrations of cell mass, substrate and ethanol in mother liquor after cell separation compared to those after the fermentor. E_k is the extraction efficiency for k^{th} stage. Further, for the first stage, b_{k-1}, x_{k-1}, $s_{g,k-1}$, $s_{x,k-1}$, p_{k-1} and E_{k-1} are also zero. The rate expressions for cell mass growth ($r_{x,k}$), glucose consumption ($r_{sg,k}$), xylose conversion ($r_{sx,k}$) and ethanol production ($r_{p,k}$) are as follows.

$$r_{x,k} = \mu_{mix,k} x_k \tag{6.6a}$$

$$r_{sg,k} = \frac{1}{Y_{p/sg}} \nu_{g,k} x_k \tag{6.6b}$$

$$r_{sx,k} = \frac{1}{Y_{p/sx}} \nu_{x,k} x_k \tag{6.6c}$$

$$r_{p,k} = (\nu_{g,k} + \nu_{x,k}) x_k \tag{6.6d}$$

$$\mu_{g,k} = \frac{\mu_{mg} s_{g,k}}{K_g + s_{g,k} + s_{g,k}^2/K_{ig}} \left\{ 1 - \left(\frac{p_k}{p_{mg}} \right)^{\phi_g} \right\} \tag{6.6e}$$

$$\mu_{x,k} = \frac{\mu_{mx} s_{x,k}}{K_x + s_{x,k} + s_{x,k}^2/K_{ix}} \left\{ 1 - \left(\frac{p_k}{p_{mx}} \right)^{\phi_x} \right\} \tag{6.6f}$$

$$\mu_{mix} = \frac{s_{g,k}}{s_{g,k} + s_{x,k}} \mu_{g,k} + \frac{s_{x,k}}{s_{g,k} + s_{x,k}} \mu_{x,k} \tag{6.6g}$$

$$\nu_{g,k} = \frac{\nu_{mg} s_{g,k}}{K'_g + s_{g,k} + s_{g,k}^2/K'_{ig}} \left\{ 1 - \left(\frac{p_k}{p'_{mg}} \right)^{\varphi_g} \right\} \tag{6.6h}$$

$$\nu_{x,k} = \frac{\nu_{mx} s_{x,k}}{K'_x + s_{x,k} + s_{x,k}^2 / K'_{ix}} \left\{ 1 - \left(\frac{p_k}{p'_{mx}} \right)^{\varphi_x} \right\} \quad (6.6i)$$

Here, μ_{mix} is the specific cell growth rate for the yeast 1400 (pLNH33) on glucose-xylose mixture. For this yeast, ν_g and ν_x are the specific ethanol production rates of glucose and xylose, respectively. The kinetic parameters in equations 6.6a-i are taken from Krishnan et al. (1999), and are reported in Table 6.5.

Table 6.5: Kinetic parameters and their values for the extractive fermentation process (Krishnan et al., 1999)

Kinetic parameters	**Estimated values**	**Kinetic parameters**	**Estimated values**
μ_{mg}, μ_{mx} (h^{-1})	0.662, 0.190	p_{mg}, p_{mx} (kg/m^3)	95.4, 59.04
ν_{mg}, ν_{mx} (h^{-1})	2.005, 0.250	p'_{mg}, p'_{mx} (kg/m^3)	103.03, 60.2
K_g, K_x (kg/m^3)	0.565, 3.4	ϕ_g, ϕ_x	1.29, 1.036
K_{ig}, K_{ix} (kg/m^3)	283.7, 18.1	φ_g, φ_x	1.42, 0.608
K'_g, K'_x (kg/m^3)	1.324, 3.4	$Y_{p/sg}$, $Y_{p/sx}$ (kg/kg)	0.47, 0.40
K'_{ig}, K'_{ix} (kg/m^3)	4890, 81.3		

The MOO problem formulation for the 3-stage extractive fermentation process is given in Table 6.6. In this case, ethanol productivity and xylose conversion are considered as objectives. Glucose conversion is not used as an objective because it is always higher than xylose conversion; it is used as an additional constraint in the optimization problem. Dilution rate and substrate concentration in the feed are the decision variables. Bleed ratios for different stages are not considered as decision variables as low values are optimal, based on Section 6.8.1; so, bleed ratio for each of the stages is fixed at 0.2. Here, positive values of ethanol productivity of each stage, glucose and xylose conversions in each stage are physical constraints. Other constraints are total sugar supply ($s_T < 180$) and limits on the residual glucose and

xylose concentrations ($s_{g,3} < 0.5$ and $s_{x,3} < 1$) in the mother liquor from the third fermentor (Chen and Wang, 2010b). The model equations 6.5a-d and 6.6a-i for each stage are the equality constraints in the MOO problem. Of these, equations 6.6a-i can be substituted in equations 6.5a-d. Then, there will be 4 equality constraints for each stage or 12 equality constraints for the 3-stage extractive fermentation process.

Table 6.6: MOO problem formulation for the extractive fermentation process; k = 1, 2, 3

Objective functions	
Max. ethanol productivity, [kg/(m^3.h)]	$\frac{D}{3}\left[b_3 + (1-b_3)\zeta_p\right]p_3 + \sum_{k=1}^{2}(1-b_k)\frac{\zeta_p E}{(1+E)}p_k$
Max. overall xylose conversion	$1 - \frac{[b_3+(1-b_3)\zeta_s]s_{x,3}}{(1-\lambda)s_{f,1}}$
Decision variables	
Dilution rate, [1/h]	$0.3 \le D \le 0.8$
Substrate conc. in feed, [kg/m^3]	$90 \le s_{f,1} \le 95$
Constraints	
Productivity in each stage, [kg/(m^3.h)]	$D\left[\left\{b_k + (1-b_k)\zeta_p\right\}p_k - \left\{b_{k-1} + \frac{1-b_{k-1}}{1+E_{k-1}}\zeta_p\right\} \times p_{k-1}\right] \ge 0$
Glucose conversion in each stage	$1 - \frac{\{b_k+(1-b_k)\zeta_s\}s_{g,k}}{\lambda s_{f,k}+\{b_{k-1}+(1-b_{k-1})\zeta_s\}s_{g,k-1}} \ge 0$
Xylose conversion in each stage	$1 - \frac{\{b_k+(1-b_k)\zeta_s\}s_{x,k}}{(1-\lambda)s_{f,k}+\{b_{k-1}+(1-b_{k-1})\zeta_s\}s_{x,k-1}} \ge 0$
Residual glucose, [kg/m^3]	$s_{g,3} \le 0.1$
Residual xylose, [kg/m^3]	$s_{x,3} \le 1$
Total glucose supplied per unit volume, [kg/(m^3.h)]	$s_T \equiv \frac{Ds_{f,1}}{3} \le 180$
Glucose conversion overall	$1 - \frac{[b_3+(1-b_3)\zeta_s]s_{g,3}}{\lambda s_{f,1}} > 0.99$
Model for the process	Equations 6.5a-d and 6.6a-i for each stage

The MOO problem, in Table 6.6 is solved using three different strategies described in Section 6.8.1. The problem for strategy A (using Solver and FA) has 2 decision variables and 13 inequality constraints. Reformulation of the problem for MODE-FA has 12 additional inequality constraints arising from material balances around each and every stage (in total, 25 inequality constraints). Problem for MODE-ACRFA has 13 inequality and 12 equality constraints. Moreover, the problem for both MODE-FA and MODE-ACRFA has 12 additional decision variables (i.e., cell mass, glucose, xylose and ethanol concentrations for each of the 3 stages); these variables and their bounds are listed in Table 6.7. Very wide bounds for additional decision variables will result in slow convergence of the algorithm; hence, non-dominated solutions obtained using solution strategy A are used to choose the bounds on the additional decision variables.

Table 6.7: Additional decision variables and their bounds for optimization strategies B and C (extractive fermentation)

Decision variable	x_1	x_2	x_3	s_{g1}	s_{g2}	s_{g3}	s_{x1}	s_{x2}	s_{x3}	p_1	p_2	p_3
Lower bound	0	20	50	0	0	0	20	0	0	20	10	0
Upper bound	20	60	70	10	1	10	40	20	1	40	30	20

The feed contains 65% glucose and 35% xylose, and extraction efficiency for each stage is 6.93 which is equivalent to 87.4% of ethanol removal from the mother liquor (Chen and Wang, 2010b). The MODE algorithm parameters used in the optimization of extractive fermentation process are same as those in Table 6.4, except value of N for strategies B and C is 585 (i.e., 15 times number of decision variables and constraints).

Figure 6.9(a) show the Pareto-optimal front obtained for the 3-stage extractive fermentation process using optimization strategy A (see Section 6.8.1). The obtained

Pareto-optimal front can be divided into two parts: (i) improvement in ethanol productivity from 5.4 to 5.8 kg/(m3.h) with a small decrease in xylose conversion, and (ii) a linear change between ethanol productivity and xylose conversion (~ 0.985-0.97). In the first part, the improvement in ethanol productivity, is mainly due to fast change in substrate concentration in feed, while dilution rate is mainly affecting the objectives in the second part (see Figures 6.9a-c). For a fixed production capacity, increase in substrate concentration in the feed produces more ethanol, but it does not affect residence time in the fermentor. Hence, increase in ethanol productivity is relatively faster compared to decrease in xylose conversion (see the first part of the Pareto-optimal front in Figure 6.9a).

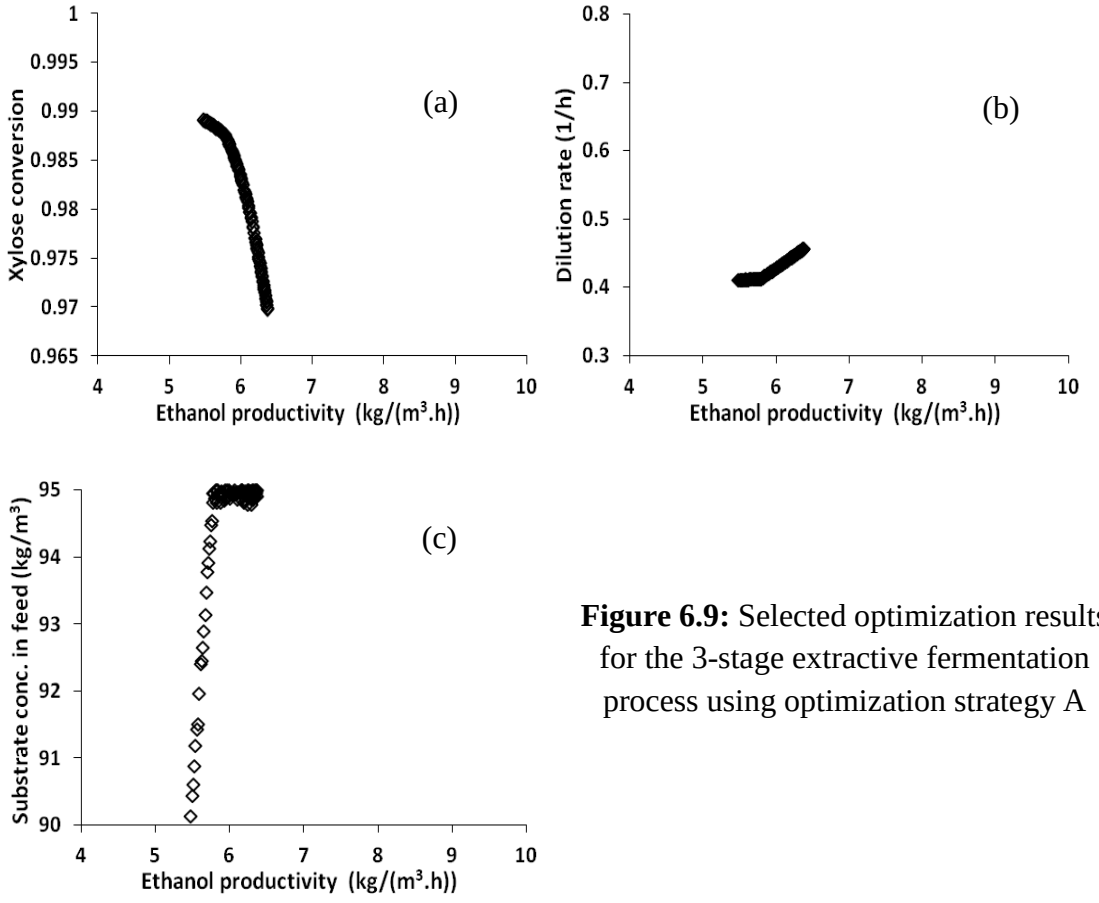


Figure 6.9: Selected optimization results for the 3-stage extractive fermentation process using optimization strategy A

Increase in dilution rate also increases ethanol productivity as larger amount of glucose enters into fermentors, but substrate conversion decreases relatively faster with increase in dilution rate due to lower residence time (see the second part of the

obtained Pareto-optimal front in Figure 6.9a). In conclusion, a relatively fast increase in ethanol productivity is achieved initially by an increase in substrate concentration in the feed, until substrate concentration reached availability limit. In the present operation optimization case, ethanol production facility is sufficient to convert feed, at its maximum available concentration (i.e., 95 kg/m^3), into product, and keeps the unreacted substrate in the product stream below the required limit.

Figure 6.10(a) shows the Pareto-optimal front obtained for the 3-stage extractive fermentation process using MODE-FA. These non-dominated solutions are obtained using TL of 1.0; MODE-FA is not able to give any feasible solution with a smaller value of TL. The non-dominated solutions in Figure 6.10(a) have AACV of 0.608. Variations in objectives can be visually correlated to the variation in dilution rate with ethanol productivity (Figure 6.10b), while substrate concentration in feed is scattered between its lower and upper bounds (Figure 6.10c). The Pareto-optimal front obtained by MODE-ACRFA is shown in Figure 6.10(d). Here, both objectives are varying in relatively narrow ranges compared to the Pareto-optimal front obtained using the other two strategies. AACV for all non-dominated solutions obtained using MODE-ACRFA is 0.022, which is acceptable for engineering applications. Values and trends in the Pareto-optimal front and decision variables in Figures 6.10(d) to (f) are similar to those obtained by strategy A (Figure 6.9).

Finally, Figure 6.11 compares the Pareto-optimal fronts obtained for extractive fermentation process using the three different optimization strategies. Pareto-optimal front obtained by strategy A (i.e., MODE-Solver-FA) can be considered as the correct solution to this problem. It can be seen that MODE-FA gives wide ranges of both objectives, but these non-dominated solutions have a large value of AACV, and so

they are incorrect and unacceptable. The non-dominated solutions obtained by MODE-ACRFA are closer to the correct Pareto-optimal front, and cover most part of the correct Pareto-optimal front except a small part corresponding to higher xylose conversion.

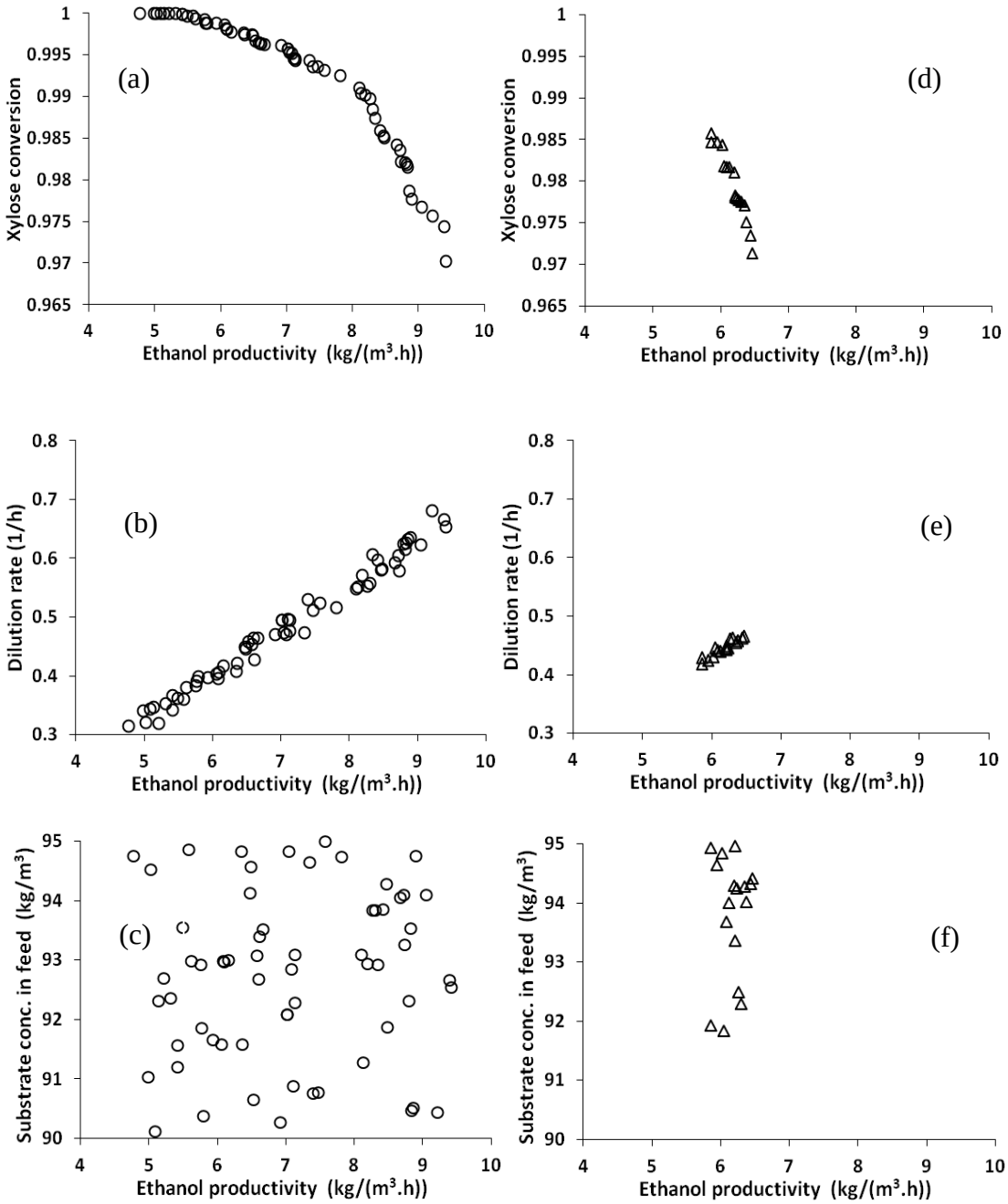


Figure 6.10: Selected optimization results for the 3-stage extractive fermentation process using MODE-FA (plots a, b and c in the left column), and using MODE-ACRFA (plots d, e and f in the right column)

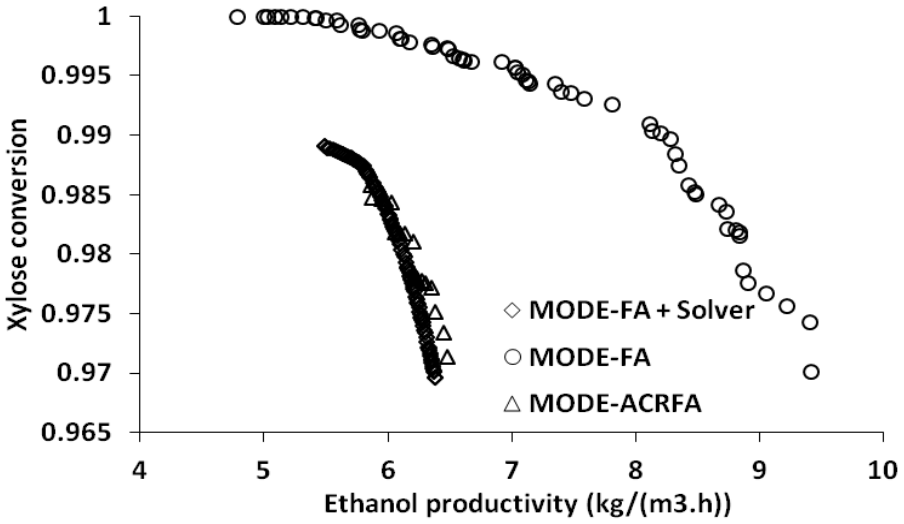


Figure 6.11: Comparison of the Pareto-optimal fronts obtained for the 3-stage fermentation process integrated with cell recycling and inter-stage extraction, using three different optimization strategies

The approximate time required for solving the MOO problem for the continuous fermentation process using strategies A, B and C is respectively 1, 8 and 3 hours on an Intel® Core™2 Duo Processor (CPU 2.8 & 2.8 GHZ and RAM 4 GB). Mflops (million floating point operations per second) on this computer is 537 for the LINPACK benchmark program for a matrix of order 500 (http://www.netlib.org/benchmark/linpackjava/). Optimization of the extractive fermentation process requires around 2, 12 and 4 hours by strategies A, B and C respectively, using same computer. In the case of MODE-FA and MODE-ACRFA, required computational time is larger due to larger population size and MNG. Hence, strategy A using Solver for sequential solution of equality constraints seems to be better followed by strategy C (i.e., using adaptive constraints relaxation with FA) for simultaneous solution of optimization problems and equality constraints.

6.9 Conclusions

Although two MOO test functions with equality constraints are used in this chapter, these problems are small with a few decision variables and constraints, and

are easy to solve. Performance of FA and ACRFA approaches is comparable on these two test functions, modified to have equality constraints. Hence, it is difficult to observe the difference in the performance of these two strategies. Many MOO test functions with equality constraints are required for a comprehensive comparison between FA and ACRFA solution strategies. Use of application problems for testing purpose is not easy as it requires process knowledge and due to unavailability of true solution.

Three-stage continuous fermentation and three-stage extractive fermentation processes, which contain many equality constraints arising from mass balances, are optimized using three different strategies: solution of equality constraints using Solver with FA for inequality constraints, FA and ACRFA. Of these, MODE-Solver-FA is the most effective to solve both fermentation processes compared to FA and ACRFA. Non-dominated solutions obtained by MODE-Solver-FA are better, satisfy equality constraints almost exactly and so can be considered to be accurate. Feasibility approach requires a suitable value for relaxation, which affects the optimization results obtained, and it performed poorly compared to ACRFA on both the fermentation processes. Non-dominated solutions obtained by MODE-ACRFA have less average absolute constraint violations than those obtained by MODE-FA, and are closer to those obtained by MODE-Solver-FA.

Sequential solution of the optimization problem and equality constraints using Solver along with FA for inequality constraints, is the better strategy for the optimization of fermentation processes considered. However, it may not be efficient if solution of equality constraints is computationally intensive. In such cases, simultaneous solution of the optimization problem and equality constraints via ACRFA strategy may be suitable. Further research is required to improve ACRFA.

Chapter 7

Modeling and Multi-objective Optimization of Fermentation Processes[5]

7.1 Introduction

Cellulosic ethanol production can use non-food crops and inedible waste products to produce bio-ethanol that causes lesser air pollution compared to the conventional fuels. Bio-ethanol production from fermentable sugars (e.g., glucose, fructose and sucrose, etc.) is well established. On the other hand, production of bio-ethanol from starchy and cellulosic materials requires hydrolysis as an additional step to produce fermentable sugars. Hydrolysis step can be performed before the fermentation (separate/sequential hydrolysis and fermentation, SHF) or with the fermentation (simultaneous saccharification and fermentation, SSF). In SHF process, hydrolysis and fermentation can be performed at their respective optimal temperatures, but hydrolysis end products (i.e., glucose and cellobiose) inhibit the hydrolysis. Yingling et al. (2011) have improved the enzymatic starch hydrolysis in SHF process using multi-objective optimization. The alternate SSF process removes end-products inhibition by immediate consumption of end products of hydrolysis. Further, lignocellulosic feed-stocks produce both glucose and xylose after hydrolysis. Feed concentration, fermentation temperature and time affect the conversion of lignocellulosic feed-stocks into glucose and xylose (Chen and Wang, 2010a). In simultaneous saccharification and co-fermentation (SSCF) process, conversion of lignocellulosic materials to glucose and xylose, and their fermentation take place

[5] This chapter is based on the article: Sharma, S. and Rangaiah, G. P. (2012), Modeling and optimization of a fermentation process integrated with cell recycling and pervaporation for multiple objectives, Ind. Eng. Chem. Res., 51(15), pp. 5542-5551.

together. Chen and Wang (2010a) have optimized SSCF process using lignocellulosic feed-stocks.

Katahira et al. (2008) tried to increase ethanol productivity during fermentation of lignocellulosic feed-stocks by enhancing the xylose intake ability of the yeast. Continuous bio-ethanol production process can also be improved by cell recycling and in-situ ethanol removal. Cell recycling improves the performance of continuous fermentation process significantly, and it has been studied by researchers for many years (e.g., Wieczorek and Michalski, 1994; Paiva et al., 1996; Oliveira et al., 1999 and Wang and Lin, 2010). Bayrock and Ingledew (2001) mentioned that industry uses both batch and multi-stage continuous processes for producing ethanol. Wang and Lin (2010) optimized the multi-stage fermentation process with cell recycling for two objectives, namely, ethanol productivity and glucose conversion.

Ethanol concentration in the fermentor inhibits conversion of fermentable sugars to ethanol, which results in lower productivity; these can be improved by better fermentation kinetics and/or process design. Several researchers improved the fermentation process by in-situ removal of ethanol using different processes, e.g., pervaporation (Mahecha-Botero et al., 2006) and liquid-liquid extraction (e.g., Daugulis et al., 1991; Gyamerah and Glover, 1996; Silva et al., 1999 and Offeman et al., 2005). Lipnizki et al. (2000) summarized several studies on the coupling of pervaporation process with bioreactors. Pervaporation unit can be arranged either internally or externally to the bioreactor. Lipnizki et al. (2000) discussed the advantages and disadvantages of both these arrangements, and suggested the use of external pervaporation unit due to its high efficiency and easy maintenance. Mahecha-Botero et al. (2006) have studied a membrane fermentor (for selective ethanol

removal) with cell recycling, and found that continuous ethanol removal increases ethanol productivity and also stabilizes the fermentation process.

Extraction has also been considered for ethanol removal during continuous fermentation. Gyamerah and Glover (1996) used dodecanol as solvent to remove ethanol from fermentor; the improved process can produce ethanol from a feed with high glucose concentration without product inhibition. Offeman et al. (2005) have compared the performance of twelve different solvents for ethanol removal from aqueous phase (i.e., fermentor broth). The used solvent for extraction should be biocompatible, inert and also easy to separate from the ethanol-solvent mixture. It should also have high value of distribution coefficient (K = the ratio of weight percent of ethanol in the organic phase to the weight percent of ethanol in the aqueous phase). Chemical stability and difference in density between aqueous and organic phases are other important requirements for a suitable solvent (Offeman et al., 2005). Generally, distillation is used to recover the solvent which requires high energy input (Heerema et al., 2011). Chen and Wang (2010b) have optimized a multi-stage fermentation process coupled with cell recycling and inter-stage extraction process for multiple objectives using fuzzy approach and single objective optimization algorithm; they have optimized this process for three objectives, namely, ethanol productivity, glucose and xylose conversion. Notably, glucose conversion is always higher than xylose conversion, and is near to 1.

The present study models and optimizes a three-stage fermentation process integrated with cell recycling and pervaporation, which is promising but has not been reported in the literature. Pervaporation units have been used outside each fermentor for continuous removal of ethanol in order to avoid product inhibition. In this study, Pareto-optimal front generating technique (namely, multi-objective differential

evolution, MODE) is used to assess the trade-offs between objectives. The three-stage integrated fermentation process has been optimized for ethanol productivity and xylose conversion, using MODE. A three-stage fermentation process integrated with cell recycling and extraction is also optimized for ethanol productivity and xylose conversion, using MODE, and its performance is compared to the fermentation with pervaporation process. In order to assess the usefulness of pervaporation or extraction for three-stage fermentation process with cell recycling, a three-stage fermentation process with cell recycling only is also optimized. Net flow method is used to rank the non-dominated solutions obtained for the three-stage fermentation process integrated with cell recycling and pervaporation.

The next section of this article describes the design and modeling of the three-stage fermentation process integrated with cell recycling and pervaporation. Section 7.3 describes the multi-objective optimization problem formulation, and Section 7.4 describes MODE algorithm briefly. Section 7.5 presents and discusses the optimization results using MODE and I-MODE algorithms. Section 7.6 compares the performance of this process with a three-stage fermentation process integrated with extraction. In Section 7.7, net flow method is used to rank the solution set obtained for the three-stage fermentation process integrated with cell recycling and pervaporation. Finally, main findings of this study are summarized in the Conclusions section.

7.2 Modeling of Three-stage Fermentation Process Integrated with Cell Recycling and Pervaporation

A schematic diagram of the three-stage fermentation process coupled with cell recycling and pervaporation is presented in Figure 7.1; important quantities are also shown in this figure. Three fermentors are placed in series, and feed enters in the first

fermentor. Here, dilution rate (= flow rate/volume of each fermentor) is used to represent different flow rates. For example, dilution rate at the inlet of the first fermentor is: $D_{1,i}$ = [Feed, m^3h^{-1}]/[V_F, m^3]. Volume of each fermentor is taken to be 257.4 m^3 (Nandong et al., 2006). A small part of mother liquor from a fermentor goes directly to the next fermentor while remaining mother liquor goes through a cell separator. A cell separator is used after each fermentor to separate the cell mass; the separated cell mass contains small amounts of substrate and ethanol, and is recycled it back to the fermentor. Yeast (cell mass) flocculates at a fixed temperature (42 ^{0}C for pLNH33; Chen and Wang, 2010b) and settles down at the bottom of the cell separator.

In this work, external pervaporation unit is used to remove ethanol continuously from the fermentor broth; cell mass is separated from the mother liquor before it enters the pervaporation unit. If required, multiple external pervaporation units (NM_k) can be used with each fermentor. Volume and area of each external pervaporation unit are assumed to be 0.16 m^3 and 24 m^2 respectively (Sander and Soukup, 1988). A vacuum pump is required on the permeate side (i.e., sweep fluid) of the pervaporation unit which can maintain a pressure of about 1-2 kPa; ethanol is condensed on a cold surface maintained at -40 ^{0}C temperature (Mori and Inaba, 1990). Further, Walsh and Bungay (1990) designed a shallow-depth yeast settler, which would be huge in size for industrial capacity although cheaper material can be used to construct it. Nowadays, centrifuge is used to separate the cell mass from fermentor broth, which increases ethanol production cost (Nandong et al., 2006).

The mathematical model of the three-stage fermentation process integrated with cell recycling and pervaporation is developed based on the models discussed in Chen and Wang (2010b) and Mahecha-Botero et al. (2006). Perfect mixing is assumed in

the fermentor and on both sides of perm-selective membrane in external pervaporation units. As volume of each fermentor is fixed and ethanol is continuously removed from the fermentors, the dilution rate for the second and third fermentors will be different from the dilution rate of the first fermentor. Equation 7.1a (below) represents the dilution rates at the outlet of all three fermentors where the term $a(p_k - P_{Mo,k})$ is the rate of ethanol removal (kg/h) in the k^{th} stage pervaporation unit. The material balances for cell mass, glucose, xylose and ethanol around stage 1 (fermentor-1, cell separator-1 and pervaporation unit-1) are given by equations 7.1b-7.1f respectively. Equation 7.1f represents the ethanol balance on the sweep fluid side of the external pervaporation module. Similarly, equations 7.1g-7.1p describe different material balances around the second and third stages. Equation 7.1q relates inlet and outlet flow rates on sweep fluid side of k^{th} stage pervaporation unit. Significance of symbols in all these model equations is described after Equation 7.1q.

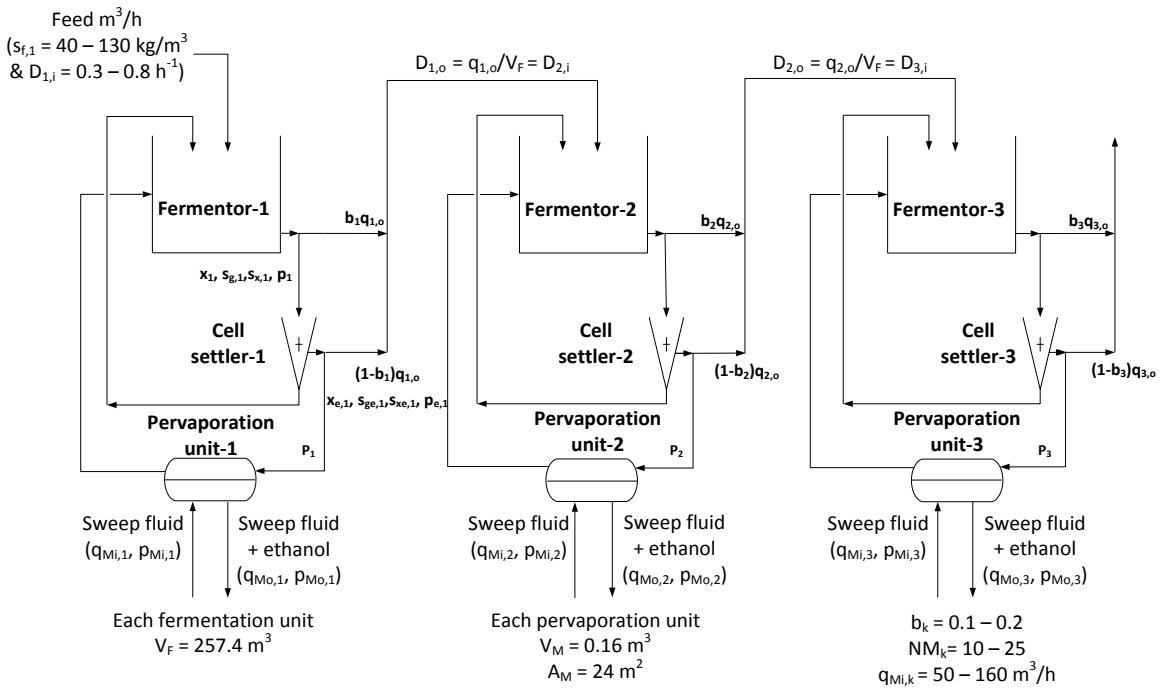


Figure 7.1: A schematic diagram of a three-stage fermentation process wherein each fermentor is coupled with cell settler and pervaporation unit

$$D_{1,o} = D_{1,i} - \frac{a(p_1 - p_{Mo,1})}{\rho V_F}; D_{2,o} = D_{1,o} - \frac{a(p_2 - p_{Mo,2})}{\rho V_F}; D_{3,o} = D_{2,o} - \frac{a(p_3 - p_{Mo,3})}{\rho V_F} \quad (7.1a)$$

Stage – 1 (Fermentor-1, Cell settler-1 and Pervaporation unit-1)

$$-D_{1,o}[b_1 + (1 - b_1)\zeta_x]x_1 + r_{x,1} = 0 \quad (7.1b)$$

$$D_{1,i}\lambda s_{f,1} - D_{1,o}[b_1 + (1 - b_1)\zeta_s]s_{g,1} - r_{sg,1} = 0 \quad (7.1c)$$

$$D_{1,i}(1 - \lambda)s_{f,1} - D_{1,o}[b_1 + (1 - b_1)\zeta_s]s_{x,1} - r_{sx,1} = 0 \quad (7.1d)$$

$$-D_{1,o}[b_1 + (1 - b_1)\zeta_p]p_1 - \frac{a}{V_F}(p_1 - p_{Mo,1}) + r_{p,1} = 0 \quad (7.1e)$$

$$q_{Mi,1}p_{Mi,1} - q_{Mo,1}p_{Mo,1} + a(p_1 - p_{Mo,1}) = 0 \quad (7.1f)$$

Stage – 2 (Fermentor-2, Cell settler-2 and Pervaporation unit-2)

$$D_{1,o}[b_1 + (1 - b_1)\zeta_x]x_1 - D_{2,o}[b_2 + (1 - b_2)\zeta_x]x_2 + r_{x,2} = 0 \quad (7.1g)$$

$$D_{1,o}[b_1 + (1 - b_1)\zeta_s]s_{g,1} - D_{2,o}[b_2 + (1 - b_2)\zeta_s]s_{g,2} - r_{sg,2} = 0 \quad (7.1h)$$

$$D_{1,o}[b_1 + (1 - b_1)\zeta_s]s_{x,1} - D_{2,o}[b_2 + (1 - b_2)\zeta_s]s_{x,2} - r_{sx,2} = 0 \quad (7.1i)$$

$$D_{1,o}[b_1 + (1 - b_1)\zeta_p]p_1 - D_{2,o}[b_2 + (1 - b_2)\zeta_p]p_2 - \frac{a}{V_F}(p_2 - p_{Mo,2}) + r_{p,2} = 0$$

$$q_{Mi,2}p_{Mi,2} - q_{Mo,2}p_{Mo,2} + a(p_2 - p_{Mo,2}) = 0 \quad (7.1j\text{-}k)$$

Stage – 3 (Fermentor-3, Cell settler-3 and Pervaporation unit-3)

$$D_{2,o}[b_2 + (1 - b_2)\zeta_x]x_2 - D_{3,o}[b_3 + (1 - b_3)\zeta_x]x_3 + r_{x,3} = 0 \quad (7.1l)$$

$$D_{2,o}[b_2 + (1 - b_2)\zeta_s]s_{g,2} - D_{3,o}[b_3 + (1 - b_3)\zeta_s]s_{g,3} - r_{sg,3} = 0 \quad (7.1m)$$

$$D_{2,o}[b_2 + (1 - b_2)\zeta_s]s_{x,2} - D_{3,o}[b_3 + (1 - b_3)\zeta_s]s_{x,3} - r_{sx,3} = 0 \quad (7.1n)$$

$$D_{2,o}\left[b_2 + (1-b_2)\zeta_p\right]p_2 - D_{3,o}\left[b_3 + (1-b_3)\zeta_p\right]p_3 - \frac{a}{V_F}(p_3 - p_{Mo,3}) + r_{p,3} = 0$$

$$q_{Mi,3}p_{Mi,3} - q_{Mo,3}p_{Mo,3} + a(p_3 - p_{Mo,3}) = 0 \qquad (7.1o\text{-}p)$$

$$q_{Mo,k} = q_{Mi,k} + \frac{a(p_k - p_{Mo,k})}{\rho} \qquad (7.1q)$$

In the above equations, $D_{k,i}$ and $D_{k,o}$ (1/h) are respectively the inlet and outlet dilution rates for k^{th} stage. x_k, $s_{g,k}$, $s_{x,k}$ and p_k are respectively cell mass, glucose, xylose and ethanol concentration (kg/m^3) in k^{th} stage fermentor. b_k and $s_{f,k}$ are bleed ratio and substrate concentration in feed (kg/m^3) for k^{th} stage. $q_{Mi,k}$ and $q_{Mo,k}$ are respectively inlet and outlet flow rates (m^3/h) on sweep fluid side of k^{th} stage pervaporation unit. $p_{Mi,k}$ and $p_{Mo,k}$ are respectively inlet and outlet ethanol concentrations (kg/m^3) on the sweep fluid side of k^{th} stage pervaporation unit. $V_{M.k}$ is the volume (m^3) of k^{th} stage pervaporation unit (on the sweep fluid side) while V_F is the volume (m^3) of each fermentor.

λ is the mass fraction of glucose in substrate (i.e., glucose and xylose) in the feed and ρ is the density (kg/m^3) of ethanol. ζ_x, ζ_s and ζ_p are cell discard/separation factor, substrate condensed factor and ethanol condensed factor respectively ($\zeta_x = x_{e,1}/x_1$, $\zeta_s = s_{ge,1}/s_{g,1} = s_{xe,1}/s_{x,1}$ and $\zeta_p = p_{e,1}/p_1$; see Figure 7.1). These factors decide the degree of separation of cell mass, substrate and product in the cell separator; this study uses values of these factors reported in Chen and Wang (2010b). a is the permeation coefficient (i.e., $a = A_M P$ where A_M m^2 is the membrane surface area and P m/h is the membrane permeability). The rate expressions for cell mass growth ($r_{x,k}$), glucose consumption ($r_{sg,k}$), xylose conversion ($r_{sx,k}$) and ethanol production ($r_{p,k}$) are presented in Chapter 6 (Section 6.8.2). The kinetic parameters used are taken from Krishnan et al. (1999), and these are also presented in Table 6.5. Similar to Chen and Wang (2010b), it is assumed that the feed contains 65% glucose and the remaining is

xylose (i.e., $\lambda = 0.65$). In this work, $P_{Mi,k} = 0$ kg/m^3, P = 0.1283 m/h, $\zeta_x = 0.01$ and $\zeta_s = \zeta_p = 1.01$.

Chen and Wang (2010b) have presented a model for three-stage fermentation integrated with cell recycling and inter-stage extraction (i.e., extractive fermentation). In this integrated process, a mixture of glucose and xylose is converted into ethanol. Chapter 6 discusses extractive fermentation process and its model (see Section 6.8.2). The process model for extractive fermentation process can be easily reduced to process model for three-stage fermentation process integrated with cell recycling only; this can be done by putting extraction efficiency equal to zero for each stage of extractive fermentation.

7.3 Multi-objective Optimization Problem Formulation

7.3.1 Three-stage Fermentation Process Integrated with Cell Recycling

In this MOO study, ethanol productivity and xylose conversion are used as the objectives, for all cases. As mentioned earlier, ethanol productivity and substrate conversion ensure efficient utilization of production capacity and substrate respectively. As glucose conversion is higher than xylose conversion, it is not considered as an objective. Ethanol productivity is defined as ethanol production rate per unit volume of all fermentors [kg/(m^3.h)]. Table 7.1 summarizes the MOO problem for this case; decision variables for MOO are dilution rate (D) for the first fermentor, substrate concentration in feed ($s_{f,1}$), bleed ratios (b_k, cell mass recycling) for all fermentors.

Table 7.1: MOO problem formulation for three-stage fermentation integrated with cell recycling

Objective functions	Ethanol productivity, $\frac{D}{3}\left[b_3 + (1 - b_3)\zeta_p\right] p_3$ Xylose conversion, $1 - \frac{[b_3 + (1-b_3)\zeta_s]s_{x,3}}{(1-\lambda)s_{f,1}}$
Decision variables	$0.3 \leq D \leq 0.8$ [1/h] $40 \leq s_{f,1} \leq 130$ [kg/m^3] $0.1 \leq b_k \leq 0.2,\ k = 1, 2, 3$
Constraints	Same as defined in Table 6.6 (Chapter 6)

7.3.2 Three-stage Fermentation Process Integrated with Cell Recycling and Pervaporation

In this case, Table 7.2 provides details of the MOO problem. The decision variables for MOO are dilution rate ($D_{1,i}$) for the first fermentor, substrate concentration in feed ($s_{f,1}$), bleed ratios (b_k, cell mass recycling) for all fermentors, number of pervaporation units used with each fermentor (NM_k), and inlet sweep fluid flow rates ($q_{Mi,k}$). In addition to mass balances (Equations 7.1b to 7.1p), positive values of ethanol productivity of each stage (π_k), glucose and xylose conversions ($\chi_{g,k}$ and $\chi_{x,k}$) in each stage are physical constraints. Other constraints are total sugar supply (s_T) and limits on the residual glucose and xylose concentrations ($s_{g,3}$ and $s_{x,3}$) in the mother liquor from the third fermentor. Residual glucose and xylose concentrations higher than a desired limit cause fouling inside the distillation column.

Table 7.2: MOO problem formulation for three-stage fermentation process integrated with cell recycling and pervaporation

Objective functions	Ethanol productivity, $\frac{D_{3,o}}{3}\left[b_3 + (1-b_3)\zeta_p\right]p_3 + \frac{[q_{Mo,1}P_{Mo,1} + q_{Mo,2}P_{Mo,2} + q_{Mo,3}P_{Mo,3}]}{3V_F}$
	Xylose conversion, $1 - \frac{[b_3 + (1-b_3)\zeta_s]s_{x,3}}{(1-\lambda)s_{f,1}}$
Decision variables	$0.3 \le D_{1,i} \le 0.8$ [1/h]
	$40 \le s_{f,1} \le 130$ [kg/m^3]
	$0.1 \le b_k \le 0.2$
	$10 \le NM_k \le 35$
	$50 \le q_{Mi,k} \le 160$ [m^3/h] for k = 1, 2 and 3
Constraints	$\pi_1 = D_{1,o}\left[b_1 + (1-b_1)\zeta_p\right]p_1 \ge 0$
	$\pi_2 = D_{2,o}\left[b_2 + (1-b_2)\zeta_p\right]p_2 - D_{1,o}\left[b_1 + (1-b_1)\zeta_p\right]p_1 \ge 0$
	$\pi_3 = D_{3,o}\left[b_3 + (1-b_3)\zeta_p\right]p_3 - D_{2,o}\left[b_2 + (1-b_2)\zeta_p\right]p_2$
	$\chi_{g,1} = 1 - \frac{[b_1 + (1-b_1)\zeta_s]s_{g,1}}{\lambda s_{f,1}} \ge 0; \quad \chi_{x,1} = 1 - \frac{[b_1 + (1-b_1)\zeta_s]s_{x,1}}{(1-\lambda)s_{f,1}} \ge 0$
	$\chi_{g,2} = 1 - \frac{[b_2 + (1-b_2)\zeta_s]s_{g,2}}{[b_1 + (1-b_1)\zeta_s]s_{g,1}} \ge 0; \quad \chi_{x,2} = 1 - \frac{[b_2 + (1-b_2)\zeta_s]s_{x,2}}{[b_1 + (1-b_1)\zeta_s]s_{x,1}} \ge 0$
	$\chi_{g,3} = 1 - \frac{[b_3 + (1-b_3)\zeta_s]s_{g,3}}{[b_2 + (1-b_2)\zeta_s]s_{g,2}} \ge 0; \quad \chi_{x,3} = 1 - \frac{[b_3 + (1-b_3)\zeta_s]s_{x,3}}{[b_2 + (1-b_2)\zeta_s]s_{x,2}} \ge 0$
	$s_{g,3} < 0.1$ kg/m^3
	$s_{x,3} < 1$ kg/m^3
	$D_{1,i}\, s_{f,1}/3 < s_T$ kg/(m^3.h)
	$1 - \frac{[b_3 + (1-b_3)\zeta_s]s_{g,3}}{\lambda s_{f,1}} > 0.99$
	Mass balances (Equations 7.1b to 7.1p)

7.3.3 Three-stage Fermentation Process Integrated with Cell Recycling and Extraction

The MOO problem formulation for three-stage fermentation process integrated with cell recycling and inter-stage extraction is given in Table 7.3. In this case, decision variables and their ranges are same as in three-stage fermentation process integrated with cell recycling only.

Table 7.3: MOO problem formulation for three-stage fermentation process integrated with cell recycling and inter-stage extraction

Objective functions	Ethanol productivity, $\frac{D}{3}\left[b_3 + (1-b_3)\zeta_p\right] p_3 + \sum_{k=1}^{2}(1-b_k)\frac{\zeta_p E}{(1+E)} p_k$ Xylose conversion, $1 - \frac{[b_3 + (1-b_3)\zeta_s] s_{x,3}}{(1-\lambda) s_{f,1}}$
Decision variables	$0.3 \leq D \leq 0.8$ [1/h] $40 \leq s_{f,1} \leq 130$ [kg/m^3] $0.1 \leq b_k \leq 0.2,\ k = 1, 2, 3$
Constraints	Same as defined in Table 6.6 (Chapter 6)

7.4 Multi-objective Differential Evolution

Chen and Wang (2010b) have optimized the multi-stage fermentation process with cell recycling and extraction, using fuzzy approach, which requires selection of fuzzy boundaries for different objectives and constraints. Finally, they have solved fuzzy MOO problem as a single objective optimization problem using hybrid differential evolution (HDE) that requires additional knowledge about the relative importance of objectives. On the other hand, many MOO generating techniques have been developed and widely applied in Chemical Engineering and related areas

(Masuduzzaman and Rangaiah, 2009). These techniques generate the Pareto-optimal front in a single run, and do not require any additional information about objective space or problem solution. Hence, it will be better and desirable to use one of these techniques for optimizing the multi-stage integrated fermentation process.

In the present work, a multi-objective differential evolution (MODE) algorithm is used to solve different MOO problems. This algorithm is similar to GDE3 (Kukkonen and Lampinen, 2007). In this work, crowding distance is used to calculate the crowdedness of non-dominated solutions. Further, the performance of GDE3 and NSGA2_SBX algorithms are better compared to six other MOO algorithms on two-, three- and five-objective test functions (see Table 2.2 for performance of different multi-objective DE algorithms). MODE algorithm has been implemented using Visual Basic Application in MS-Excel worksheets. Objectives and constraints are calculated using Excel worksheet and other features and functions available in Excel. A flowchart of the MODE algorithm is presented in Figure 7.2. Inequality constraints are handled by constrained-dominance (also known as feasibility) approach of Deb et al. (2002). The search procedure is repeated for the specified maximum number of generations (MNG).

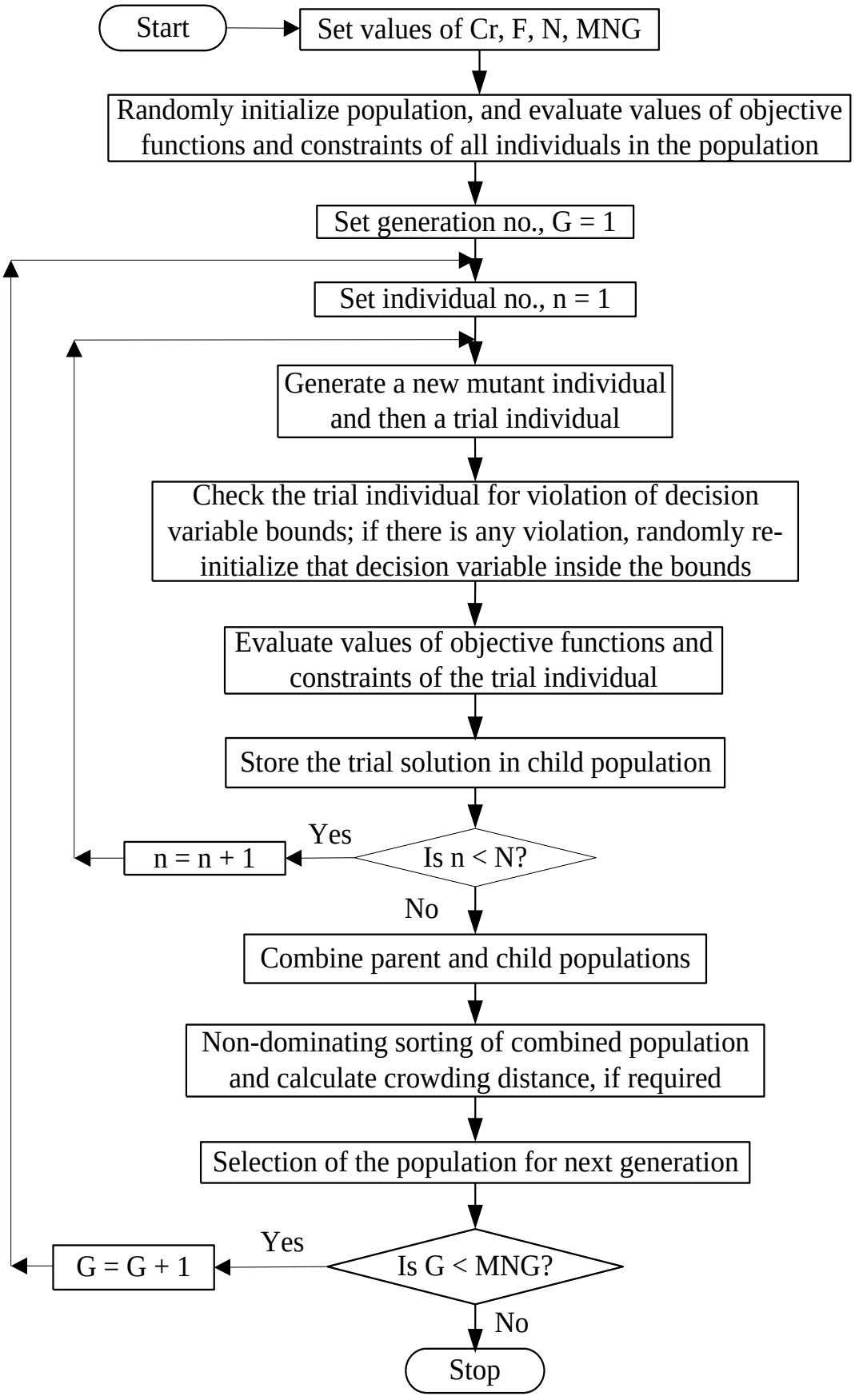


Figure 7.2: Flowchart of the MODE algorithm

7.5 Optimization Results

7.5.1 MODE Algorithm with MNG

The values of MODE parameters used in this study are: population size = 100 (200 for pervaporation case), crossover probability (Cr) = 0.3, mutation rate (F) = 0.5 and maximum numbers of generations (MNG) = 200. These values are selected based on the preliminary testing. The steady state model (equations 7.1b to 7.1p) of the process has to be solved numerically to calculate the values of objective functions and constraints for each individual (decision variable vector) provided by the MODE algorithm. In this work, Solver in MS-Excel has been used to solve the steady state model equations by minimizing sum of squares. The computational time taken to optimize the three-stage fermentation process integrated with cell recycling and pervaporation is about 4.5 hours on an Intel Core i5 Computer (Dual CPU of 2.4 GHz each, RAM 8 GB, 64-bits). MFlop/s (million floating point operations per second) on this computer is 594 for the LINPACK benchmark program for a matrix of order 500 (http://www.netlib.org/benchmark/linpackjava/).

7.5.1.1 Fermentation with Cell Recycling

Initially, a multi-stage fermentation process with cell recycling but without extraction or pervaporation unit (base case) has been optimized for ethanol productivity and xylose conversion simultaneously. Figure 7.3 shows the optimization results obtained for the three-stage fermentation process coupled with cell recycling only. In this case, dilution rate for first fermentor, substrate concentration in feed and bleed ratios for different stages are the decision variables (see Table 7.1). The obtained Pareto-optimal front for the ‘base case’ is well distributed, which means MODE performs well and is successful for this problem. As expected, ethanol

productivity and xylose conversion are conflicting in nature (i.e., an increase in ethanol productivity is accompanied by decrease in xylose conversion). The obtained Pareto-optimal front in Figure 7.3(a) can be divided into two parts: (i) a steep increase in ethanol productivity (2.6-5.5 kg/(m^3.h)) with little decrease in xylose conversion, and (ii) a linear change between ethanol productivity (5.5-7 kg/(m^3.h)) and xylose conversion. The fast improvement in ethanol productivity in the first part is due to change in substrate concentration in feed and then dilution rate (Figures 7.3b and 7.3c). In the second part, mainly dilution rate is affecting both the objectives. All the bleed ratios are nearly constant at their lower bounds (i.e., 0.1), and so these are not shown in Figure 7.3 for brevity.

In general, effect of decision variables on the objectives is non-linear for application problems, and decision variables may also have complicated interaction. In the 'base case' optimization, substrate concentration is first increasing and then decreasing (see Figure 7.3c). Hence, a non-dominated solution, shown as "▲" in Figure 7.3, is selected for additional analysis. Now, value of substrate concentration for this solution is replaced by 85 kg/m^3 (see point "×" in Figure 7.3c); the obtained solution corresponding to the modified decision variable vector is shown as "×" in Figure 7.3(a). The newly obtained solution "×" is non-dominated with "▲" solution; but it is dominated by several other non-dominated solutions on the Pareto-optimal front (see Figure 7.3a). In conclusion, fast increase in ethanol productivity can be achieved by increase in substrate concentration, until un-reacted substrate becomes large. Then, dilution rate increases to increase ethanol productivity while substrate concentration decreases to achieve higher xylose conversion.

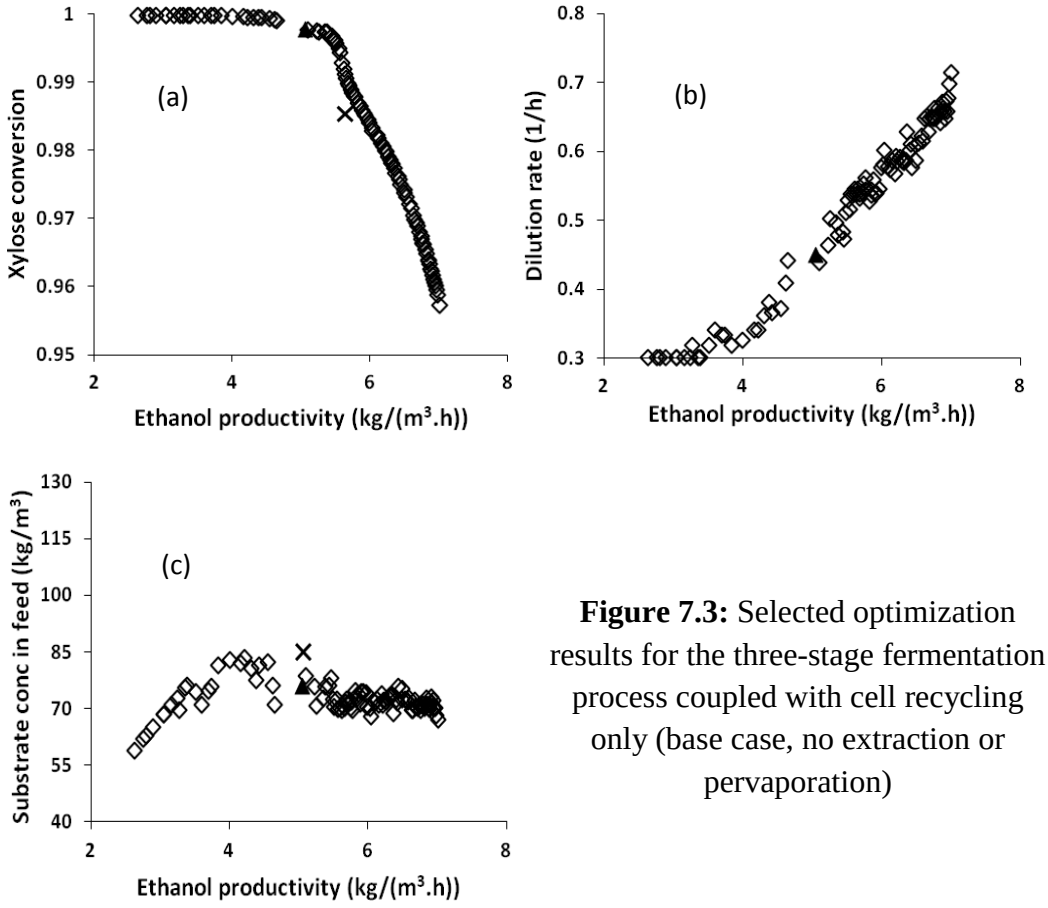


Figure 7.3: Selected optimization results for the three-stage fermentation process coupled with cell recycling only (base case, no extraction or pervaporation)

7.5.1.2 Fermentation with Pervaporation

Pareto-optimal front obtained for the three-stage fermentation process integrated with cell recycling and pervaporation is shown in Figure 7.4(a) (pervaporation case). In this case, ethanol productivity has improved significantly compared to the 'base case'; maximum value of productivity has increased from 7 to 13.3 kg/(m^3.h). Further, xylose conversion has also improved, and it is close to one. Figure 7.4(a) also presents Pareto-optimal front obtained after 150 generations; both Pareto-optimal fronts (i.e., after 150 and 200 generations) are practically comparable which confirm global optimality of the obtained solutions.

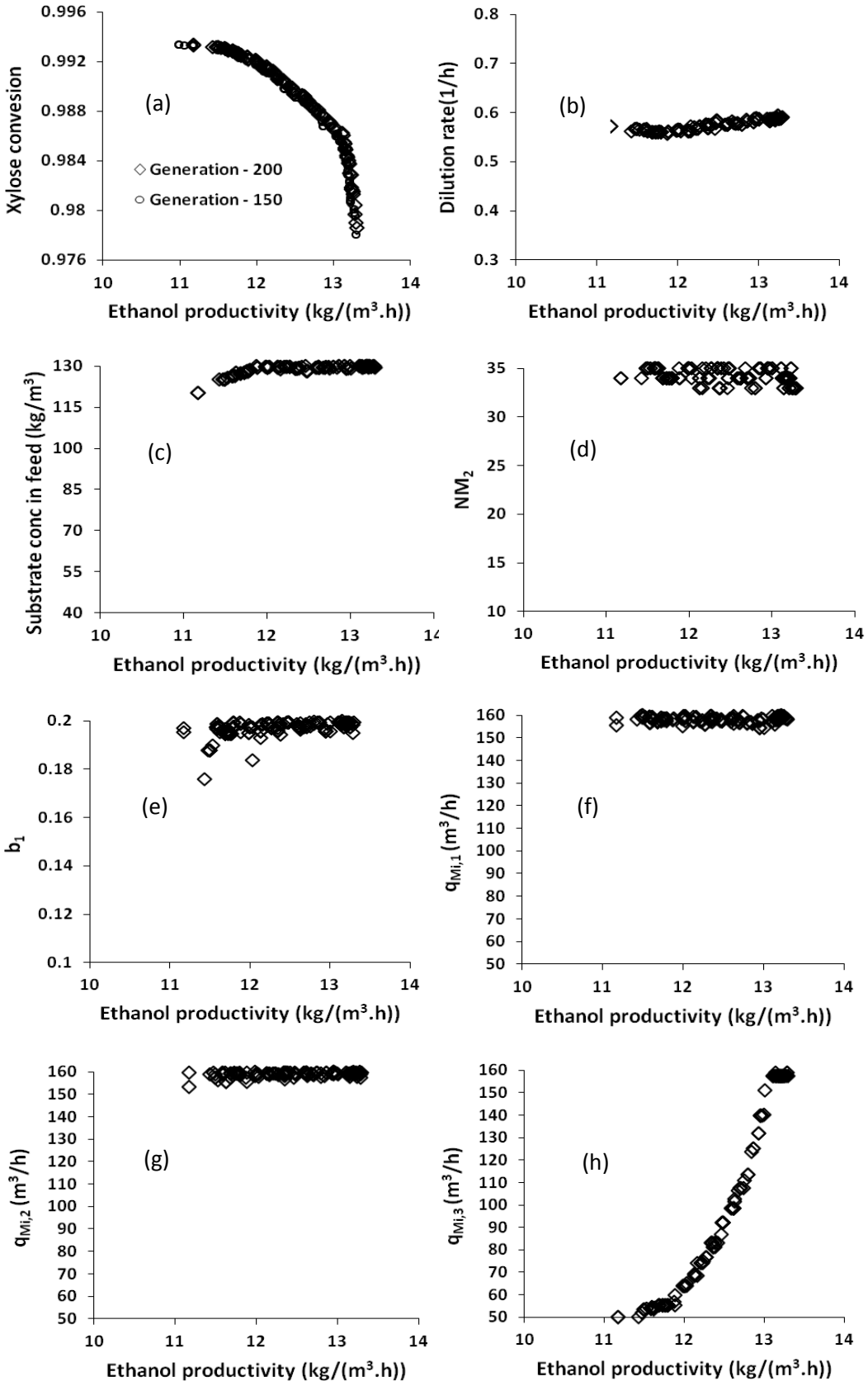


Figure 7.4: Selected optimization results for the three-stage fermentation process integrated with cell recycling and pervaporation (pervaporation case)

Variations in dilution rate, substrate concentration in feed and inlet sweep fluid flow rate for 3rd stage (see Figures 7.4b, 7.4c and 7.4h) are mainly contributing to the variation in objectives (Figure 7.4a). Both dilution rate ($D_{1,i}$) and substrate concentration in feed ($s_{f,1}$) are varying over small range; $D_{1,i}$ is varying between 0.5 to 0.6, and $s_{f,1}$ is near to its upper bound. Number of pervaporation units used with the first and third fermentors (NM_1 and NM_3) is near to its upper and lower bound respectively (i.e., 35 and 10), while NM_2 is scattered near its upper bound (i.e., 35 in Figure 7.4d). Large amount of feed is converted to ethanol in first fermentor, and so high number of pervaporation units (i.e., NM_1) is required to reduce the product inhibition in first stage. In the third fermentor, lesser amount of feed is converted to ethanol, and so fewer pervaporation units are required in the third stage. Bleed ratio for the first fermentor (b_1) is scattered near its upper bound (i.e., 0.2 in Figure 7.4e) while bleed ratio for second and third fermentors (b_2 and b_3) is close to its lower bound (i.e., 0.1). In this case, bleed ratio for the first stage is higher compared to 'base case' (i.e., $b_1 = 0.1$). Here, high bleed ratio provides larger amount of un-reacted feed to the second stage fermentor, which has better performance due to the presence of pervaporation units, compared to the 'base case'. For brevity, variations in variables NM_1, NM_3, b_2 and b_3 are not presented in Figure 7.4. Sweep fluid flow rate for the first two stages ($q_{Mi,1}$ and $q_{Mi,2}$) is nearly constant at its upper bound (i.e., 160 m^3/h), while $q_{Mi,3}$ is varying between 50 to 160 kg/m^3 (Figures 7.4f-h).

7.5.1.3 Fermentation with Inter-stage Extraction

Detailed mathematical model for the three-stage fermentation process integrated with cell recycling and inter-stage extraction (extraction case) is presented in Chen and Wang (2010b). 2-ethyl-1-hexanol (boiling point of 184.6 ^0C) is used as the

solvent for ethanol extraction from fermentor broth (Offeman et al., 2005). It is assumed that the solvent can recover 87.4% of ethanol present in the feed (Chen and Wang, 2010b). In this case, decision variables and their ranges are same as in the 'base case'. In the extraction case, dilution rates for all the stages are fixed (similar to Chen and Wang, 2010b), and so volume of the 2nd and 3rd fermentors will vary due to fermentation and consequent density variation. Volume of the first fermentor is 257.4 m^3, same as in 'pervaporation case'.

Pareto-optimal front obtained for 'extraction case' using MODE algorithm is shown in Figure 7.5(a). Addition of inter-stage extraction improves performance of the three-stage fermentation process for both ethanol productivity and xylose conversion. The maximum value of ethanol productivity (~13.46 kg/(m^3.h)) is nearly equal to that of pervaporation case, but the ranges for both the objectives are wider compared to 'pervaporation case' (Figures 7.5a and 7.4a). In the optimization of extraction case, optimal values of dilution rate and substrate concentration in feed vary with the objectives (Figures 7.5b and 7.5c). On the other hand, all bleed ratios are nearly constant at their lower bounds (i.e., 0.1).

The obtained Pareto-optimal front in the extraction case shows some discontinuity near ethanol productivity of 5 kg/(m^3.h). In order to ensure the global optimality of the obtained solution, 'extraction case' has been run several times with the bleed ratios fixed at their lower bounds (as these are nearly constant); practically, there is no difference between solutions obtained in different runs. This indicates that MODE algorithm has converged to the global Pareto-optimal front. Further, a decision variable vector (shown as "×" in Figures 7.5b and 7.5c) is chosen between the decision variable vectors corresponding to two non-dominated solutions at the ends of the discontinuity (shown as "▲" in Figure 7.5); the obtained solution for this set of

decision variables vector (shown as "×" in Figure 7.5a) is very slightly dominated by one of the solutions at the discontinuity and five other nearby non-dominated solutions in the Pareto-optimal front (e.g., ethanol productivity and xylose conversion at point "×" are respectively 5.478 and 0.9995 compared to 5.787 and 0.9996 for one dominating solution). Several other combinations of decision variables are also tried, and it is found that the corresponding solutions for most of them are very slightly dominated by several non-dominated solutions on the Pareto-optimal front. So, the discontinuity in Figure 7.5 is probably due to the definition of dominance, accuracy of optimization calculations, and complex relationships among objectives and decision variables.

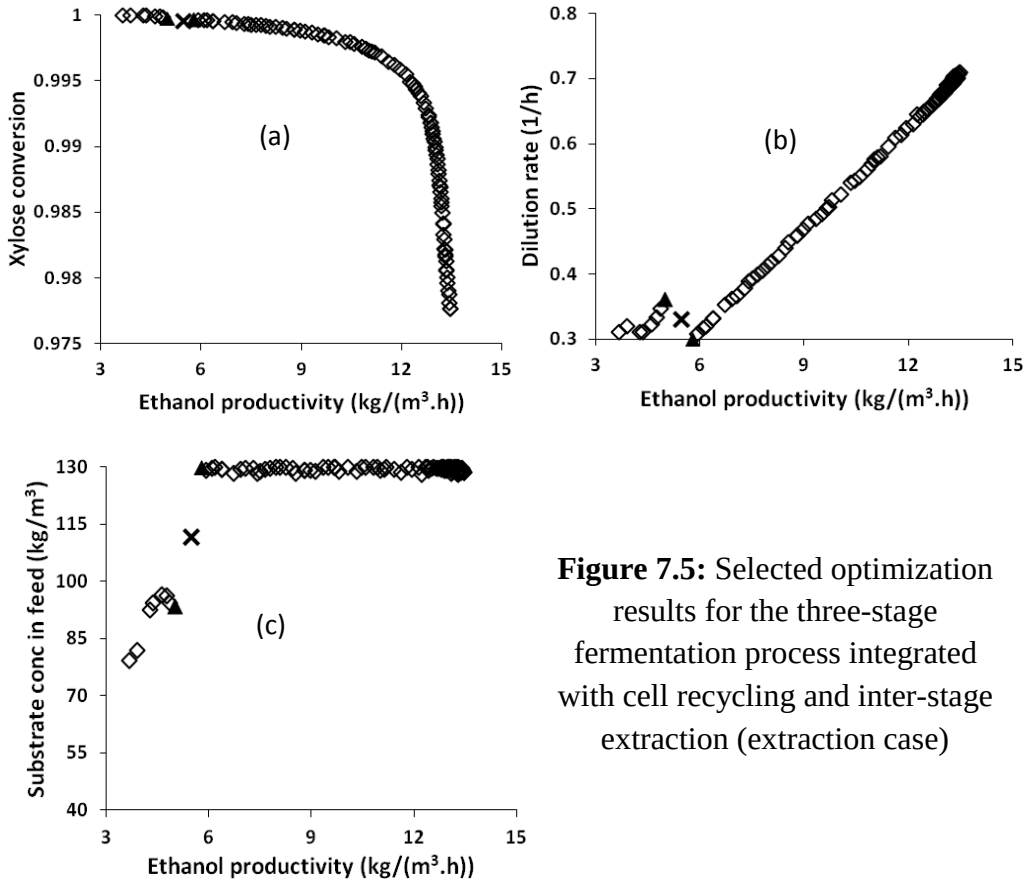


Figure 7.5: Selected optimization results for the three-stage fermentation process integrated with cell recycling and inter-stage extraction (extraction case)

7.5.2 Use of I-MODE Algorithm

In the previous sub-section, MODE (with MNG) is used to obtain the Pareto-optimal fronts for different cases. Here, I-MODE is used to solve the above three optimization problems. It has improvement-based termination criterion and taboo list (see Chapter 3 for more details). In this study, values of termination parameters (δ_{GD} = 0.0003 & δ_{SP} = 0.1) and taboo list (= 0.01) are based on the recommendation in Chapter 3. The obtained Pareto fronts after termination generation are compared against the Pareto fronts obtained in above sub-section (MODE and MNG = 200). In I-MODE, DE parameters are self-adapted, whereas population size of 100 is used for all problems. To avoid indefinite looping, termination criterion based on the MNG (= 200) is also added in I-MODE.

Figure 7.6(a) shows non-dominated solutions obtained for three-stage fermentation process integrated with cell recycling; here, both productivity and xylose conversion are simultaneously maximized. It can be seen that termination criterion is successfully terminating the search near to the solution; non-dominated solutions obtained after GT (= 57) are closer to those obtained in the previous sub-section. Thus, termination criterion avoids unnecessary computations with little compromise in the solution quality. In the case of three-stage fermentation process integrated with cell recycling and extraction or pervaporation, same objectives are considered. Figures 7.6(b) and 7.6(c) show non-dominated solutions obtained for these two fermentation processes. In both the cases, termination criterion is successful in terminating I-MODE algorithm at the right generations. The non-dominated solutions obtained after termination generations (GT = 186 and 67) are near to the solutions presented in the previous sub-section (MODE with MNG = 200), and these are also well distributed along the Pareto-optimal fronts. In the case of extractive

fermentation, termination criterion saves numerous generations (GT = 67), whereas, for fermentation with pervaporation, search terminates (GT = 186) just before MNG (= 200).

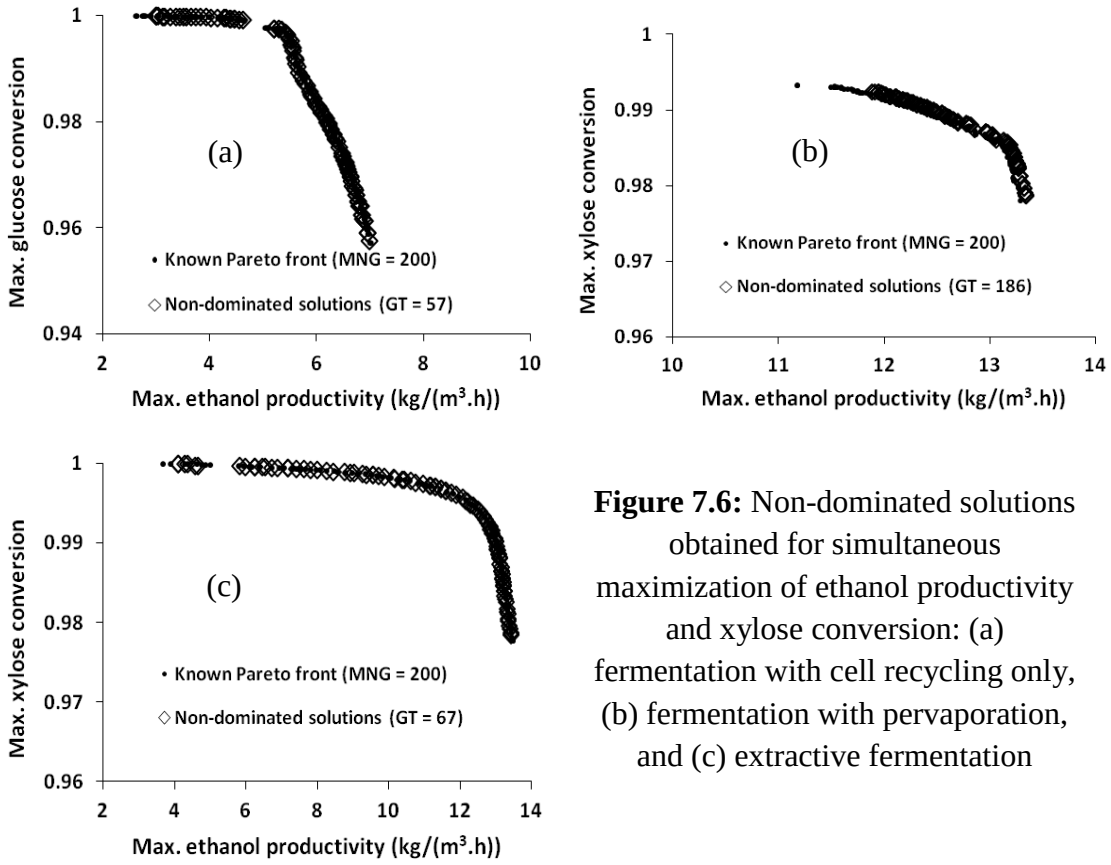


Figure 7.6: Non-dominated solutions obtained for simultaneous maximization of ethanol productivity and xylose conversion: (a) fermentation with cell recycling only, (b) fermentation with pervaporation, and (c) extractive fermentation

7.6 Comparison of Extraction and Pervaporation for the Three-stage Fermentation Process

For 'pervaporation case', optimal value of dilution rate is changing between 0.5 and 0.6 (Figure 7.4b). On the other hand, for extraction case, it generally increases with ethanol productivity and reaches to a maximum of 0.71 (Figure 7.5b). Substrate concentration in feed for 'pervaporation case' is varying over a small range near to its upper bound (Figure 7.4c). On the other hand, for 'extraction case', substrate concentration in feed is near to its upper bound except for a few solutions at low

productivity having $s_{f,1}$ between 70 to 100 kg/m^3 (Figure 7.5c). In both the cases, xylose conversions are comparable (see Figures 7.4a and 7.5a).

It is difficult to compare the performance of both pervaporation and extraction cases by seeing the obtained Pareto-optimal fronts. So, a detailed comparison between product streams of both arrangements (i.e., extraction and pervaporation cases) has been made for one optimal solution; this includes comparison of flow rates and ethanol concentrations. A solution with maximum productivity (i.e., 13.29 kg/(m^3.h)) has been picked from the 'pervaporation case' for comparison with a solution of the same productivity from the 'extraction case' (see Table 7.4). The optimum temperature for fermentation is 30-35 ^{0}C (Krishnan et al., 1999) and cell separators operate at 42 ^{0}C (Chen and Wang, 2010b). Amount of solvent (2-ethyl-1-hexanol) used in the inter-stage extraction, for the selected optimal solution, is calculated using liquid-liquid extraction (LLE) column in Aspen Hysys V-7.2; NRTL thermodynamic model has been used for this system.

Distillation is often required to recover high purity ethanol from solvent, and is highly energy intensive. Pervaporation process is more energy efficient compared to distillation for high purity ethanol recovery (Bek-Pedersen et al., 2000). In both the cases, part of ethanol is recovered from fermentation broth (i.e., ethanol-water mixture from Stage-F3). From Table 7.4, it can be seen that Stage-F3 flow rate for the 'extraction case' is larger (i.e., 161.8 m^3/h) compared to the 'pervaporation case' (i.e., 146.9 m^3/h), and ethanol concentration is lower in the Stage-F3 product stream of the former case (extraction - 10.8 kg/m^3, pervaporation - 31.4 kg/m^3). Hence, ethanol recovery from the fermentation broth of 'extraction case' is more expensive compared to 'pervaporation case'.

Table 7.4: Values of flow rates and ethanol concentration for different product streams in extraction and pervaporation cases; ***** - these are not the ethanol product streams.

(A) Extraction case: ethanol productivity = 13.298 kg/(m^3.h) & xylose conversion = 0.978

($D_{1,i}$ = 0.69 h^{-1}, $s_{f,1}$ = 129.88 kg/m^3, $b_1 = b_2 = b_3$ = 0.1)

Extraction section	**Stage-E1**	**Stage-E2**	
Ethanol removed by inter-stage extraction (kg/h)	5280.4	2920.8	
Ethanol conc. in solvent at extractor outlet (kg/m^3)			
- LLE column with single stage	8.7	4.9	
- LLE column with 5 stages	52.5	29.5	
- LLE column with 10 stages	63.4	35.4	
- LLE column with 15 stages	66.7	37.1	
Fermentation section	**Stage-F1***	**Stage-F2***	**Stage-F3**
Mother liquor flow rate at fermentor inlet (m^3/h)	-	168.1	162.7
Mother liquor flow rate at fermentor outlet (m^3/h)	174.8	166.3	161.8
Ethanol conc. in mother liquor at fermentor outlet (kg/m^3)	37.3	21.4	10.8

(B) Pervaporation case: ethanol productivity = 13.295 kg/(m^3.h) & xylose conversion = 0.979

($D_{1,i}$ = 0.59 h^{-1}, $s_{f,1}$ = 129.55 kg/m^3, b_1 = 0.199, $b_2 = b_3$ = 0.1, $q_{Mi,1}$ = 158.18 m^3/h, $q_{Mi,2}$ = 159.56 m^3/h, $q_{Mi,3}$ = 157.48 m^3/h, NM_1 = 35, NM_2 = 33 and NM_3 = 10)

Pervaporation section	**Stage-P1**	**Stage-P2**	**Stage-P3**
Sweep fluid flow rate at outlet (m^3/h)	160.3	161.9	158.2
Ethanol conc. in sweep fluid at outlet (kg/m^3)	10.3	12.0	12.7
Fermentation section	**Stage-F1***	**Stage-F2***	**Stage-F3**
Mother liquor flow rate at fermentor outlet (m^3/h)	149.9	147.6	146.9
Ethanol conc. in mother liquor at fermentor outlet (kg/m^3)	25.7	30.1	31.4

In ‘extraction case’, remaining ethanol (i.e., from Stage-E1 and Stage-E2) has to be recovered from solvent. Amount of solvent and concentration of ethanol in the extract stream depends on the number of stages used in the LLE column (Table 7.4). As concentrations of ethanol in solvent streams with single-stage LLE columns are small (i.e., 8.7 and 4.9 kg/m^3), ethanol recovery from the solvent will be expensive. Hence, multi-stage LLE is better for extracting ethanol from fermentation broths in first and second stages. In ‘pervaporation case’, ethanol is recovered continuously from sweep fluid leaving from Stage-P1, Stage-P2 and Stage-P3 using a cold surface; purity of recovered ethanol will be very high in this case. In ‘extraction case’, ethanol is not removed continuously from the fermentor; rather it is removed after fermentation.

Finally, for the same value of productivity (i.e., 13.29 kg/(m^3.h)), the ‘extraction case’ requires higher feed flow rate (i.e., dilution rate = 0.69) compared to the ‘pervaporation case’ (i.e., dilution rate = 0.59). Substrate concentration in feed is comparable in both the cases (pervaporation: 129.88 kg/m^3, extraction: 129.55 kg/m^3). Hence, the ‘pervaporation case’ is better than the ‘extraction case’ as it requires less amount of feed with nearly same substrate concentrations in feed for the same ethanol productivity. Further, cost of recovery and purification of ethanol is expected be cheaper for the ‘pervaporation case’.

7.7 Ranking of Non-dominated Solutions obtained for Fermentation with Pervaporation

The selection of one or two solutions from the obtained Pareto-optimal front requires additional knowledge about the process (i.e., preference of decision maker). Sometimes, the Pareto-optimal front has sudden change in one of the objective, while remaining objectives have small variations; this type of trend can be used to select a

preferred solution. If the Pareto-optimal front is smooth, then net flow method (NFM) or rough set method can be used to rank the set of non-dominated solutions (Thibault, 2009). Here, the former is chosen to rank the non-dominated solutions obtained for three-stage fermentation process integrated with cell recycling and pervaporation (Figure 7.4a).

In NFM, decision maker decides relative importance or weight of each objectives (w_m; $\sum w_m = 1$), and also the indifference (Q_m), preference (P_m) and veto (V_m) thresholds (see Table 7.5). Q_m is the range of each objective for which the decision maker is not able to decide the favorite solution from a pair of non-dominated solutions. If the difference between two solutions for a given objective exceeds P_m, the decision maker should be able to select one solution with the better objective value. Finally, if difference between two solutions for a particular objective is more than V_m, then solution with worse objective should be banned. The three threshold values for m^{th} objective should be such that: $0 \leq Q_m \leq P_m \leq V_m$. In NFM, concordance and discordance indices are calculated for each pair of non-dominated solutions, which are derived from user-supplied weights and three thresholds. After this, relative performance of each pair in the Pareto-optimal front is calculated (Thibault, 2009). Finally, the following equation is used to calculate the score for different non-dominated solutions.

Final ranking score = performance of i^{th} individual relative to remaining individuals - performance of remaining individuals relative to i^{th} individual (7.2)

Extreme values and range of each objective are presented in Table 7.5; Q_m, P_m, and V_m values are respectively 10%, 20% and 80% of objective range (Lee and Rangaiah, 2009). Figure 7.7 presents best 20 non-dominated solutions from the

solution set for three different weight combinations. Preferred (or top ranked) non-dominated solutions, using different weights, do not cover the entire range of the Pareto-optimal front, rather they are concentrated in different regions. The obtained Pareto-optimal front can be divided into three parts (see dashed line in Figure 7.7 for partition); the preferred non-dominated solutions using different weights are near to the edges of Pareto-optimal front (or both ends of middle part). As expected, non-dominated solutions with 0.25 and 0.75 weights for ethanol productivity are more in the left and right parts, respectively. Preferred non-dominated solutions are near to the ends of middle part, for equal weights for both objectives (i.e., w_m = 0.5 each). Sometimes, some non-dominated solutions may have same rank using two different sets of weights, as can be seen in Table 7.6 which presents top 10 non-dominated solutions for three different weight combinations. Finally, based on the relative importance and the trade-off between the two objectives, the top ranked solution can be chosen by the decision maker for implementation.

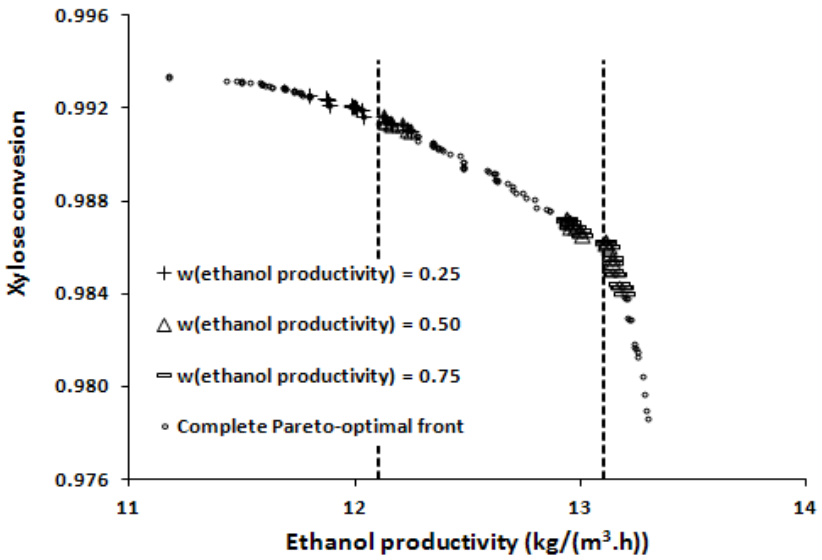


Figure 7.7: Ranking of Pareto-optimal front using NFM for pervaporation case; top 20 non-dominated solutions with three sets of weights are presented

Table 7.5: Parameters in the net flow method for ranking non-dominated solutions obtained in the pervaporation case (Figure 7.4a)

	Ethanol productivity	**Xylose conversion**
Goal	Max	Max
Minimum value	11.171	0.9786
Maximum value	13.295	0.9934
Objective range	2.124	0.0148
Weights (w_m)	0.25, 0.5 or 0.75	0.75, 0.5 or 0.25
Indifference threshold (Q_m)	0.21	0.0015
Preference threshold (P_m)	0.42	0.0030
Veto threshold (V_m)	1.7	0.0118

Table 7.6: Top 10 non-dominated solutions for pervaporation case using NFM; (EP - ethanol productivity in kg/(m^3.h) and XC - xylose conversion)

Rank	$w_{\text{ethanol productivity}} = 0.25$		$w_{\text{ethanol productivity}} = 0.5$		$w_{\text{ethanol productivity}} = 0.75$	
	EP	XC	EP	XC	EP	XC
1	12.128	0.992	**13.108**	**0.986**	**13.108**	**0.986**
2	12.158	0.991	**13.109**	**0.986**	**13.109**	**0.986**
3	12.133	0.991	**13.124**	**0.986**	**13.124**	**0.986**
4	11.985	0.992	13.000	0.987	13.136	0.986
5	12.130	0.991	12.951	0.987	13.142	0.985
6	12.028	0.992	12.970	0.987	13.000	0.987
7	**12.208**	**0.991**	**12.208**	**0.991**	13.005	0.987
8	12.002	0.992	12.933	0.987	13.142	0.985
9	11.989	0.992	12.933	0.987	13.155	0.985
10	12.002	0.992	12.128	0.992	12.970	0.987

7.8 Conclusions

This study optimizes a fermentation process with in-situ ethanol removal and cell recycling, and using glucose and xylose as the feedstock. For this, a three-stage fermentation process integrated with cell recycling and pervaporation, has been modeled and optimized for ethanol productivity and xylose conversion simultaneously. The MODE algorithm, implemented in MS-Excel environment, is used to generate the Pareto-optimal front. The obtained Pareto-optimal fronts give better insights and provide many alternatives with different ethanol productivity and substrate conversion. The obtained Pareto-optimal front for the three-stage fermentation integrated with cell recycling and pervaporation is well distributed; dilution rate for the first stage, substrate concentration in feed and sweep fluid flow rate for third stage mainly contribute to the Pareto-optimal front. High value of bleed ratio for first stage and low values of bleed ratios for last two stages are favorable for both ethanol productivity and xylose conversion.

The performance of extraction and pervaporation units with a three-stage fermentation process coupled with cell recycling has been compared for ethanol productivity and xylose conversion. Addition of pervaporation unit with fermentors gives higher productivity and nearly same xylose conversion compared to extraction for the same amount of feed. Further, owing to ease of ethanol recovery, fermentation with pervaporation is better compared to fermentation with extraction. The improvement based termination criterion is able to terminate search (i.e., I-MODE) at the right generations. Finally, NFM is used to find the top 10 non-dominated solutions for the three-stage fermentation process integrated with cell recycling and pervaporation.

Chapter 8

Multi-objective Optimization of a Bio-diesel Production Process[6]

8.1 Introduction

Vegetable oils are mixtures of triglycerides, which can be converted into a mixture of fatty acid methyl esters (FAME) by reaction with methanol (i.e., trans-esterification). The product, FAME is the bio-diesel. The traditional bio-diesel processes use homogenous acid or alkali as catalyst for trans-esterification. Raw material cost is the major contributor to cost of bio-diesel. Waste cooking oils are cheaper than pure vegetable oils, but the former contain high amount of free fatty acid (FFA). The trans-esterification is more efficient and faster with an alkali catalyst compared to an acid catalyst, but it requires absence of FFA in the feed. FFA produces soap and water by reacting with an alkali catalyst; hence, alkali-catalyzed process cannot directly be used to produce bio-diesel from waste cooking oils. Canakci and Van Gerpen (2001) have combined the advantages of alkali- and acid-catalyzed processes into a single process, where waste vegetable oil is first treated in the presence of an acid catalyst (i.e., esterification), and later bio-diesel is produced from treated oil using an alkali catalyst (i.e., trans-esterification). Heterogeneous catalyzed and supercritical processes are other processes to produce bio-diesel (Freedman et al., 1986; Lotero et al., 2005). Heterogeneous catalyzed process does not produce aqueous waste since catalyst can be easily separated from the reaction mixture.

[6] This chapter is based on the article: Sharma, S. and Rangaiah, G. P. (2012), Multi-objective optimization of a bio-diesel production process, Fuel, 103, pp. 269-277.

Zhang et al. (2003a) proposed four different processes, namely, alkali-catalyzed process using pure vegetable oil, alkali-catalyzed process using waste cooking oil, acid-catalyzed process using waste cooking oil and acid-catalyzed process using hexane extraction, to produce bio-diesel. Their study also reports operational and design details for these process flow-sheets. In another work, Zhang et al. (2003b) conducted an economic evaluation of the above four processes, and found that acid-catalyzed process using waste cooking oil is more economical than others. Hass et al. (2006) developed a process model to estimate capital and operating costs of bio-diesel production using an alkali catalyst. West et al. (2008) analyzed four alternative processes: acid catalyzed, alkali catalyzed, heterogeneous acid catalyzed and supercritical, for producing bio-diesel from waste cooking oils, and also conducted economic analysis of these processes. Their study concluded that heterogeneous acid-catalyzed process is superior to the other three processes, but it is still in the development phase. Recently, Zhang et al. (2012) presented a process scheme for producing bio-diesel from pure vegetable oil, and developed a plant-wide control structure.

Several researchers have optimized the bio-diesel process for single objective. Ghadge and Raheman (2006) used response surface mythology to study the effect of methanol quantity, acid concentration and reaction time on FFA content reduction, and found that methanol quantity is the most prominent factor to reduce FFA content. Myint and El-Halwagi (2009) optimized four alternate alkali-catalyzed process flow-sheets, with different separation sequences, for bio-diesel production. They concluded that the best flow-sheet is with the bio-diesel and glycerol separation first, followed by methanol recovery and finally water washing. Nicola et al. (2010) optimized two process variants of alkaline trans-esterification of refined vegetable oil for two

objectives: minimum energy consumption and better product quality, simultaneously. This process was simulated using ASPEN Plus, and a multi-objective genetic algorithm was used to optimize it. However, optimization of bio-diesel production process using waste cooking oil has not been studied for multiple objectives. Waste cooking oils have significant impact on the environment if they are disposed without re-use, and so their use to produce bio-diesel is attractive for both economic and environmental reasons.

In this study, bio-diesel production process using waste cooking oil is simulated in Aspen Hysys; waste cooking oil is first treated in the presence of sulfuric acid (i.e., pre-treatment), followed by bio-diesel production from the treated oil using sodium hydroxide. Pre-treatment of waste cooking oil reduces FFA content below 1% by weight. The present study optimizes the design of bio-diesel process for three important objectives: maximum profit, minimum fixed capital investment and minimum organic waste. This chapter develops an economically attractive and environmentally acceptable bio-diesel production process using MOO approach. Effect of variation in waste cooking oil flow rate is also explored. Evolutionary multi-objective optimization techniques have been popular for studying the trade-offs between conflicting objectives in many Chemical Engineering applications (Masuduzzaman and Rangaiah, 2009). In this work, multi-objective differential evolution with taboo list (MODE-TL) is used to generate the Pareto-optimal front. Taboo list (TL) concept of taboo search has been incorporated with MODE to avoid revisiting the search space, which can reduce the number of objective function evaluations to obtain the global optimum (Srinivas and Rangaiah, 2007).

Next section of this article develops bio-diesel production process using waste cooking oil. Section 8.3 discusses process simulation, esterification and trans-

esterification kinetics, and thermodynamic model used. Section 8.4 describes MOO problem formulation for bio-diesel production, followed by a brief description of MODE-TL algorithm in Section 8.5. Section 8.6 discusses the obtained results for different optimization cases using MODE-TL algorithm. Section 8.7 presents design optimization results using I-MODE algorithm. The last section summarizes the main findings of this study.

8.2 Process Development

For this study, bio-diesel plant capacity is assumed to be 20,000 metric tons per annum, based on potential waste cooking oil availability in a city of 5 million people such as Singapore (Chua et al., 2010). This process can be divided into two sections: (1) pre-treatment of waste cooking oil using H_2SO_4 to reduce FFA content, and (2) production of bio-diesel from treated oil using NaOH. Alkali catalyst has been chosen for trans-esterification because it can give high yield in short time and bio-diesel production using alkali catalyst is in industrial practice (http://www.lurgi.com/website/Biodiesel.57.0.html?&L=1). The two sections of the bio-diesel process using waste cooking oil are discussed in more details below.

8.2.1 Pre-treatment of Waste Cooking Oil

Zhang et al. (2003a) have presented a process to reduce FFA content in waste cooking oil. In their process, FFA is reacted with methanol in the presence of acid catalyst. A liquid-liquid extraction (LLE) column is used to remove the catalyst from the reactor effluent using fresh glycerol. The extract phase from LLE has glycerol, catalyst and methanol while the raffinate phase has bio-diesel, oil and methanol. Subsequently, two component splitters are used in the study of Zhang et al. (2003a) to remove bio-diesel and oil from the extract phase, and water from the raffinate phase.

Methanol is separated from glycerol using a distillation column, and recycled to the reactor. The treated oil with some FAME and methanol goes to the downstream process where it is converted to bio-diesel. A similar flow-sheet without glycerol-methanol separation has been used by Garcia et al. (2010); the focus of their study is on the prediction of bio-diesel properties.

As a component splitter is not a physical unit, the flow-sheet in Zhang et al. (2003a) is not suitable for economic evaluation of the process. Additionally, it is found from process simulation that a three-phase separator is sufficient, instead of a LLE column, for removing the catalyst from the reactor effluent. Hence, in this study, bio-diesel process flow-sheet of Zhang et al. (2003a) is modified to avoid the component splitter and to use the three-phase separator. Waste cooking oil enters the esterification reactor, where FFAs react with methanol in the presence of sulfuric acid (see the upper half of Figure 8.1). The esterification reactor is maintained at 60^0C and 400 kPa; the reactant streams are pressurized to reactor pressure, and low pressure steam is used to maintain the reactor temperature. The effluent from $CSTR_0$ preheats the waste cooking oil feed. After that, fresh glycerol is mixed with the effluent before it enters the three-phase separator at around 40^0C. Two liquid phases are formed in the three-phase separator; the light phase contains bio-diesel, oil, methanol and water while the heavy phase contains glycerol, catalyst, methanol and water. Two distillation columns (i.e., G-C1 and BD-C1) are used to recover methanol from both outlet streams of the three-phase separator. Number of stages and operating pressures of these columns are shown in Figure 8.1. Columns are under vacuum to avoid decomposition of glycerol and biodiesel at high temperatures. Morais et al. (2010) have reported thermal decomposition temperature of 250 ^{0}C and 150 ^{0}C for pure FAME and glycerol respectively. Presence of methanol with FAME and glycerol

gives additional temperature margin to avoid deterioration of FAME and glycerol. The recovered methanol from both the columns is mixed, pressurized and recycled back to the esterification reactor. The bottom stream from BD-C1 column contains bio-diesel, oil and methanol; this stream is pressurized to about 4 bars and sent to the trans-esterification section. The bottom stream from G-C1 column has mainly glycerol, acid and water.

8.2.2 Bio-diesel Production from Treated Waste Cooking Oil

The alkali-catalyzed trans-esterification is performed at about 60^0C. Pressure depends upon the type of reactor used: ambient pressure for batch reactor (Van Gerpen et al., 2004) and 4 bars for CSTR and PFR (Conneman and Fischer, 1998). Trans-esterification is mass-transfer controlled, and so excess methanol improves oil to bio-diesel conversion (Noureddini and Zhu, 1997). Molar flow ratio of methanol to triglycerides of more than 6 was used by Noureddini and Zhu (1997). Recently, Zhang et al. (2012) developed a process flow-sheet for bio-diesel from pure vegetable oil. In this work, this flow-sheet has been used for bio-diesel production from treated waste cooking oil (see the lower half of Figure 8.1). Continuous stirred tank reactors (CSTRs), distillation columns, three-phase separators, a neutralization reactor and a LLE column are the main units in this process. Three CSTRs are placed in series, and treated oil mixed with methanol and NaOH catalyst enters $CSTR_1$. The effluent mixtures from $CSTR_1$ and $CSTR_2$ individually pass through the three-phase separators, where glycerol with some methanol is separated as the light phase.

The heavy phase from the three-phase separators (S_1-H and S_2-H in Figure 8.1) goes to the next CSTR; this phase mainly contains bio-diesel, oil and methanol. Glycerol streams (i.e., light phases from three-phase separators 1 and 2) are mixed

together, and a distillation column (G-C2) is used to recover unreacted methanol for recycle. Another distillation column (BD-C2) is used to recover methanol from the effluent of $CSTR_3$. Bottom product from this column contains mainly bio-diesel, and is treated in a neutralization unit to remove alkali catalyst, followed by a water wash column. The recovered methanol in the distillate stream of column BD-C2 is also recycled back to trans-esterification reactors. Since the recycled methanol should be free of water, water wash column is used after separating methanol from the reaction mixture.

8.3 Bio-diesel Process Simulation

The bio-diesel process using waste cooking oil (Figure 8.1) has been simulated in the Aspen Hysys V-7.2. Vegetable oil is a mixture of triglycerides of oleic, linoleic, linolenic, palmitic, stearic and other acids. Physical properties of different triglycerides present in vegetable oil are not much different (Myint and El-Halwagi, 2009); hence, one of the triglycerides can be used to represent the vegetable oil. In this work, tri-olein (i.e., triglyceride of oleic acid) is considered as the triglyceride in the waste cooking oil. Physical properties of mono-, di- and tri-olein are taken from Aspen Plus database. Franca et al. (2009) have studied the phase equilibrium of bio-diesel + glycerol + methanol system, and found that experimental LLE data were satisfactorily predicted by the UNIQUAC model. Hence, in this work, this model has been used to predict the physical behavior such as liquid-liquid equilibria. Esterification and trans-esterification kinetics used are respectively from Berrios et al. (2007) and Noureddini and Zhu (1997).

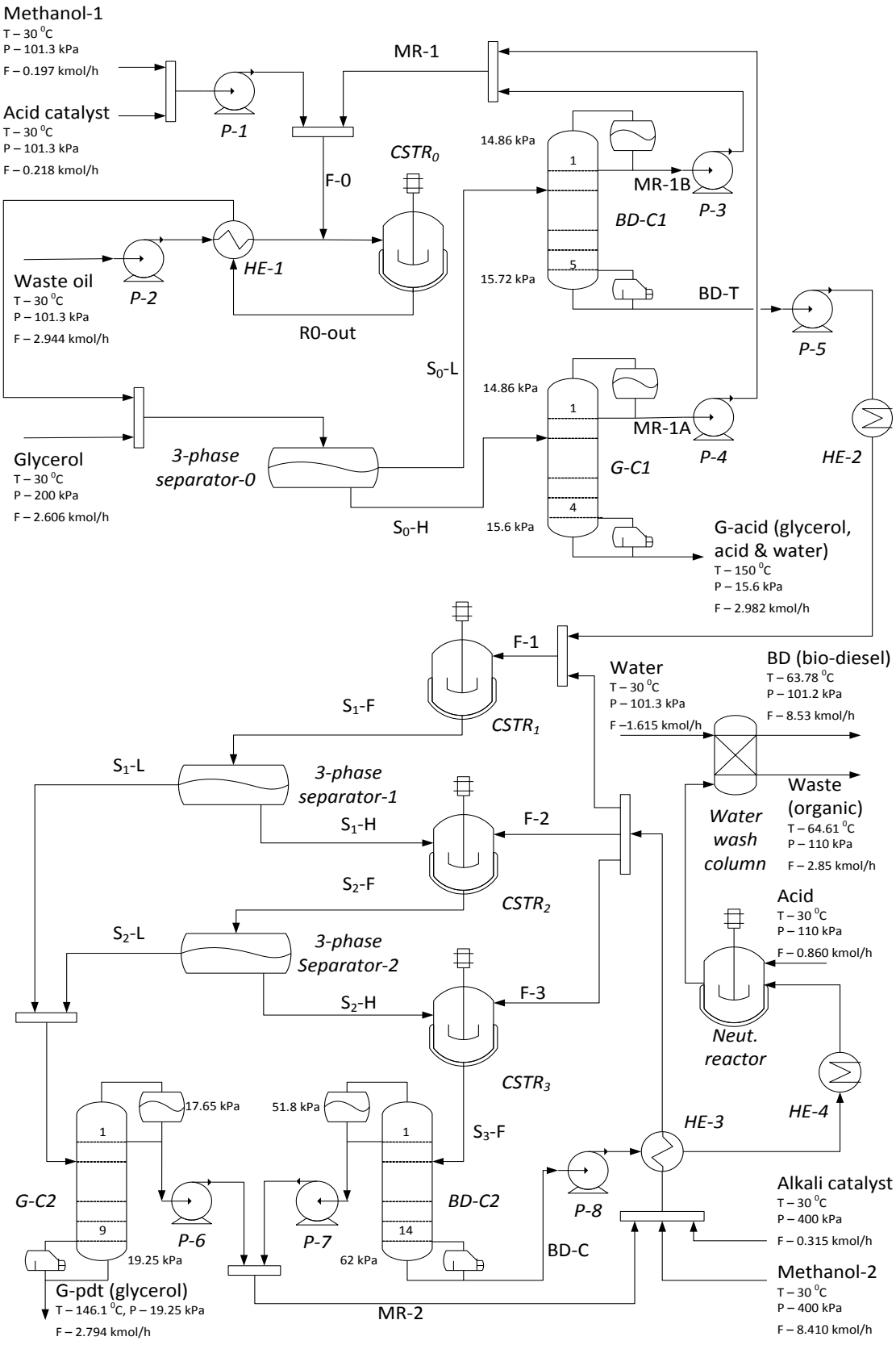


Figure 8.1: Schematic of bio-diesel production plant using waste cooking oil as feed; see Table 8.2 for typical stream data in this process

Pressure in the esterification and trans-esterification reactors is fixed at 400 kPa while reactor temperatures are considered as decision variables in the optimization. As mentioned earlier, both glycerol and bio-diesel are sensitive to high temperature; hence, all distillation columns are under vacuum and consequently lower temperatures to avoid product deterioration. Neutralization reactor and water wash column are at atmospheric pressure. Information for different feed/reactant streams is taken from Zhang et al. (2003a) and Zhang et al. (2012). Figure 8.1 presents important information for all feed/reactant streams in the bio-diesel process. Feed waste cooking oil contains FFA (0.06 mole fraction) and tri-olein (0.94 mole fraction). 2,500 kg of waste cooking oil per hour is processed to produce bio-diesel.

8.4 Multi-objective Optimization Problem Formulation

Bio-diesel process is optimized for profit, fixed capital investment (FCI) and organic waste (i.e., methanol, glycerol, tri/di/mono-olein and oleic acid in the waste stream leaving the water wash column) from the process in Figure 8.1. Profit, which includes revenue and cost of manufacturing (COM), is used as one objective function (equation 8.5 given later). FCI and organic waste represent investment in the plant and potential environmental impact of the plant respectively. The decision variables for optimization are temperature and volume of each of the four CSTRs; these are the crucial process variables for conversion of feed to bio-diesel. The temperature range for each reactor is selected around the suggested operating temperature whereas volume range for each reactor is chosen based on the required residence time.

Nicola et al. (2010) have combined purities of biodiesel and glycerol, and used it as one objective. The present study treats product purities as constraints as these have to satisfy the prevailing requirements (Table 8.1). To prevent product deterioration,

upper limits are imposed on the temperature at the bottom of all distillation columns (Morais et al., 2010). Further, a minimum temperature difference has to be maintained at both ends of heaters/reboilers and coolers/condensers (i.e., 25 ^{0}C for heating and 5^0 C for cooling). Minimum temperature difference at both ends of a heat exchanger is constrained to be 10 ^{0}C. Initially, design of biodiesel process is optimized for two cases; Case A considers trade-off between profit and FCI, whereas trade-off between profit and organic waste is studied in Case B. After that, three cases of operation optimization are performed for a selected process from the design optimization. The operation optimization, Case C provides the basis for the next two cases (D and E), which consider the effect of variation in waste cooking oil feed rate on the process performance. Table 8.1 provides details on these optimization cases considered in this study. Constraints are same in all design and operation optimization cases.

Fixed capital investment (FCI), also known as total module cost (C_{TM}), is calculated using equations 8.1-8.3 below and related data from Turton et al. (2009) for each equipment in Figure 8.1. Equipment purchase cost (C_p) is calculated using equation 8.1 and cost data in Turton et al. (2009). If the required capacity of any process equipment is larger than the capacity range given in Turton et al. (2009), then multiple units of that equipment, with equal capacity, are assumed. Although this increases the capital cost, use of multiple units for costing is reasonable due to equipment availability. Further, this does not affect FCI much in this study since only some optimized solutions require two units of three-phase separators. If the distillation column diameter is less than 0.9 m, then the column is taken to be filled with ceramic packing; otherwise, trays are assumed in the column (West et al., 2008). Bare module cost (C_{BM}) is calculated using equation 8.2; F_{BM} is calculated based on the correlations and data given in Turton et al. (2009).

Table 8.1: Different optimization cases studied for bio-diesel production process

Objective functions in different cases	**Decision variables and their bounds**	**Constraints**
Design optimization		
A. Max. Profit, Min. FCI **B.** Max. Profit, Min. Organic waste (Waste cooking oil feed rate = 2,500 kg/h in both the design cases)	$50 \leq T_{CSTR0} \leq 60^0C$; $50 \leq T_{CSTR1/2/3} \leq 70^0C$; $2 \leq V_{CSTR0} \leq 3\ m^3$; $5 \leq V_{CSTR1/2/3} \leq 10\ m^3$.	$Purity_{BD} \geq 0.99$; $Purity_G \geq 0.95$; $T_{BD\text{-}C1} \leq 250^0C$; $T_{BD\text{-}C2} \leq 250^0C$; $T_{G\text{-}C1} \leq 155^0C$; $T_{G\text{-}C2} \leq 155^0C$ $\Delta T_{both\ ends} \geq 5^0C$ for coolers and condensers; $\Delta T_{both\ ends} \geq 25^0C$ for heaters and reboilers; $\Delta T_{both\ ends} \geq 10^0C$ for heat exchangers.
Operation optimization		
C. Max. Profit, Min. Organic waste (waste cooking oil feed rate = 2,500 kg/h)	$50 \leq T_{CSTR0} \leq 60^0C$; $50 \leq T_{CSTR1/2/3} \leq 70^0C$.	
D. Max. Profit, Min. Organic waste (2,700 kg/h, i.e., 10% increase in waste cooking oil feed rate)		
E. Max. Profit, Min. Organic waste (2,000 kg/h, i.e., 20% decrease in waste cooking oil feed rate)		

In Figure 8.1, if the fluid inside an equipment contains acid, then stainless steel (SS) is used as material of construction (MOC). Accordingly, CSTR0, 3-phase separator-0, BD-C1, G-C1, BD-C2, G-C2, neut. reactor, P-1 are made of SS, whereas CS-shell/SS-tube is used as MOC for HE-1, condenser and reboiler of BD-C1. MOC for the remaining equipments is carbon steel. C_{TM} is 1.18 times bare module cost of all equipments (equation 8.3). Chemical Engineering Plant Cost Index, CEPCI of 600 is used to account for inflation.

$$\log(Cp) = K_1 + K_2\log(Capacity) + K_3[\log(Capacity)]^2 \quad (8.1)$$

$$C_{BM} = \sum_{\text{all equipments}} C_p F_{BM} \quad (8.2)$$

$$\text{FCI or } C_{TM} = 1.18 C_{BM} \quad (8.3)$$

Cost of manufacture (COM) is given by equation 8.4 Turton et al. (2009), and profit is the difference between the revenue (earned by selling the products: bio-diesel and glycerol) and COM (see equation 8.5).

$$COM = 0.28FCI + 2.73(\text{operating labor}) + 1.23(\text{utilities} + \text{raw material cost}) \quad (8.4)$$

$$\text{Profit} = \text{Revenue} - COM \quad (8.5)$$

Costs of waste cooking oil (= 0.39 $/kg), methanol (= 0.28 $/kg), glycerol (= 1.1 and 1.15 $/kg for 95 and 99 wt.%, respectively), NaOH (= 0.75 $/kg for 37 wt.%), HCl (= 0.92 $/kg), and steam at 106 bar (= 0.032 $/kg) are obtained from our industry contacts. Cost of sulfuric acid (= 0.071 $/kg) is obtained from http://www.ICIS.com (access date: August, 2011), while cost of bio-diesel (= 1.464 $/kg) is taken from http://www.biofueloasis.com (access date: August, 2011). Finally, costs of (process, chilled and cooling) water, (low, medium and high pressure) steam and electricity are taken from Turton et al. (2009).

8.5 Multi-objective Differential Evolution with Taboo List

In many process optimization problems, evaluation of objectives and constraints at each trial solution requires simulation of the entire process, and so it is often computationally expensive. Use of taboo list (TL) avoids the revisit of search space by keeping a record of recently visited points. So, multi-objective DE (MODE) with taboo list (TL) can avoid unnecessary function evaluations (Srinivas and Rangaiah, 2007). Figure 8.2 shows the flowchart of MODE with TL (MODE-TL) algorithm.

Taboo check is implemented in the evaluation step of trial vector/individual; if the trial individual is near to any individual in the TL by a specified distance, then it is rejected without calculating objectives and constraints. MODE-TL algorithm handles inequality constraints by constrained-dominance approach of Deb et al. (2002). This algorithm is implemented using Excel worksheets (to calculate the objective functions: profit, FCI and organic waste, constraints and linking between cells) and Visual Basic for Applications (to implement the algorithm steps and also to interface the bio-diesel process simulation in Aspen Hysys Version 7.2 with the Excel based MODE-TL algorithm).

8.6 Optimization Results

Values of MODE-TL algorithm parameters used for all optimization cases are: population size (N) = 100, crossover probability (Cr) = 0.3, mutation rate (F) = 0.5 and maximum number of generations (MNG) = 100. Size of TL is fixed at half of N (= 50), and taboo radius (TR) of 0.01 is chosen for all cases, to decide acceptance/rejection of new trial individuals. Euclidean distance between the newly generated trial individual and each individual in the TL is calculated for normalized values (between 0 and 1) of decision variables. These values of algorithm parameters are selected based on our experience on solving many benchmark functions using MODE-TL. Aspen Hysys has convergence difficulty for some combination of decision variables, and the associated error message need to be clicked manually. So, computation time required in the complete optimization cannot be obtained; about 30 seconds are required to complete one simulation (based on 10 different runs). An optimization run for 100 generations with a population of 100 will then require about 4 days in the absence of errors messages.

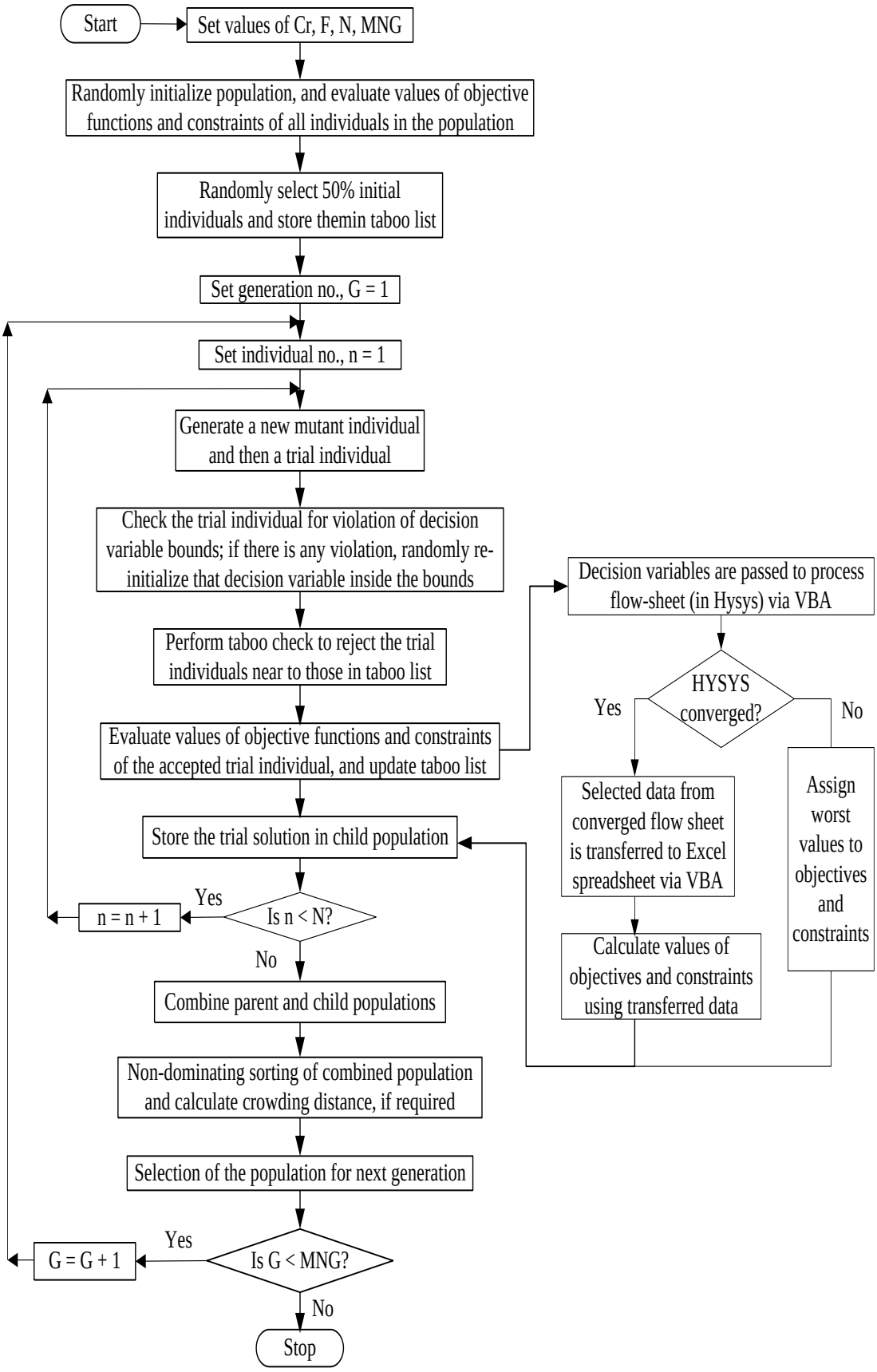


Figure 8.2: Flowchart of the MODE-TL algorithm and its implementation

8.6.1 Design Optimization

A. Trade-off between Profit and FCI

The Pareto-optimal front obtained for optimizing profit and FCI simultaneously is shown in Figure 8.3(a). As mentioned earlier, fixed amount of waste cooking oil is converted into bio-diesel; therefore, both objectives in the obtained Pareto-optimal front are varying over small ranges (Figure 8.3a). West et al. (2008) reported FCI of 1.1 million $ (CEPCI = 394) for a bio-diesel plant of capacity 8,000 metric tons/annum and waste cooking oil as the feed-stock. Projected FCI for a plant capacity of 20,000 metric tons/annum, using the six-tenth rule (Turton et al., 2009) and when CEPCI = 600, is 2.88 million $, which is comparable to the FCI range in Figure 8.3a (i.e., 2.86-2.94 million $).

The trend in the Pareto-optimal front can be visually correlated to the decision variables: T_{CSTR1} and T_{CSTR2} (Figures 8.3a, 8.3c and 8.3d). Larger amount of bio-diesel is produced at higher T_{CSTR1} and T_{CSTR2}, which results in increased revenue and profit. Light and heavy phases in the 3-phase separator-1 and separator-2 have a smaller difference in densities at high temperature, which increases required size of these separators. Consequently, FCI of the bio-diesel production process increases with increase in the reactor temperature. Temperature of $CSTR_0$ is also increasing in a small range with increasing profit, while there is no particular trend in the temperature of $CSTR_3$ (Figures 8.3b and 8.3e). Further, smaller reactors can reduce FCI of process; hence, volume of each reactor is nearly constant at its lower bound (i.e., esterification: 2 m^3 and trans-esterification: 5 m^3). Figure 8.3(f) shows variation in $CSTR_0$ volume while variations in the remaining reactor volumes are not shown, for brevity.

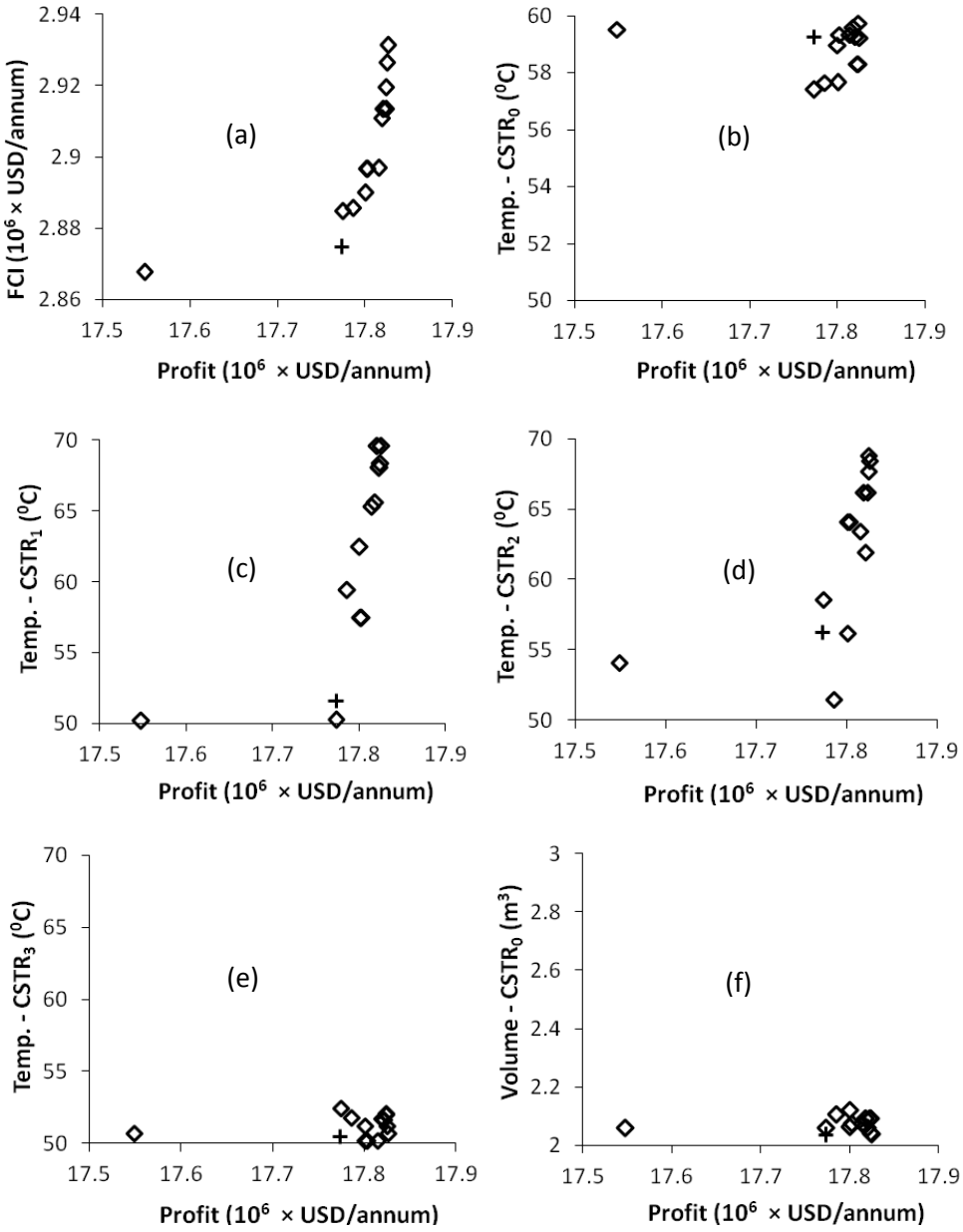


Figure 8.3: Selected results for simultaneous maximization of profit and minimization of FCI

Each solution on the Pareto-optimal front is equally good from the point of specified objectives, and decision maker can select one solution based on his/her experience and/or requirement. One corner solution on the Pareto-optimal front, shown as "+" in Figure 8.3(a), has been selected for discussion and for further study. Table 8.2 presents key information for important streams corresponding to this optimal solution.

Table 8.2: Important data of selected streams in Figure 8.1, corresponding to the optimal solution "+" in Figure 8.3(a); total molar flow is in kmol/h and total mass flow is in kg/h

Stream	**R_0-out**	**S_0-L**	**S_0-H**	**G-Acid**	**BD-T**	**MR-1A**	**MR-1B**
Temperature (^{0}C)	59.25	42.77	42.77	150	244.8	23.98	25.38
Pressure (kPa)	400	200	200	15.6	15.72	14.86	14.86
Total molar flow	12.56	11.22	3.94	2.982	2.983	0.962	8.237
Total mass flow	2885	2832	293.5	264.4	2503	29.18	328.2
Mole fractions							
-Tri-olein	0.223	0.252	0	0	0.923	0	0.008
-Methanol	0.568	0.560	0.216	0.003	0.006	0.878	0.760
-Methyl oleate	0.024	0.027	0	0	0.057	0	0.015
-Glycerol	0	0	0.661	0.874	0	0	0
-Water	0.167	0.161	0.068	0.050	0.007	0.122	0.216
-H_2SO_4	0.018	0	0.055	0.073	0	0	0

Stream	**F-1**	**S_1-F**	**S_1-H**	**S_2-F**	**S_2-H**	**S_3-F**	**BD-C**	**BD**	**G-pdt**	**Waste**
Temperature (^{0}C)	64.33	51.59	51.59	56.24	56.24	50.47	226.4	63.78	146.1	64.61
Pressure (kPa)	400	400	400	400	400	400	62	101.2	19.25	110
Total molar flow	31.36	31.36	24.99	27.55	26.09	26.46	8.91	8.53	2.794	2.85
Total mass flow	3413	3413	3074	3156	3083	3095	2533	2495	250.7	87.6
Mole fractions										
-Tri-olein	0.088	0.010	0.012	0.002	0.002	0	0.001	0	0	0.004
-Di-olein	0	0.005	0.006	0.001	0.001	0	0.001	0	0	0.003
-Mono-olein	0	0.002	0.003	0.001	0.001	0	0	0	0	0
-Methanol	0.896	0.673	0.681	0.670	0.669	0.668	0.014	0.012	0.018	0.009
-Methyl oleate	0.005	0.229	0.287	0.300	0.317	0.318	0.946	0.985	0.001	0.008
-Glycerol	0	0.072	0	0.016	0	0.003	0.008	0	0.955	0.027
-Water	0.001	0.001	0.001	0.001	0.001	0.001	0	0.003	0.007	0.858

B. Trade-off between Profit and Organic Waste

The Pareto-optimal front obtained between profit and organic waste is shown in Figure 8.4(a). It can be divided into two parts; initially, profit (i.e., 17.795-17.815 million $) varies linearly with the organic waste, and, finally, organic waste increases faster for a very small improvement in the profit (~ 17.82 million $). The initial improvement in profit is due to increase in T_{CSTR0} and T_{CSTR2}, and decrease in V_{CSTR1} (Figures 8.4a, 8.4b, 8.4d and 8.4g), while the remaining decision variables are either constant or scattered. High values of T_{CSTR0} and T_{CSTR2} increase conversion of waste cooking oil in $CSTR_0$ and $CSTR_2$, which improves the profit.

Temperature of $CSTR_1$ is nearly constant at its upper bound of 70 ^{0}C (Figure 8.4c). Variations in T_{CSTR0}, T_{CSTR2}, T_{CSTR3}, V_{CSTR0}, V_{CSTR1} and V_{CSTR2} are contributing to the second part of the obtained Pareto-optimal front (see Figures 8.4a, 8.4b, 8.4d-h). Decrease in reactor volumes improves the profit by decreasing FCI, while amount of organic waste increases due to reduction in residence time. On the other hand, partly to compensate the effects of decrease in reactor volumes, high value of reactor temperature improves the profit and also decreases the organic waste. For example, profit and organic waste corresponding to solution "O" in Figure 8.4(a) are 1.78×10^7 USD/annum and 64,519 kg/annum respectively. Now, values of different reactor volumes for this solution are replaced by those corresponding to solution "□" (relatively smaller values; Figure 8.4a); organic waste corresponding to the modified set of decision variables is 68,867 kg/annum with some improvement in profit. Organic waste corresponding to solution "□" is slightly lower at 67,118 kg/annum due to reactor temperatures higher than those of the modified solution.

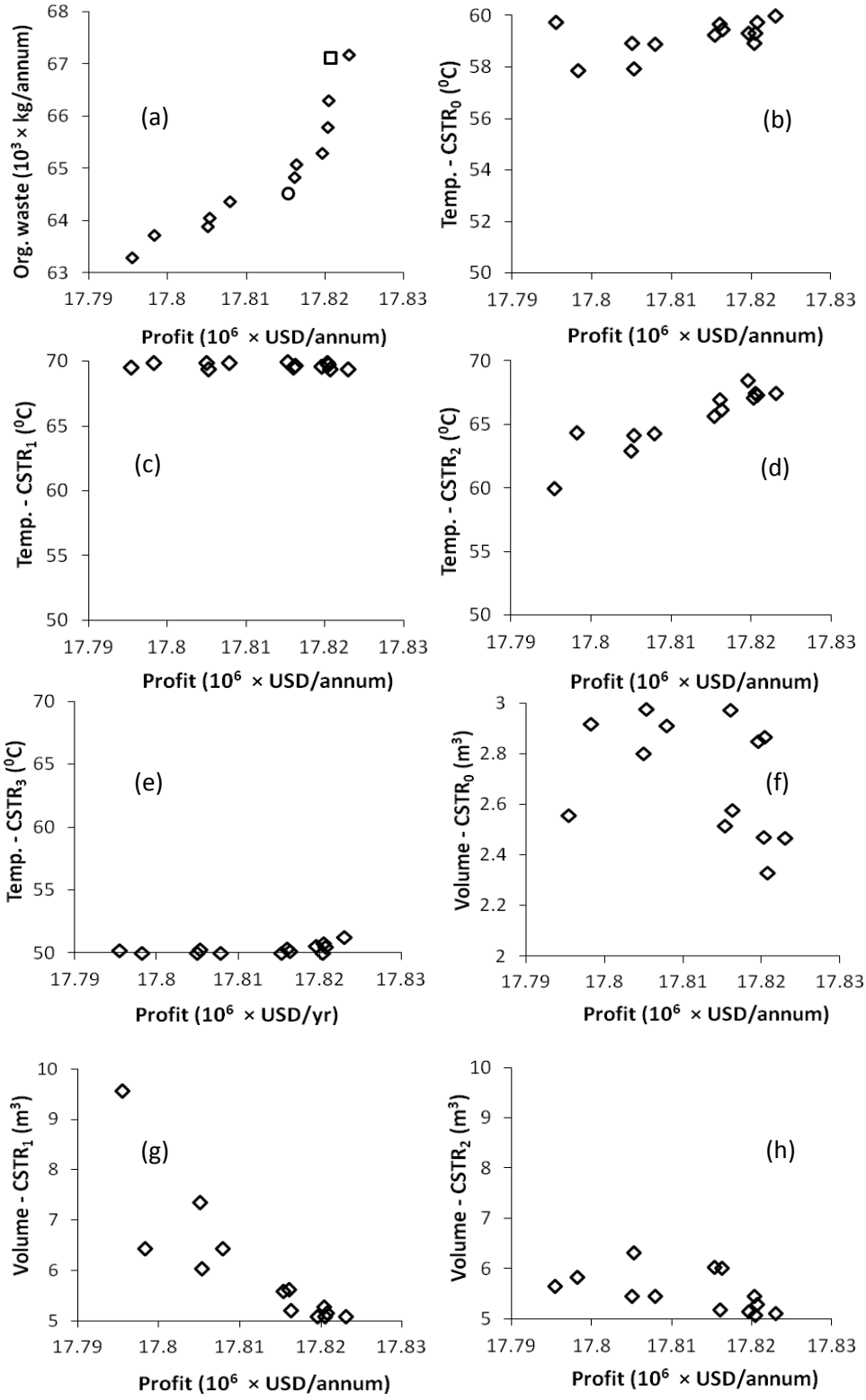


Figure 8.4: Selected results for simultaneous profit maximization and organic waste minimization

Thus, different combinations of decision variables are giving different non-dominated solutions between these two solutions ("O" and "□"). T_{CSTR3} is constant at

50 ^{0}C in the first part, and marginal increase in it can be seen in the second part, which is required to convert un-reacted feed. As a 3-phase separator is not used after $CSTR_3$, un-reacted feed and glycerol appear in the waste stream. Volume of $CSTR_3$ is constant near its lower bound (5 m^3), and is not shown in Figure 8.4 for brevity.

8.6.2 Operation Optimization

In operation optimization, only operation variables are considered as decision variables, while design variables related to equipment sizes are fixed based on the design optimization (Section 8.6.1 A). A bio-diesel process design shown as "+" in Figure 8.3(a) has been selected for studying trade-off between profit and organic waste, for variations in waste cooking oil feed rate. In all operation optimization cases, ranges of operation variables are same as those used in the design optimization (see Table 8.1).

C. Trade-off between Profit and Organic Waste: "Base Case" - Waste Cooking Oil Feed Rate = 2,500 kg/h

The selected design is optimized for the nominal value of waste cooking oil feed rate (i.e., 2,500 kg/h). Although effect of operation variables on profit, and processing capacity has been studied in the design optimization case (Section 8.6.1 B), this optimization case is required for comparing different operation optimization cases. The obtained Pareto-optimal front for the "base" case of operation optimization is not shown as variation in profit value (18.826×10^6 USD/annum) is very small. Organic waste varies in a narrow range (66.4×10^3 to 67.0×10^3 kg/annum) compared to the design optimization case, which means design variables (i.e., size of reactors) have larger effect on the amount of organic waste from the process (see Figure 8.4a). Temperatures of $CSTR_0$, $CSTR_1$ and $CSTR_2$ are nearly constant, close to their

respective upper bound, while T_{CSTR3} is nearly constant at its lower bound (not shown for brevity). In conclusion, all decision variables in operation optimization favor both objectives similarly, which result in their constant values near to one of their bounds.

D. Trade-off between Profit and Organic Waste: 10% Increase in Waste Cooking Oil Feed Rate to 2,750 kg/h

In this case, 10% additional waste cooking oil is processed compared to "base" case of operation. The obtained Pareto-optimal front is shown in Figure 8.5(a). Similar to "base" case, objective values vary over small ranges. Increase in feed rate of plant does not increase FCI and operating labor as number of processing units is fixed. Consequently, profit for 10% increase in waste cooking oil feed rate, is slightly higher (by 10.9%) compared to "base" case. In this optimization for increased feed rate, temperatures of $CSTR_1$ and $CSTR_2$ are constant near their upper bound of 70 ^{0}C (not shown for brevity). Variation in temperature of $CSTR_0$ with profit has opposite trend to the Pareto-optimal front (see Figures 8.5a and 8.5b). Lower value of T_{CSTR0} leads to lower conversion of oleic acid into bio-diesel in the pre-treatment reactor, which increases organic waste from the process. Temperature of $CSTR_3$ is also increasing with profit, to its upper bound (Figures 8.5a and 8.5c). In this case of increased feed, organic waste has increased by 9-13% compared to "base" case. Temperatures of $CSTR_1$ and $CSTR_2$ are nearly same at 70 ^{0}C in both the cases, which means more un-reacted feed enters $CSTR_3$ in the increased feed rate case compared to "base" case. When larger amount of un-reacted feed enters $CSTR_3$, high temperature of $CSTR_3$ favors both the objectives. Compared to base case, it can be observed that temperature of $CSTR_3$ is higher and increasing with profit (Figure 8.5c), which is required to convert un-reacted feed into bio-diesel.

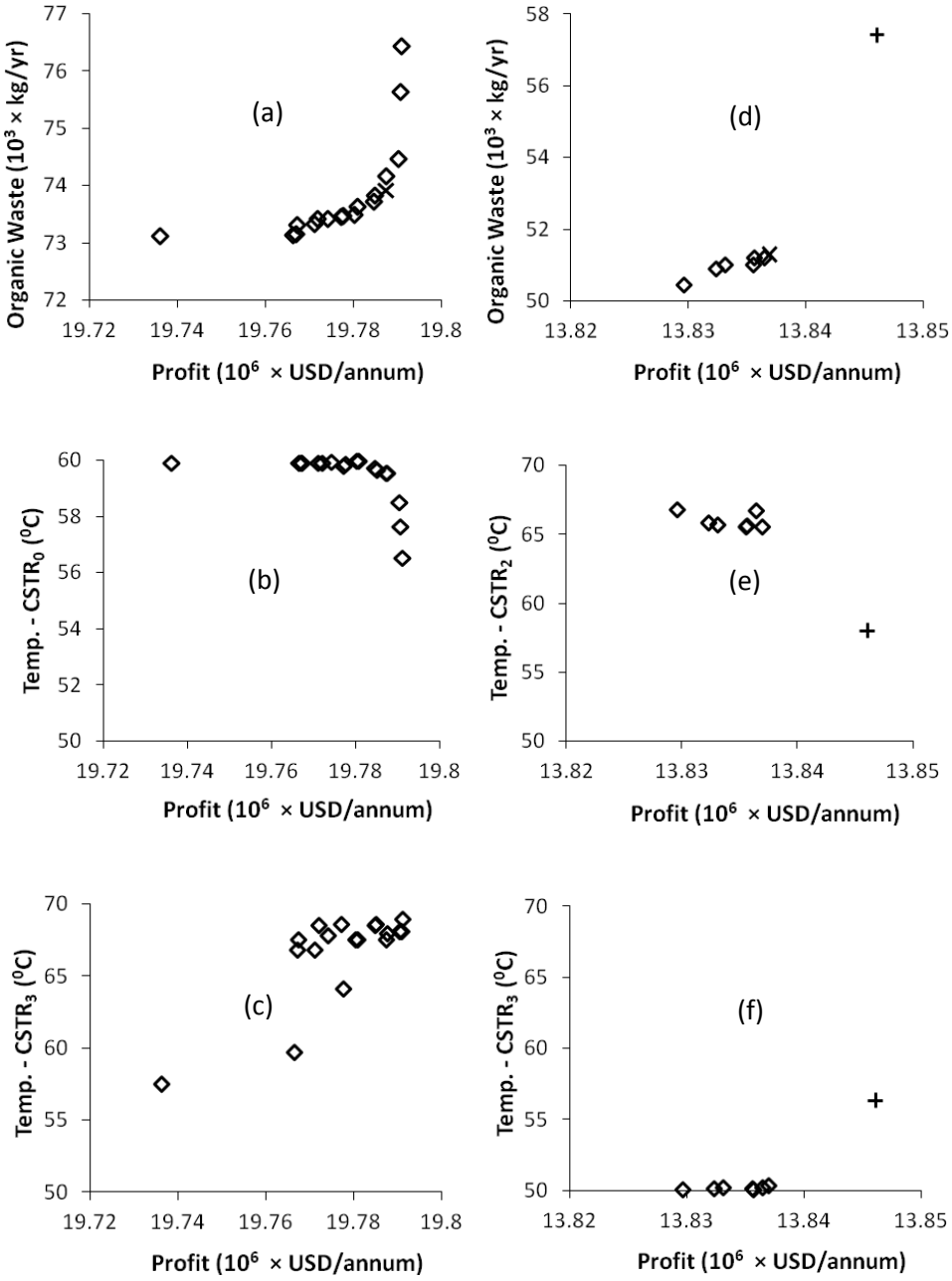


Figure 8.5: Selected results for simultaneous profit maximization and organic waste minimization: 10% increase in waste cooking oil feed rate (plots a, b and c on the left side), 20% decrease in waste cooking oil feed rate (plots d, e and f on the right side)

E. Trade-off between Profit and Organic Waste: 20% Decrease in Waste Cooking Oil Processing Rate to 2,000 kg/h

Figure 8.5(d) shows the obtained Pareto-optimal front between profit and organic waste, for 20% decrease in waste cooking oil feed rate compared to the "base" case.

As expected profit has decreased and organic waste is lower by 8-25% compared to the base case, although the latter is varying over a wide range. So, this requires further analysis of results obtained in both the cases. Now, profit is about 22% lower than that in the base case, which is due to larger magnitude of constant terms in the profit calculation. Revenue obtained in this case is nearly 20% lower than that in the "base" case, which means that waste cooking oil to bio-diesel conversion is the same in both the cases. Similar to the "base" case, temperatures of $CSTR_0$ and $CSTR_1$ are constant, near to their respective upper bounds (not shown for brevity). T_{CSTR2} is varying between its lower and upper bounds, while T_{CSTR3} is constant near its lower bound except a high temperature corresponding to the large amount of organic waste (see solution "+" in Figure 8.5f). Variation in temperature of $CSTR_2$ shows opposite trend to the Pareto-optimal front (see Figures 8.5d and 8.5e).

In the decreased feed rate case, one solution shown as "+" in Figure 8.5d is further away from the remaining non-dominated solutions; organic waste has increased significantly (~ 57×10^3 kg/annum) for this solution. Temperature of $CSTR_2$ is lower for solution "+" than that for the remaining non-dominated solutions. At lower T_{CSTR2}, more un-reacted feed enters $CSTR_3$, which is converted into bio-diesel at the higher temperature of $CSTR_3$. Organic waste (i.e., feed and glycerol) from $CSTR_3$ goes to the downstream units as it is not removed using a three-phase separator. Hence, organic waste has increased suddenly, while profit has also increased as larger amount of bio-diesel is produced at high value of T_{CSTR3}.

One corner solution on the Pareto-optima front of "base" case and two corner solutions on the Pareto-optimal fronts, shown as "×" in Figures 8.5(a) and 8.5(d), have been selected for comparison. Table 8.3 presents values of objectives and decision variables corresponding to these selected solutions. Temperatures of $CSTR_0$

and $CSTR_1$ are constant, near to their respective upper bounds (i.e., 60 and 70 ^{0}C) in all three cases. T_{CSTR2} increases with the feed rate increase from 2,000 to 2,500 kg/h; it is limited by its upper bound for feed rate of 2,750 kg/h. Temperature of $CSTR_3$ is near to its lower bound for the feed rates of 2,000 and 2,500 kg/h, and increases for feed rate of 2,750 kg/h as larger amount of feed has to be converted into bio-diesel in $CSTR_3$.

Table 8.3: Comparison of three optimal solutions chosen for different feed rates (one solution from "base" case and solutions shown as "×" in Figures 8.5a and 8.5d)

Objectives and decision variables	**Decreased feed rate (2,000 kg/h)**	**Normal feed rate (2,500 kg/h)**	**Increased feed rate (2,750 kg/h)**
Profit (million $)	13.84	17.83	19.79
Org. waste (kg/annum)	51,300	66,430	73,917
T_{CSTR0} (^{0}C)	59.85	59.97	59.55
T_{CSTR1} (^{0}C)	69.17	69.96	69.87
T_{CSTR2} (^{0}C)	65.56	69.99	69.84
T_{CSTR3} (^{0}C)	50.40	50.74	67.59

8.7 Design Optimization using I-MODE Algorithm

In Section 8.6.1, MODE-TL with MNG = 100 is used to obtain the non-dominated solutions. Here, I-MODE is used to solve the above two design optimization problems (Sections 8.6.1 A and B). For this, recommended values of termination parameters (δ_{GD} = 0.0003 and δ_{SP} = 0.1) are used in I-MODE (see Chapter 3). Further, taboo radius of 0.01 and MNG of 100 are used in I-MODE. The Pareto-optimal front obtained for maximization of profit and minimization of FCI at GT = 82 (Figure 8.6a) is comparable to the Pareto-optimal front presented in Section 8.6.1 A; I-MODE

algorithm is not able to obtain non-dominated solutions at high FCI (~ 2.93×10^6 USD/annum). In the case of maximization of profit and minimization of organic waste, I-MODE terminates based on MNG (= 100). Figure 8.6(b) shows Pareto-optimal fronts obtained using I-MODE and MODE-TL (see Section 8.6.1 B). In this case, I-MODE is not able to find the non-dominated solutions for low value of profit, but it gives better non-dominated solutions for high value of profit. This may be due to use of fixed parameters values in MODE-TL compared to self-adaptation of parameters in I-MODE, apart from the stochastic nature of search.

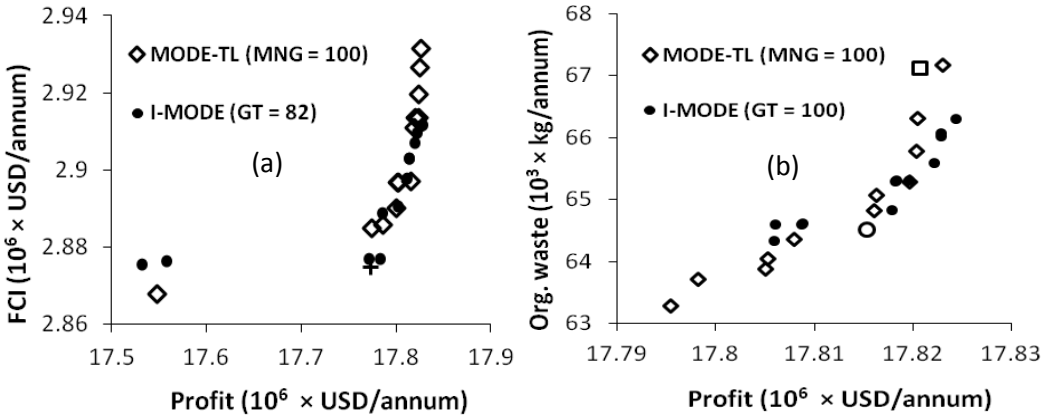


Figure 8.6: Non-dominated solutions obtained for design optimization of bio-diesel process: (a) max. profit and min. FCI, and (b) max. profit and min. organic waste

8.8 Conclusions

In this study, bio-diesel production process is simulated in Aspen Hysys; this process uses waste cooking oil as the feed, which facilitates its better utilization and promotes sustainability. Multi-objective differential evolution with taboo list is used to obtain trade-off solutions among different objectives. Two bi-objective design optimization problems are solved to study the effect of important design and operation variables on the performance of this process. First, trade-off between profit and FCI is explored in design optimization, and it is found that objectives do not vary much due

to the fixed amount of feed rate. Next, bio-diesel process is optimized for maximization of profit and minimization of organic waste simultaneously, which examines environmental impact of the process design. Following this, operation of a selected bio-diesel process design, in the presence of feed rate variation, has been successfully optimized for profit maximization and organic waste minimization simultaneously. In short, the study demonstrates the potential of MOO for bio-diesel process design and operation. The results show that amount of organic waste can change significantly with only small variation in the economic objective, and MOO provides alternative process designs with different environmental impacts, for implementation.

Chapter 9

Multi-objective Optimization of a Membrane Distillation System for Desalination of Sea Water[7]

9.1 Introduction

Water scarcity around the world has led to drinking water production from sea and brackish water. Production of drinking water by membrane processes is capital and energy efficient compared to other processes. Over the decades, reverse osmosis (RO) has been used as an attractive choice to purify water for both industrial and household usage. Membrane distillation (MD), forward osmosis (FO) and membrane crystallizers (MCr) are other important membrane processes for water purification, and these processes are gaining attention now. Of these, MD is an attractive choice to purify saline water; it requires low temperature and pressure compared to multi-stage flash distillation and RO respectively. Additionally, it has good performance at high salt concentration.

MD can use low-grade waste heat or renewable energy to produce drinking water. The performance of MD module, although depends on the membrane transport properties, can be improved by better module and process design. Hence, this work focuses on the design of a small scale MD system for multiple objectives; its maximum capacity is about 500 liters/day, which is sufficient for a house-hold (e.g., water consumption in Singapore is 153 liters/capita /day in 2012). Several researchers have optimized large scale membrane processes, mainly RO, for water production

[7] This chapter is based on the conference paper: Sharma, S. and Rangaiah, G. P. (2012), Multi-objective optimization of a membrane distillation system for desalination of sea water, European Symposium on Computer Aided Chemical Engineering (ESCAPE 22), London.

(e.g., Villafafila and Mujtaba, 2003; Guria et al., 2005b). Criscuoli et al. (2008) evaluated energy requirement for a lab-scale MD module (area = 40 cm^2). Recently, Zuo et al. (2011) have studied the effect of different combinations of MD design and operating variables on water flux, gain output ratio and production cost. In the literature, rigorous optimization of the MD system for multiple objectives has not been studied.

In this work, design of a MD module and process is optimized for high water production rate, lower energy consumption and lower brine disposal rate. MOO is performed to explore the trade-off between these conflicting performance criteria. The obtained results provide optimal designs of the MD process for different water production rate. A direct contact MD (DCMD) module has been simulated in Aspen Custom Modeler (ACM), whereas remaining process units are simulated in Aspen Plus. Rigorous mathematical model is used for predicting the performance of MD module (Song et al., 2007). Further, cross flow of feed inside the MD module is chosen due to its lower temperature polarization effect (Khayet and Matsuura, 2010).

The next section of this chapter presents design and simulation of MD system. Section 9.3 describes formulation of two bi-objective optimization problems. Optimization results for both the cases are presented in Section 9.4. Finally, finding of this work are summarized at the end of this chapter.

9.2 Membrane Distillation System Design and its Simulation

A schematic diagram of MD system is presented in Figure 9.1. The hollow fibers are fixed inside the rectangular assembly. Sea-water/make-up feed (M_2) is pre-heated in a heat exchanger using the hot permeate stream and then mixed with the concentrate stream (F_2) leaving MD unit. After purging a small part of this mixed

stream (F_3), it is further heated to the desired temperature level (T_{f1}) in a heater. Then, the hot feed (F_1) enters on the shell-side of MD, and some amount of water is transferred through the wall of fibers. Hence, temperature and concentration of outlet feed stream (F_2) decreases and increases respectively. Make-up feed (M_2) is used to maintain the concentration of stream F_2 at the desired level (c_{f1}). Pure water (P_1) at 35^0C passes through the hollow fibers; the outlet stream (P_2) has larger flow rate and higher temperature compared to the inlet stream P_1. Stream P_2 is used to pre-heat the make-up feed (M_1) and then cooled in an air cooler to the desired temperature ($35\ ^0$C).

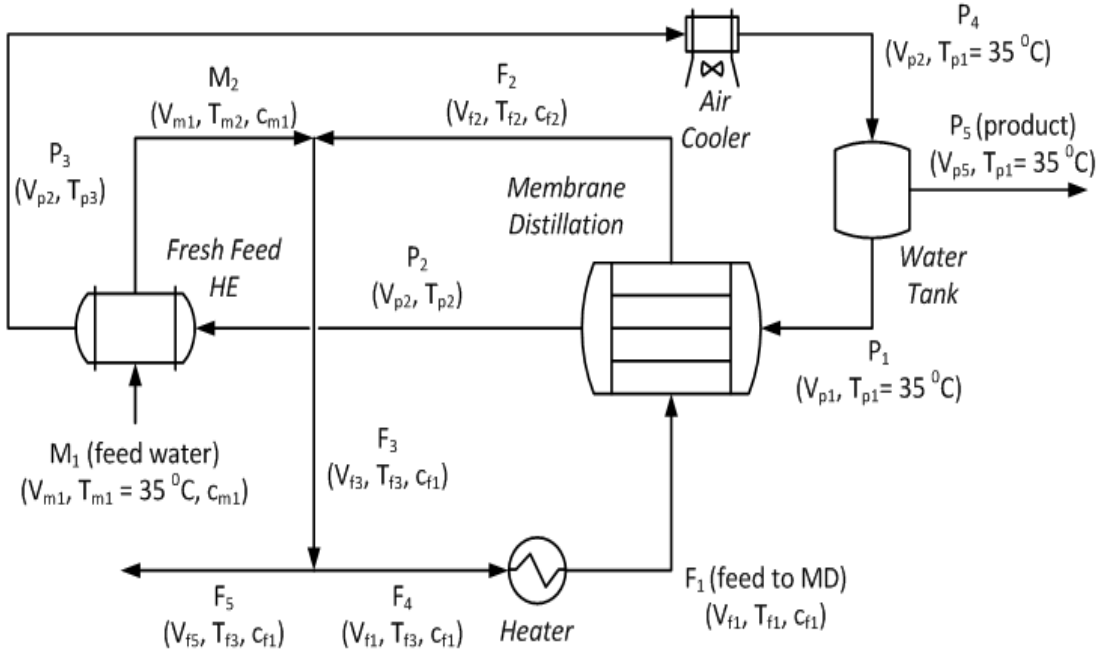


Figure 9.1: Schematic of membrane distillation module and process

The mathematical model of MD module is taken from Song et al. (2007). Plug flow of feed is assumed on the shell side, i.e., no mixing of feed after each fiber layer (see Figure 9.2). MD module has many fiber layers (e.g., 16 layers in Figure 9.2), and each layer can have numerous fibers. For this study, each fiber layer is assumed to have 68 fibers (Song et al., 2007). Equation 9.1 presents water vapor flux through

membrane at any location x in MD fiber. Fiber-side mass and energy balances are given by equations 9.2 and 9.3 respectively. Amount of heat transferred from feed to permeate is compared in equations 9.4 and 9.5; the three heat transfer terms correspond to the feed side, fiber wall and permeate side resistances, respectively.

$$N_v(x) = k_m \frac{(d_o - d_i)}{d_i \ln(d_o - d_i)} \left(p_{fm}(x) - p_{pm}(x)\right) \tag{9.1}$$

$$\int_0^L N_v(x) \pi d_i dx = V_{p2}\rho_{p2} - V_{p1}\rho_{p1} \tag{9.2}$$

$$\int_0^L h_p \pi d_i \left(T_{pm}(x) - T_p(x)\right) dx = V_{p2}\rho_{p2}C_{p2}T_{p2} - V_{p1}\rho_{p1}C_{p1}T_{p1} \tag{9.3}$$

$$h_f \pi d_o \left(T_{f1}(x) - T_{fm}(x)\right) = h_m \frac{\pi(d_o - d_i)}{\ln(d_o - d_i)} \left(T_{fm}(x) - T_{pm}(x)\right) + N_v(x)\pi d_i \times \left(\Delta H_{v,pm} + C_{pm}T_{pm}(x)\right) = h_p \pi d_i \left(T_{pm}(x) - T_{p2}(x)\right) \tag{9.4 \& 9.5}$$

In the above equations, N_v, k_m and h_m are respectively water vapor flux through membrane, membrane mass and heat transfer coefficients. p_{fm} and p_{pm} are water vapor partial pressures at the membrane surface on the feed and permeate side, respectively. h_p and h_f are tube- and shell-side heat transfer coefficients, respectively. Specific details on p_{fm}, p_{pm}, h_p and h_f calculations can be found in Song et al. (2007). L is the length of hollow fiber. T, V, ρ and C are respectively temperature, volumetric flow rate, density and specific heat at different locations. Subscripts f1, fm, p1, pm and p2 refer to the locations in bulk feed, shell-side membrane surface, permeate inlet, tube-side membrane surface and permeate outlet respectively (see Figure 9.2). $\Delta H_{v,pm}$ is the heat of vaporization of water at temperature T_{pm}.

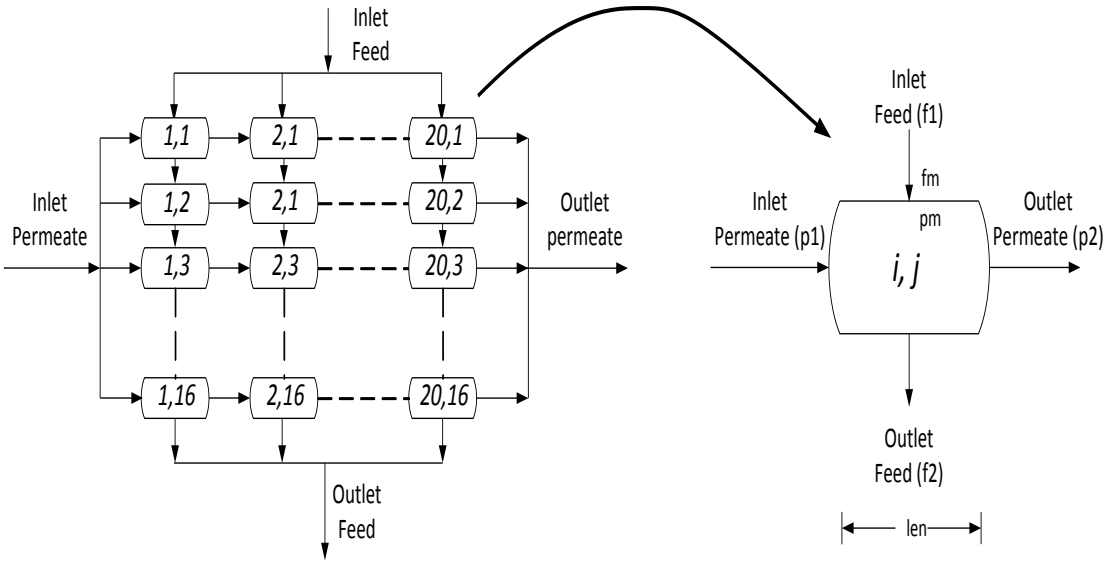


Figure 9.2: Simulation strategy used for solving MD module (i - discretized part number, j - fiber layer number; len - length of each discretized part)

In order to solve MD model for a membrane fiber, each membrane fiber is discretized into 20 parts (see Figure 9.2). The above MD process model is solved for each of these parts. Water vapor flux can be considered constant for each part. The discretization converts the integral terms in equations 9.2 and 9.3 into summation terms. The process model for each part has 5 equations (i.e., 9.1-9.5) with 5 unknowns, namely, T_{fm}, T_{pm}, T_{p2}, N_v and V_{p2}. Values of inlet feed and inlet permeate variables (i.e., T, V, c - concentration) are assumed to be known, and parameters for MD module are given in Table 9.1. Further, mass and heat balances are applied around each fiber part, to obtain outlet feed flow rate, temperature and concentration by the following equations.

$$V_{f1}\rho_{f1} + V_{p1}\rho_{p1} = V_{f2}\rho_{f2} + V_{p2}\rho_{p2} \tag{9.6}$$

$$V_{f1}\rho_{f1}Z_{NaCl,f1} = V_{f2}\rho_{f2}Z_{NaCl,f2} \tag{9.7}$$

$$V_{f1}\rho_{f1}C_{f1}T_{f1} + V_{p1}\rho_{p1}C_{p1}T_{p1} = V_{f2}\rho_{f2}C_{f2}T_{f2} + V_{p2}\rho_{p2}C_{p2}T_{p2} \tag{9.8}$$

Here, equations 9.6, 9.7 and 9.8 are respectively mass, salt and energy balances around each part of fiber. Z_{NaCl} is the mass fraction of salt, and subscript f2 corresponds to the feed outlet (see Figure 9.2). ACM is used to solve these model equations (i.e., 9.1-9.8) for each fiber part, and ELECNRTL thermodynamic model is used for prediction of properties.

Table 9.1: Parameters of membrane distillation module (Song et al., 2007)

Parameter	**Value**
Internal diameter of fiber (d_i)	0.00033 m
External diameter of fiber (d_o)	0.00048 m
Membrane porosity	0.8
Membrane mass transfer coefficient, k_m	0.0024 kg/(m^2.h.Pa)
Membrane heat transfer coefficient, h_m	733.33 W/(m^2.K)
Membrane surface area	0.2864 m^2

9.3 Multi-objective Optimization Problem Formulation

Water production rate, total energy required to heat the feed and to pump the liquid through process units, and brine disposal rate are important objectives related to economics and environmental impact. Energy required by air cooler is nearly proportional to the energy input in heater, so it is not included in the energy objective. Performance of MD module can be affected by several inlet variables, e.g., feed temperature, feed concentration, feed volumetric flow rate, permeate temperature, permeate velocity, length of fiber, number of fiber layers, etc. Inlet permeate temperature is not chosen as decision variables, rather suitable values for it (= 35 ^{0}C) is assumed in this optimization study. A minimum approach temperature of 10 ^{0}C is maintained for Fresh Feed HE, whereas pressure drop of 20 kPa is fixed for Heater

and Fresh Feed HE (Turton et al., 2009). Further, a pressure drop of 10 kPa is used for both sides of MD module (Song et al., 2007). Table 9.2 provides details on two MOO problems considered in this study.

Table 9.2: Objectives and decision variables for different MOO problems

Objective functions	**Decision variables and their ranges**
A. max. water production rate [lit/h] and min. energy consumption [KW]	$50 < T_{f1}$ (feed temperature) $< 85\ ^{0}C$ $150 < c_{f1}$ (feed concentration) < 200 g/lit $10 < V_{f1}$ (feed volumetric flow rate) < 50 lit/min
B. max. water production rate [lit/day] and min. brine disposal [kg/day]	$5 < v_{p1}$ (permeate velocity) < 40 m/min $12 <$ NOF (number of fiber layers) < 16 $0.2 <$ LOF (length of each fiber) < 0.4 m

9.4 Results and Discussion

In this work, Excel based I-MODE algorithm is used to generate the set of non-dominated solutions (see Chapter 3). Values of algorithm parameters are taken based on the recommendation in Chapter 3: TR = 0.01, $\delta_{GD} = 0.0003$, $\delta_{SP} = 0.1$, N = 50 and MNG = 50. Visual Basic for Applications is used to interface I-MODE algorithm with ACM and Aspen Plus. I-MODE algorithm provides decision variable vector; and ACM simulation is called for the set of decision variables and other inlet variables. Results from ACM simulation are used in Aspen Plus to simulate the remaining parts of the process flow-sheet. Finally, results from ACM and Aspen Plus simulations are used to calculate values of objective functions.

9.4.1 Trade-off between Water Production Rate and Energy Consumption

The Pareto-optimal front obtained between water production rate and energy consumption is shown in Figure 9.3(a). This Pareto-optimal front is obtained after 50 generations, and the non-dominated solutions obtained are well distributed over the Pareto-optimal front. As expected, required energy increases with increase in the water production rate (Figure 9.3a), which is varying between 0.8-22.4 lit/h (or maximum of 537 liters water/day). Trend in the Pareto-optimal front can be visually correlated to the decision variables: T_{f1}, v_{p1}, NOF and LOF (see Figures 9.3a-b, 9.3d-f). V_{f1} is near to its upper bound except some scattered points (Figure 9.3c), which could be improved by running the algorithm for larger number of generations.

Initial increase in the water production rate from 0.8 to 4 kg/h is mainly due to increase in feed temperature (T_{f1}); v_{p1} and NOF are nearly constant at their respective lower bounds. Variation in v_{p1} for the remaining range of the water production rate (i.e., 4-22.4 kg/h) shows similar trend as the obtained Pareto-optimal front; here, NOF is also following some trend with the water production rate, whereas T_{f1} is nearly constant at its upper bound (i.e., 85 ^{0}C). Further, c_{f1} remains constant near to its lower bound (i.e., 150 g/lit); lower salt concentration in feed is beneficial for producing large amount of water with relatively low energy consumption.

As I-MODE algorithm has improvement based termination criterion, so 25th generation has been identified as the termination generation (i.e., GT). The obtained Pareto-optimal fronts after GT and MNG are comparable (see Figure 9.3h), but trends in some of decision variables are improved by more generations after GT. For brevity, Figure 9.3 does not show variations of different decision variables with water production rate at GT = 25.

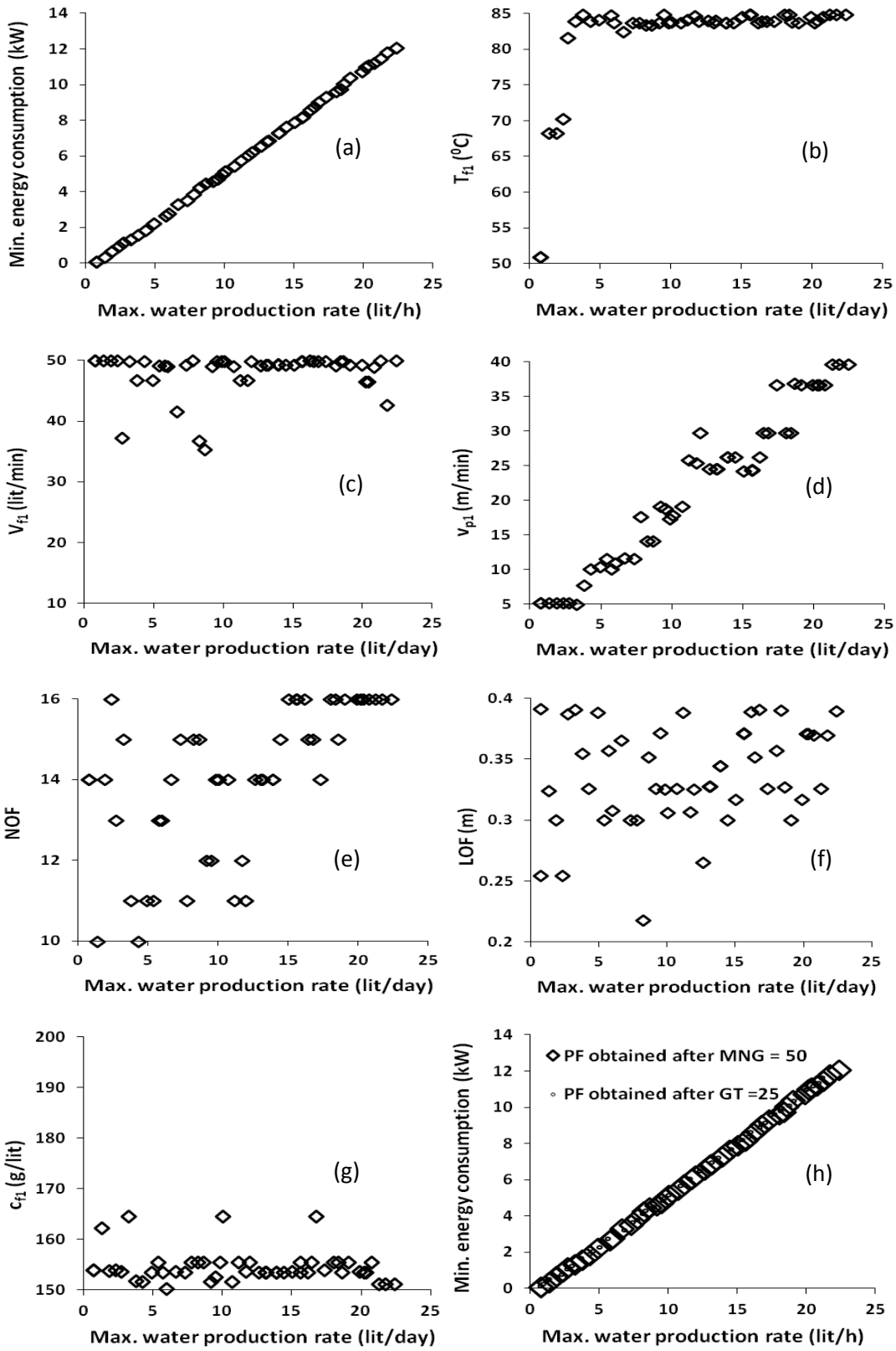


Figure 9.3: Optimization results for simultaneous maximization of water production rate and minimization of energy consumption

9.4.2 Trade-off between Water Production Rate and Brine Disposal Rate

Figure 9.4(a) shows the Pareto-optimal front obtained between water production rate and brine disposal rate, after 50 generations. As expected, brine disposal rate increases with increase in water production rate. The obtained Pareto-optimal front can be roughly divided into two parts: (i) a steep increase in the water production rate (18.7-130 lit/day), (ii) a linear change between water production rate (130-494.3 lit/day) and brine disposal rate. In the first part, the change is mainly due to changes in v_{p1} and NOF at nearly constant value of T_{f1}. In the second part, T_{f1} is mainly affecting both the objectives. Further, v_{p1} is constant at its upper bound (40 m/min) in the second part of the Pareto-optimal front, as its high value gives large water production rate. Furthermore, c_{f1} is near to its upper bound except some scattered points (Figure 9.4g); this ensures lower brine disposal rate at high salt concentration, and finally it gives high water recovery in MD system.

The trend in the obtained Pareto-optimal front can also be visually correlated to the NOF and LOF; generally water production rate is large for high values of NOF and LOF. Finally, V_{f1} is scattered between its lower and upper bounds (Figure 9.4c). The Pareto-optimal front obtained after GT (= 25) is shown in Figure 9(h), and it is close to the Pareto-optimal front obtained after MNG (= 50). Variations in different decision variables with water production rate at GT = 25 are not presented in Figure 9.4, for brevity.

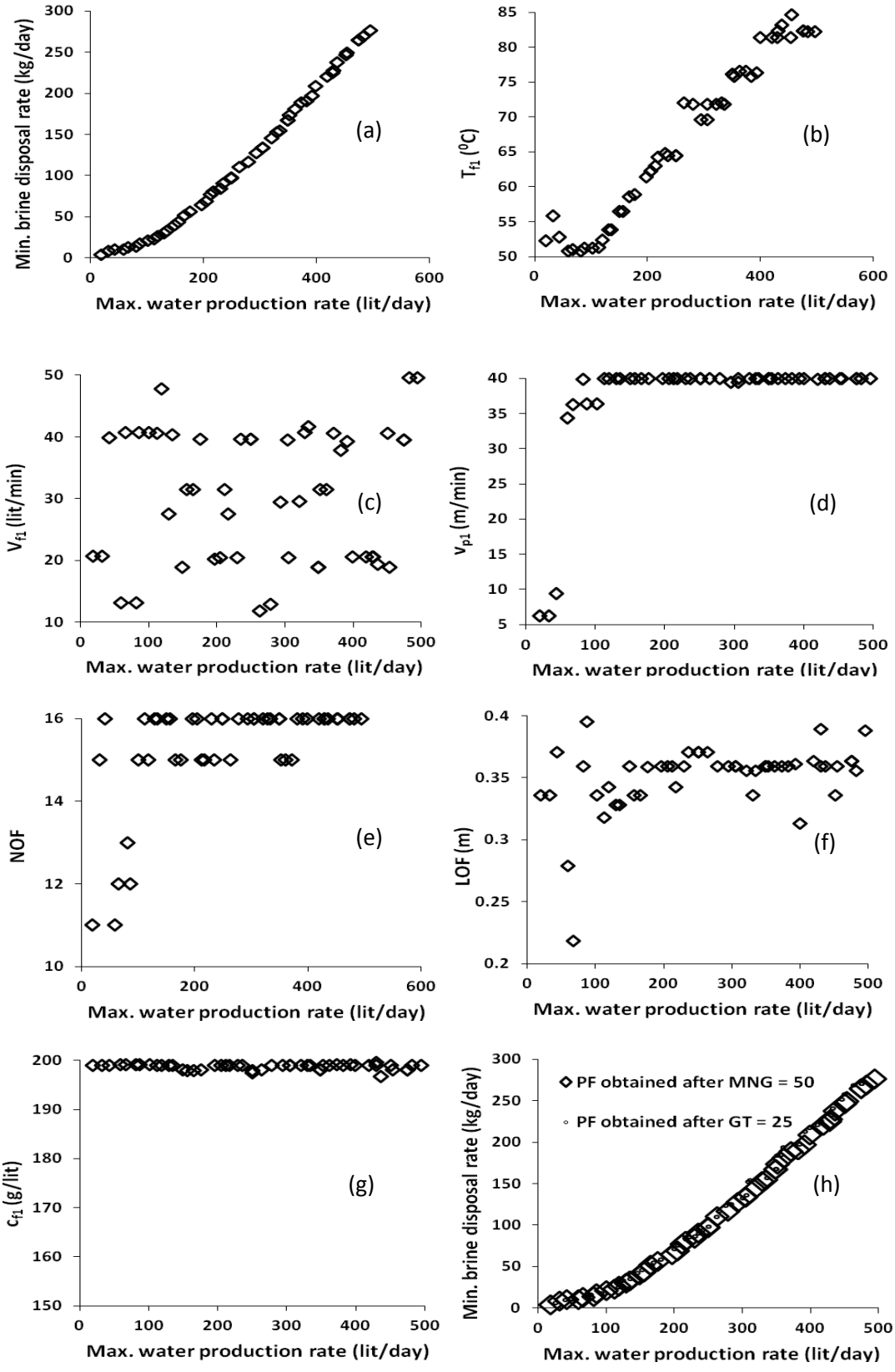


Figure 9.4: Optimization results for simultaneous maximization of water production rate and minimization of brine disposal rate

A process design with a water production rate of about 130 lit/day can be selected for implementation purpose (Figure 9.4a); it has relatively large production rate at lower brine disposal rate. In order to achieve the large water production rate, several MD modules can be placed in parallel. Trade-off between water production rate and energy consumption does not play any significant role in the process design selection (Figure 9.3a), but decision variable space can be analyzed to select one suitable process design for easier operation.

Further, the MD and crystallizer can be used to expend the existing RO plants. RO process produces brine of about 7 wt% salt concentration. MD can use the brine from RO, and it can increase brine concentration near to saturation limit (~ 36 wt% at 30 ^{0}C). The concentrated brine from MD can be used to separate salt in a crystallizer at low temperature. The integrated process can then produce drinking water at zero brine discharge.

9.5 Conclusions

This study optimizes a small-scale MD system for water production rate, energy consumption and brine disposal rate. The obtained Pareto-optimal fronts give better insights by providing a range of alternative designs. Optimal values of some decision variables follow certain trends with the water production rate. The I-MODE algorithm is able to terminate the search at the right generations, in both the optimization cases studied. In general, MOO is useful to improve the understanding and design of MD system.

Chapter 10

Conclusions and Recommendations

10.1 Conclusions of the Present Study

In this thesis, I-MODE algorithm has been developed for MOO problems. It has been tested on constrained test problems, followed by its application to optimize a number of chemical engineering applications for multiple objectives. The proposed termination criterion has been evaluated with two other algorithms. Further, a constraint handling technique has been proposed and analyzed for solving constrained MOO problems. Three new applications, related to renewable fuel and drinking water production, have been studied for important performance criteria. Major contributions and findings of this thesis are summarized below.

i. Current optimization techniques for solving MOO problems are briefly reviewed. In this work, differential evolution is used as search algorithm for I-MODE algorithm, due to its better performance in the recent CEC competitions. So, performance of different MODE algorithms is compared based on the reported studies. Further, MOO applications in chemical engineering have been reviewed. In the recent times, MOO was increasingly used in areas related energy and environmental impact such as renewable energy, power plants, fuel cells and carbon dioxide emissions, besides its continued use in process design, petroleum refining, petrochemicals, polymerization, food industry, pharmaceuticals and biotechnology.

ii. I-MODE algorithm has been developed to solve MOO application problems with high efficiency and reliability. Its key features are termination criterion, taboo list

and self adaptation of DE parameters. Several performance metrics are analyzed for their variations with generations, and then GD and SP are selected for the development of a search termination criterion, which is able to terminate the search at the right generations for many test problems. Inclusion of taboo list has improved the reliability of search in locating the optimal solutions using less computational effort. I-MODE algorithm performed better than DMOEA-DD algorithm, which was the best evolutionary algorithm on constrained optimization problems in CEC 2009 competition.

iii. The developed termination criterion is also evaluated for use in four selected jumping gene adaptations of NSGA-II algorithm. It is able to terminate the search at the right generations, for both unconstrained and constrained test problems. The quality of non-dominated solutions obtained for different problems by I-MODE is comparable to MOSADE, which has performed better than NSGA-II-RC, SPEA2 and MOPSO (Wang et al., 2010). Further, I-MODE takes less number of function evaluations (median NFE for all problems = 5335) compared to MOSADE (NFE = 25,000 for each problem). Additionally, NNC precisely refines the non-dominated solutions obtained by I-MODE, and it can also reduce variations in GD values obtained in different runs. I-MODE is able to find non-dominated solutions comparable to the known solutions of three chemical engineering application problems, in less number of generations.

iv. In order to solve constrained MOO problems, adaptive constraint relaxation and feasibility approach (ACRFA) are modified and tested on two benchmark problems. Further, the developed constraint handling approach is applied to optimize the performance of two fermentation processes with many equality constraints. It is found that ACRFA approach works better than feasibility

approach (with a user defined relaxation value). However, MODE with ACRFA (simultaneous solution approach) performs inferior compared to sequential solution approach on the studied applications. In the latter approach, numerical techniques are required to solve highly complex and non-linear process model equations, and these techniques may require good initial estimates. Conversely, simultaneous solution approach does not require any additional information.

v. In-situ ethanol removal from the fermentor increases conversion of reducible sugars to ethanol. Hence, a three-stage fermentation process integrated with cell recycling and pervaporation is modeled and optimized for ethanol productivity and xylose conversion, simultaneously. The integrated fermentation-pervaporation process performs better than integrated fermentation-extraction process. Inclusion of pervaporation with fermentor gives higher ethanol productivity, for the same amount of feed, compared to fermentation with extraction. In the case of fermentation with extraction, mother liquor from third-stage fermentor has lower ethanol concentration and larger flow rate compared to fermentation with pervaporation; hence, ethanol recovery and purification will be more expensive for fermentation with extraction. Further, I-MODE algorithm is evaluated on three different integrated fermentation processes, and it is found that non-dominated solutions obtained after termination generations are closer to the Pareto-optimal fronts obtained using maximum number of generations.

vi. Bio-diesel production using waste cooking oil is beneficial from both economic and environmental perspectives. Hence, a complete bio-diesel production plant consisting of esterification (pre-treatment) and trans-esterification, is simulated in Aspen Hysys and then optimized for relevant performance objectives. Variations in the feed availability and market demands are quite obvious, and so a process

design is selected and then studied for the variation in the feed flow rate. This operation optimization identifies key manipulated variables for achieving high process performance at different production rates. It also suggests changes required for significant increase in the process throughput.

vii. Membrane distillation (MD) can utilize waste heat and solar energy to produce drinking water from sea water. In this work, design and operation of a small MD system for domestic use is optimized for high water production rate, lower energy consumption and lower brine disposal rate. The MOO approach provides a range of alternative designs with different water production rate.

10.2 Recommendations for Future Studies

MOO approach is increasingly being applied to improve the performance of various processes in different areas. Hence, there is a need to develop efficient and reliable MOO techniques for highly complex and non-linear optimization problems. Important future studies related to this thesis are discussed below.

i. Algorithm development for large optimization problems: Although stochastic methods are successfully applied to optimize various small and medium size application problems, these become less effective on problems with a large number of decision variables (Wang, 2008). If state variable are considered as decision variables, application problems become large (e.g., fermentation processes considered in Chapter 6). Hence, there is a need for an efficient and reliable search algorithm suitable for large optimization problems.

Moreover, stochastic methods are sensitive to values of their parameters. Several researchers have adapted values of parameters for different evolutionary

algorithms, but these adaptations are often independent of search performance (Cao et al., 2007; Zhang and Sanderson, 2008). Hence, algorithm parameters need to be self-adapted based on the performance of search in recent generations.

Many MOO algorithms employ non-dominated sorting to select individuals for the subsequent generation, but this takes considerable computational time for optimization problems with many performance objectives (Guillen-Gosalbez, 2011). Hence, there is a need to develop objective reduction techniques or to modify individual selection operation.

ii. Development of hybrid search methods: Stochastic search has comprehensive exploration capability, whereas deterministic search can easily exploit local search region. Hence, a hybrid search may give both high reliability and computational efficiency for a wide range of optimization problems. Hybrid algorithms have performed better than stochastic algorithms to solve SOO problems (e.g., Zhang and Rangaiah, 2011). Several possibilities can be explored for combining the stochastic and deterministic methods for MOO problems. For example, some individuals can be selected in each generation, and then these individuals can be refined using local search. Similar refinement can also be performed at the end of stochastic search.

iii. Improvements in constraint handling techniques: Generally, practical problems involve both equality and inequality constraints due to design equations, equipment limitations and safety aspects. ACRFA approach has been found to be better than feasibility approach, but it performed inferior compared to the sequential solution approach. So, further work is required to improve ACRFA.

There are no MOO benchmark test functions with equality constraints in the literature. Hence, there is a need to develop such test functions.

Moreover, optimal solutions often exist near to the boundaries of inequality constraints. Due to stochastic nature of evolutionary algorithms, newly generated trial individual are unlikely to be located at the boundaries of inequality constraints or inside the small feasible regions for equality constraints. Hence, if possible, information obtained during the search should be utilized to repair infeasible individuals into feasible or less infeasible individuals (Harada et al., 2007).

iv. Optimization of lignocellulosic ethanol production process: The lignocellulosic ethanol production has less environmental impact, but it has higher production cost (Stephen et al., 2012). It can be reduced using cheaper feed-stocks, reducing capital and operating costs and better utilization of waste materials. The work done in this thesis (i.e., on fermentation process integrated with cell recycling and pervaporation or extraction for ethanol productivity and xylose conversion) can be extended to include up-stream and down-stream units, and then optimization of the complete lignocellulosic ethanol production process can be performed for important performance criteria. Further, different arrangements of the pervaporation and/or extraction units with fermentor can also be explored.

v. Heat integration in the bio-diesel process: In this thesis, an industrial bio-diesel plant is studied for economic and environmental objectives. This process can be improved to use different types of feed-stocks. The bio-diesel process is energy intensive, and so there is scope for heat integration among different process

streams. Further, integration of bio-ethanol and bio-diesel plants can also be studied for better economics and lower environmental burden.

vi. Optimization of membrane distillation plant: In this thesis, a small MD system for domestic use is studied for multiple objectives. This work can be extended to large-scale plants. Further, RO plants produce concentrated brine, and its disposal is expensive and creates several environmental problems. MD can operate on the brine from RO processes, and so extension of a RO plant using MD can be investigated.

References

1. Abbass, H. A., Sarker, R. and Newton, C. PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems, Proceeding of IEEE Congress on Evo. Comp., pp. 971-978, 2001.
2. Agarwal, A. and Gupta, S. K. Jumping gene adaptations of NSGA-II and their use in the multi-objective optimal design of shell and tube heat exchangers, Chem. Eng. Res. Des., *86*, pp. 132-139, 2008a.
3. Agarwal, A. and Gupta, S. K. Multiobjective optimal design of heat exchanger networks using new adaptations of the elitist non-dominated sorting genetic algorithm, NSGA-II, Ind. Eng. Chem. Res., *47*, pp. 3489-3501, 2008b.
4. Agrawal, N., Rangaiah, G. P., Ray, A. K. and Gupta, S. K. 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, 2006.
5. Agrawal, N., Rangaiah, G. P., Ray, A. K. and Gupta, S. K. Design stage optimization of an industrial low-density polyethylene tubular reactor for multiple objectives using NSGA-II and its jumping gene adaptations, Chem. Eng. Sc., *62*, pp. 2346-2365, 2007.
6. Bayrock, D. P. and Ingledew, W. M. Application of multi-stage continuous fermentation for production of fuel alcohol by very high gravity fermentation technology, J. of Ind. Microbiology and Biotechnology, *27*, pp. 87-93, 2001.
7. Bek-Pederser, E., Gani, R. and Levaux, O. Determination of optimal energy efficient separation schemes based on driving forces, Comp. and Chem. Eng., *24(2-7)*, pp. 253-259, 2000.

8. Berrios, M., Siles, J., Martin, M. A. and Martin, A. A kinetic study of the esterification of free fatty acid (FFA) in sunflower oil, Fuel, *86(15)*, pp. 2383-2388, 2007.

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

10. Bhat, S. A. 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, 2007.

11. Bhat, G. R. and Gupta, S. K. Multi-objective optimization of phthalic anhydride industrial catalytic reactors using guided GA with the adapted jumping gene operator, Chem. Eng. Res. Des., *86*, pp. 959-976, 2008.

12. Bhat, S. A., Saraf, D. N., Gupta, S. and Gupta, S. K. 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, 2006.

13. Bosman, P. A. N. and Jong, E. D. Exploiting gradient information in numerical multi-objective evolutionary optimization, Proceedings of Conf. on Genetic and Evo. Comp., pp. 755-762, 2005.

14. Brest, J., Greiner, S., Boskovic, B., Mernik, M. and Zumer, V. Self-adaptive control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Transactions on Evo. Comp., *10(6)*, pp. 646-457, 2006.

15. Canakci, M. and Van Gerpen, G. Bio-diesel production from oils and fats with high free fatty acids, Transactions of the ASAE, *44*, pp. 1429-1436, 2001.

16. Cao, R., Li, G. and Wu, Y. A self-adaptive evolutionary algorithm for multi-objective optimization, Lecture Notes in Comp. Sc., Springer, Berlin/ Heidelberg, pp. 553-564, 2007.

17. Chaudhari, P. and Gupta, S. K. Multi-objective optimization of a fixed bed maleic anhydride reactor using an improved biomimetic adaptation of NSGA-II, Ind. Eng. Chem. Res., *51*, pp. 3279-3294, 2012.

18. Chen, X. Q., Hou, Z. X. and Liu, J. X. Multi-objective optimization with modified Pareto differential evolution, IEEE Conf. on Intelligent Computation Technology and Automation, pp. 90-95, 2008.

19. Chen, M. L. and Wang, F. S. Optimization of a fed-batch simultaneous saccharification and co-fermentation process from lignocellulose to ethanol, Ind. Eng. Chem. Res., *49(12)*, pp. 5775-5785, 2010a.

20. Chen, M. L. and Wang, F. S. Optimal trade-off design of integrated fermentation processes for ethanol production using genetically engineered yeast, Chemical Engineering J., *158*, pp. 271-280, 2010b.

21. Chen, S. Q., Rangaiah, G. P. and Srinivas, M. Differential evolution: method, development and chemical engineering applications, In: Rangaiah, G. P. (editor), Stochastic global optimization: techniques and applications in chemical engineering, Word Scientific, Singapore, 2010.

22. Chua, C. B. H., Lee, H. M. and Low, J. S. C. Life cycle emissions and energy study of bio-diesel derived from waste cooking oil and diesel in Singapore, Int. J. Life Cycle Assess., *15*, pp. 417-423, 2010.

23. Coello Coello, C. A. Theoretical and numerical constraints handling techniques used with evolutionary algorithm: a survey of the state of the art, Compt. Methods Appl. Mech. Eng., *191*, pp. 1245-1287, 2002.

24. Coello Coello, C. A. and Salazar Lechuga, M. MOPSO: a proposal for multi-objective particle swarm, IEEE Congress on Evol. Comp., pp. 1051-1056, 2002.

25. Coello Coello, C. A., Lamont, G. B. and Van Veldhuizen, D. A. Evolutionary algorithms for solving multi-objective problems, 2nd ed., Springer, Berlin/Heidelberg, 2007.

26. Conneman, J. and Fischer, J. Bio-diesel in Europe 1998: bio-diesel processing technology, Int. Liquid Bio-fuels Congress, Brazil, 1998.

27. Criscuoli, A., Canevale, M. C. and Drioli, E. Evaluation of energy requirements in membrane distillation, Chem. Eng. Pro.: P.I., *47(7)*, pp. 1098-1105, 2008.

28. Das, I. and Dennis, J. Normal boundary intersection: an alternate method for generating Pareto optimal points in multi-criteria optimization problems, NASA Contractor Report 201616, 1996.

29. Daugulis, A. J., Axford, D. B. and Mclellan P. J. Economics of ethanol production by extractive fermentation, Canadian J. of Chemical Engineering, *69*, pp. 488-497, 1991.

30. Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evo. Comp., *6(2)*, pp. 182-197, 2002.

31. Deb, K. Multi-objective optimization using evolutionary algorithm, John Wiley & Sons, Chichester, UK, 2001.

32. Deb, K., Agarwal S., Pratap A. and Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II, Technical Report 2000001, IIT Kanpur, KanGAL, (http://www.iitk.ac.in/kangal/reports.shtml, access date: December 2011), 2000.

33. Dong, N. and Wang, Y. Multiobjective differential evolution based on opposite operation, IEEE Conf. on Computational Intelligence and Security, pp. 123-127, 2009.

34. Dorigo, M. and Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evo. Comp., *1(1)*, pp. 53-66, 1997.

35. Edgar, T. F., Himmelblau, D. M. and Lasdon L. S. Optimization of chemical processes, 2nd ed., McGraw-Hill, 2001.

36. Ericson, M. A., Mayer, A. and Horn, J. Multi-objective optimal design of groundwater remediation system: application of the niched Pareto genetic algorithm, Advances in Water Resource, *25(1)*, pp. 51-56, 2002.

37. Fliege, J. and Svaiter, B. F. Steepest descent methods for multi-criteria optimization, Math. Meth. Oper. Res., *51*, pp. 479-494, 2000.

38. Fliege, J., Grana Drummond, L. M. and Svaiter, B. F. Newton's method for multi-objective optimization, SIAM J. Opt., *20(2)*, pp. 602-626, 2009.

39. Fonseca, C. M. and Fleming, P. J. Genetic algorithms for multi-objective optimization: formulation, discussion and generalization, Proceeding of the 5th Int. Conf. on Genetic Algorithms, pp. 416-423, 1993.

40. Franca, B. B., Pinto, F. M., Pessoa, F. L. P. and Uller, A. M. C. Liquid-liquid equilibria for castor oil bio-diesel + glycerol + alcohol, J. Chem. Eng. Data., *54*, pp. 2359-2364, 2009.

41. Freedman, B., Butterfield, R. O. and Pryde, E. H. Trans-esterification kinetics of soybean oil, J. American Oil Chemists Society, *63*, pp. 1375-1380, 1986.

42. Furtuna, R., Curteanu, S., Racles, C. NSGA-II-RJG applied to multi-objective optimization of polymeric nanoparticles synthesis with silicone surfactants, Cent. Eur. J. Chem., *9(6)*, pp. 1080-1095, 2011.

43. Gandibleux, X., Mezdaoui, N. and Freville, A. A tabu search procedure to solve multi-objective combinatorial optimization problems, In: Caballero, R., Ruiz, F. and Steuer, R. (editors), Advances in multiple objective and goal programming, Springer Verlag, pp. 291-300, 1997.

44. Garcia, M., Gonzalo, A., Sanchez, J. L., Arauzo, J. and Pena, J. A. Prediction of normalized bio-diesel properties by simulation of multiple feed-stocks blends, Bioresource Technology, *101*, pp. 4431-4439, 2010.

45. Ghadge, S. V. and Raheman, H. Process optimization for bio-diesel production from mahua (Madhuca Indica) oil using response surface methodology, Bioresource Technology, *97*, pp. 379-384, 2006.

46. Glover, F. Future paths for integer programming and links to artificial intelligence, Comp. and Ops. Res., *13(5)*, pp. 533-549, 1986.

47. Glover, F. Tabu search - Part I, ORSA J. of Computing, *1(3)*, pp. 190-206, 1989.

48. Gong, W. and Cai, Z. An improved multiobjective differential evolution based on Pareto-adaptive ε-dominance and orthogonal design, European J. of Oper. Res., *198(2)*, pp. 576-601, 2009.

49. Guillen-Gosalbez, G. A novel MILP-based objective reduction method for multi-objective optimization: application to environmental problems, Comp. and Chem. Eng., *35(8)*, pp. 1469-1477, 2011.

50. Guria, C., Verma, M., Mehrotra, S. P. and Gupta, S. K. Multi-objective optimal synthesis and design of froth flotation circuits for mineral processing, using the

jumping gene adaptations of genetic algorithm, Ind. Eng. Chem. Res., *44*, pp. 2621-2633, 2005a.

51. Guria, C., Bhattacharya, K. and Gupta, S. K. Multi-objective optimization of reverse osmosis desalination units using different adaptations of the non-dominated sorting genetic algorithm (NSGA), Comp. and Chem. Eng., *29*, pp. 1977-1995, 2005b.
52. Guria, C., Verma, M., Mehrotra, S. P. and Gupta, S. K. 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. Mineral Processing, *79*, pp. 149-166, 2006.
53. Gyamerah, M. and Glover, J. Production of ethanol by continuous fermentation and liquid-liquid extraction, J. of Chemical Technology and Biotechnology, *66*, pp. 145-152, 1996.
54. Harada, K., Sakuma, J. and Kobayashi, S. Local search for multi-objective function optimization: Pareto descent method, Genetic and Evo. Comp. Conf., *8*, pp. 659-666, 2006.
55. Harada, K., Sakuma, J., Ono, I. and Kobayashi, S. Constraint handling method for multi-objective optimization: Pareto descent repair operator, Lecture Notes in Comp. Sc., Springer, Berlin/ Heidelberg, pp. 156-170, 2007.
56. Hasan, M. P. and Jaszkiewicz, A. Evaluating the quality of approximations to the non-dominated set, Technical Report IMM-REP, Technical University Denmark, 1998.
57. Hass, M. J., McAloon, A. J., Yee, W. C. and Foglia, T. A. A process model to estimate bio-diesel production costs, Bioresource Technology, *97*, pp. 671-678, 2006.

58. Heerema, L., Roelands, M., Goetheer, E., Verdoes, D. and Keurentjes, J. In-situ product removal from fermentations by membrane extraction: conceptual process design and economics, Ind. Eng. Chem. Res., *50(15)*, pp. 9197-9208, 2011

59. Holland, J. H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence, Ann Arbor, MI: University of Michigan Press, 1975.

60. Horn, J., Nafpliotis, N. and Goldverg, D. E. A niched Pareto genetic algorithm for multi-objective optimization, Proceeding of the 1st IEEE Conf. on Evo. Comp., pp. 82-87, 1994.

61. Huang, V. L., Suganthan, P. N., Qin, A. K. and Baskar, A. Multi-objective optimization based on self-adaptive differential evolution algorithm, CEC Special Session on the Performance Assessment of real-parameter MOEAs, 2007.

62. Huang, V. L., Zhao, S. Z., Mallipedi, R. and Suganthan, P. N. Multi-objective optimization using self-adaptive differential evolution algorithm, IEEE Congress on Evo. Comp., pp. 191-195, 2009.

63. Ji, S. F., Sheng, W. X., Jing, Z. W. and Cheng, L. G. IMODE: improving multi-objective differential evolution algorithm, Int. Conf. of Natural Computation, pp. 212-216, 2008.

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

65. Katahira, S., Ito, M., Takema, H., Fujita, Y., Tanino, T., Tanaka, T., Fukuda, H. and Kondo, A. Improvement of ethanol productivity during xylose and glucose

co-fermentation by xylose-assimilating S. Cerevisiae via expression of glucose transporter sut1, Enzyme and Microbial Technology, *43(2)*, pp. 115-119, 2008.

66. Kennedy, J. and Eberhart, R. C. Particle swarm optimization, IEEE Int. Conf. Neural Networks, pp. 1942-1948, 1995.
67. Khayet, M. S. and Matsuura, T. Membrane distillation: principles and applications, Elsevier, 2010.
68. Khosla, D. K., Gupta, S. K. and Saraf, D. N. Multi-objective optimization of fuel oil blending using the jumping gene adaptations of genetic algorithm, Fuel Processing Technology, *88*, pp. 51-63, 2007.
69. Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. Optimization by simulated annealing, Science, *220*, pp. 671-680, 1983.
70. Krishnan, M. S., Xia, Y., Ho, N. W. Y. and Tsao, G. T. Fermentation kinetics of ethanol production from glucose and xylose by recombinant saccharomyces 1400 (pLNH33), Applied Biochemistry and Biotechnology, *77/79*, pp. 373-388, 1999.
71. Kukkonen, S. and Lampinen, J. Comparison of generalized differential evolution algorithm to other multi-objective evolutionary algorithms, Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering, Jyvaskyla, Finland, 2004a.
72. Kukkonen, S. and Lampinen, J. An extension of generalized differential evolution for multi-objective optimization with constraints, Conf. on Parallel Problem Solving from Nature, *752-761*, pp. 1943-1950, 2004b.
73. Kukkonen, S. and Lampinen, J. Performance assessment of generalized differential evolution 3 (GDE3) with a given set of problems, IEEE Congress on Evo. Comp., pp. 3593-3600, 2007.

74. Kukkonen, S. and Lampinen, J. Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems, IEEE Congress on Evo. Comp., pp. 1943-1950, 2009.

75. Kundu, P. K., Ray, A. K. and Elkamel, A. Numerical simulation and optimization of unconventional three section simulated countercurrent moving bed chromatographic reactor for oxidative coupling of methane reaction, Canadian J. Chem. Eng., In Press, 2011.

76. Lee, E. S. Q. and Rangaiah, G. P. Optimization of recovery processes for multiple economic and environmental objectives, Ind. Eng. Chem. Res., *48(16)*, pp. 7662-7681, 2009.

77. Lee, E. S. Q., Rangaiah, G. P. and Agrawal, N. Optimal design of chemical processes for multiple economic and environmental objectives, In: Rangaiah G. P. (editor), Multi-objective optimization: techniques and applications in chemical engineering, World Scientific, Singapore, 2009.

78. Li, H., Jiao, Y. C. and Zhang, L. Hybrid differential evolution with a simplified quadratic approximation for constrained optimization problems, Engineering Optimization, *43(2)*, pp. 115-134, 2010.

79. Li, K., Fialho, A. and Kwong, S. Multi-objective differential evolution with adaptive control of parameters and operators, Lecture Notes in Comp. Sc., Springer, Berlin/ Heidelberg, pp. 473-487, 2011.

80. Li, K., Zheng, J., Zhou, C. and Lv, H. An improved differential evolution for multi-objective optimization, Proceedings of World Congress on Computer Science and Information Engineering, 2008.

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

82. Lipnizki, F., Hausmanns, G., Laufenberg, G., Field, R. and Kunz, B. Use of pervaporation bioreactor hybrid processes in biotechnology, Chemical Engineering and Technology, *23(7)*, pp. 569-577, 2000.

83. Liu, M., Zou, X., Chen, Y. and Wu, Z. Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances, CEC Special Session on the Performance Assessment of Multi-objective Optimization Algorithms, 2009.

84. Lotero, E., Liu, Y. J., Lopez, D. E., Suwannakarn, K., Bruce, D. A. and Goodwin, J. G. Synthesis of bio-diesel via acid catalysis, Ind. Eng. Chem. Res., *44*, pp. 5353-5363, 2005.

85. Luus, R. Optimization of system with multiple objective functions, International Congress, European Federation of Chemical Engineering, Paris, pp. 3-8, 1978.

86. Madavan, N. K. Multi-objective optimization using Pareto differential evolution approach, IEEE Congress on Evo. Comp., pp. 1145-1150, 2002.

87. Mahecha-Botero, A., Garhyan, P. and Elnashaie, S. S. E. H. Non-linear characteristics of a membrane fermentor for ethanol production and their implications, Non-linear Analysis: Real World Application, *7*, pp. 432-457, 2006.

88. Mariano, C. E. and Morales, E. A multiple objective ant-Q algorithm for the design of water distribution irrigation networks, Technical Report HC-9904, Mexican Institute of Technology of Water, 1999.

89. Marti, L., Garcia, J., Berlanga, A. and Molina, J. M. An approach to stopping criteria for multi-objective optimization evolutionary algorithms: the MGBM criterion, IEEE Congress on Evo. Comp., pp. 1263-1270, 2009.

90. Masuduzzaman and Rangaiah, G. P. Multi-objective optimization applications in chemical engineering, In: Rangaiah, G. P. (editor), Multi-objective optimization: techniques and applications in chemical engineering, World Scientific, Singapore, 2009.

91. Mazumder, J., Zhu, J. and Ray, A. K. Optimal design of liquid-solid circulating fluidized bed for continuous protein recovery, Powder Tech., *199*, pp. 32-47, 2010.

92. Messac, A., Ismail-Yahaya, A. and Mattson, C. A. The normalized normal constraint method for generating Pareto frontier, Structural and Multidisciplinary Optimization, *25(2)*, pp. 86-98, 2003.

93. Miettinen, K. Non-linear multi-objective optimization, Kluwer Academic Publishers, Boston, 1999.

94. Mitra, K., Deb, K. and Gupta, S. K. Multi-objective dynamic optimization of an industrial nylon 6 semi batch reactor using genetic algorithm, J. Appl. Poly. Sci., *69*, pp. 69-87, 1998.

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

96. Moore, J. and Chapman, R. Application of particle swarm to multi-objective optimization, Department of Computer Science and Software Engineering, Auburn University, 1999.

97. Morais, S., Couto, S., Martins, A. A. and Mata, T. M. Designing eco-efficient bio-diesel production processes from waste vegetable oils, 20th European Symposium on Computer Aided Process Engineering, pp. 253-258, 2010.

98. Mori, Y. and Inaba, T. Ethanol production from starch in a pervaporation membrane bioreactor using clostridium thermo-hydosulfuricum, Biotechnology and Bioengineering, *36*, pp. 849-859, 1990.

99. Myint, L. L. and El-Halwagi, M. M. Process analysis and optimization of bio-diesel production from soybean oil, Clean Techno. Environ. Policy, *11*, pp. 263-276, 2009.

100. Nabavi, S. R., Rangaiah, G. P., Niaei, A. and Salari, D. Multi-objective optimization of an industrial LPG thermal cracker using a first principles model, Ind. Eng. Chem. Res., *48*, pp. 9523-9533, 2009.

101. Nandong, J., Samyudia, Y. and Tade, M. O. Dynamic simulation and optimization of two-stage extraction alcoholic fermentation process: design impact on controllability, Chemical Product and Process Modeling, *1(1)*, 2006.

102. Nicola, G. D., Moglie, M., Pacetti, M. and Santori, G. Bioenergy II: modeling and multi-objective optimization of different bio-diesel production processes, Int. J. Chemical Reactor Engineering, *8(1)*, 2010.

103. Noureddini, H. and Zhu, D. Kinetics of trans-esterification of soybean oil, Biocatalysis articles, *74(11)*, pp. 1457-1463, 1997.

104. Offeman, R. D., Stephenson, S. K., Robertson, G. H. and Orts, W. J. Solvent extraction of ethanol from aqueous solutions I: screening methodology for solvents, Ind. Eng. Chem. Res., *44*, pp. 6789-679, 2005.

105. Oliveira, S. C., Castro, H. F. D., Visconti, A. E. S. and Giudici, R. Continuous ethanol fermentation in a tower reactor with flocculating yeast recycle: scale-up

effects on process performance, kinetic parameters and model predictions, Bioprocess Engineering, *20*, pp. 157-160, 1999.

106. Omran, M. G. H., Engelbrecht, A. P. and Salman, A. Bare bones differential evolution, European J. of Oper. Res., *196*, pp. 128-139, 2009.
107. Paiva, T. C. B., Sato, S., Visconti, A. E. S. and Castro, L. A. B. Continuous alcoholic fermentation process in a tower reactor with recycling of flocculating yeast, Applied Biochemistry and Biotechnology, *57/58 (1)*, pp. 535-541, 1996.
108. Park, S. and Lee, J. Improvement of a multi-objective Differential Evolution using Clustering Algorithm, Int. Sym. on Indust. Elect., pp. 1213-1217, 2009.
109. Ponsich, A., Azzaro-Pantel, C., Domenech, S. and Pibouleau, L. Constraint handling strategies in genetic algorithms application to optimal batch plant design, Chem. Eng. and Processing, *47*, pp. 420-434, 2008.
110. Price, K. V., Storn, R. M. and Lampinen, J. A. Differential evolution - a practical approach to global optimization, Springer, Berlin/Heidelberg, 2005.
111. Qian, W. and Li, A. Adaptive differential evolution algorithm for multiobjective optimization problems, Applied Mathematics and Computation, *201*, pp. 431-440, 2008.
112. Qian, F., Xu, B., Qi, R. and Tianfield, H. Self-adaptive differential evolution algorithm with α-constrained-domination principle for constrained multi-objective optimization, Soft Computing - A Fusion of Foundations, Methodologies and Applications, *16(8)*, pp. 1353-1372, 2012.
113. Qin, H., Zhou, J., Li, Y. and Lu, Y. Enhanced strength Pareto differential evolution (ESPDE): an extension of differential evolution for multi-objective optimization, Int. Conf. of Natural Computation, pp. 191-196, 2008.

114. Qin, A. K., Huang, V. L. and Suganthan, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Transactions on Evo. Comp., *13(2)*, pp. 398-417, 2009.

115. Qu, B. Y. and Suganthan, P. N. Multi-objective differential evolution with diversity enhancement, J. Zhejiang University - Science C, *11(7)*, pp. 538-543, 2010.

116. Qu, B. Y. and Suganthan, P. N. Multi-objective differential evolution based on the summation of normalized objectives and improved selection method, IEEE Symposium on Differential Evolution, pp. 1-8, 2011.

117. Rahnamayan, S., Tizhoosh, H. R. and Salama, M. M. A. Opposition versus randomness in soft computing techniques, Applied Soft Computing, *8(2)*, pp. 906-918, 2008.

118. Ramteke, M. and Gupta, S. K. Multi-objective optimization of an industrial Nylon-6 semi batch reactor using the a-jumping gene adaptations of genetic algorithm and simulated annealing, Polym. Eng. Sci., *48*, pp. 2198-2215, 2008.

119. Ramteke, M. and Gupta, S. K. Multi-objective genetic algorithm and simulated annealing with jumping gene adaptations, In: Rangaiah G. P. (editor), Multi-objective optimization: techniques and applications in chemical engineering, World Scientific, Singapore, 2009a.

120. Ramteke, M. and Gupta, S. K. Biomimicking altruistic behavior of honey bees in multi-objective genetic algorithm, Ind. Eng. Chem. Res., *48*, pp. 9671-9685, 2009b.

121. Ramteke, M. and Gupta, S. K. Biomimetic adaptations of the evolutionary algorithm, NSGA-II-aJG, using the biogenetic law of embryology for intelligent optimization, Ind. Eng. Chem. Res., *48*, pp. 8054-8067, 2009c.

122. Rangaiah, G. P. Multi-objective optimization: techniques and applications in chemical engineering, World Scientific, Singapore, 2009a.

123. Rangaiah, G. P. Introduction, In: Rangaiah G. P. (editor), Multi-objective optimization: techniques and applications in chemical engineering, World Scientific, Singapore, 2009b.

124. Ravindran, A., Raqsdell, K. M. and Reklaitis, G. V. Engineering optimization: methods and applications, 2nd ed., John Wiley & Sons, 2006.

125. Ripon, K. S. N., Kwong, S., Man, K. F. A real-coding jumping gene genetic algorithm (RJGGA) for multiobjective optimization, Information Science, *117*, pp. 632-654, 2007.

126. Rudenko, O. and Schoenauer, M., A steady performance stopping criterion for Pareto-based evolutionary algorithm, Proceedings of the 6th Int. Multi-Objective Programming and Goal Programming, 2004.

127. Sand, G., Till, J., Tometzki, T., Urselmann, M., Engell, S. and Emmerich, M. Engineered versus standard evolutionary algorithms: a case study in batch scheduling with recourse, Comp. and Chem. Eng., *32*, pp. 2706-2722, 2008.

128. Sander, U. and Soukup, P. Design and operation of a pervaporation plant for ethanol dehydration, J. of Membrane Science, *36*, pp. 463-475, 1988.

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

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

131. Sankararao, B. and Gupta, S. K. Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using two jumping gene

adaptations of simulated annealing, Comp. and Chem. Eng., *31*, pp. 1282-1295, 2007b.

132. Sarkar, D. and Modak, J. M. Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using non-dominated sorting genetic algorithm, Chem. Eng. Sc., *60*, pp. 481-492, 2005.
133. Sauer, R. N., Colville, J. A. R. and Burwick, C. W. Computer points the way to more profits, Hydrocarbon Processing and Petroleum Refiner, *43*, pp. 84-92, 1964.
134. Schaffer, J. D. Multiple objective optimizations with vector evaluated genetic algorithms, Proceeding of the 1st Int. Conf. on Genetic Algorithms, pp. 93-100, 1985.
135. Serafini, P. Simulated annealing for multi-objective optimization problems, Multiple Criteria Decision Making, pp. 283-292, 1994.
136. Silva, F. L. H. D., Rodrigues, M. I. and Maugeri, F. Dynamic modeling, simulation and optimization of an extractive continuous alcoholic fermentation process, J. of Chemical Technology and Biotechnology, *74*, pp. 176-182, 1999.
137. Sindhya, K., Deb, K. and Miettinen, K. A local search based evolutionary multi-objective optimization approach for fast and accurate convergence, Lecture Notes in Comp. Sc., Springer, Berlin/ Heidelberg, pp. 815-824, 2008.
138. Sindhya, K., Deb, K. and Miettinen, K. Improving convergence of evolutionary multi-objective optimization with local search: a concurrent-hybrid algorithm, Nat. Comput., *10*, pp. 1407-1430, 2011.
139. Song, L., Li, B., Sirkar, K. K. and Gilron, J. L. Direct contact membrane distillation based desalination: novel membranes, devices, large scale studies, and a model, Ind. Eng. Chem. Res., *46*, pp. 2307-2323, 2007.

140. Srinivas, N. and Deb, K. Multi-objective optimization using non-dominated sorting in genetic algorithm, Evolutionary Computation, *2(3)*, pp. 221-248, 1994

141. Srinivas, M. and Rangaiah, G. P. Differential evolution with taboo list for solving nonlinear and mixed-integer nonlinear programming problems, Ind. Eng. Chem. Res., *46*, pp. 7126-7135, 2007.

142. Stephen, J. D., Mabee, W. E. and Saddler. Will second-generation ethanol be able to compete with first-generation ethanol? Opportunities for cost reduction, Bio., Biop. and Bior., *6(2)*, pp. 159-176, 2012.

143. Storn, R. and Price, K. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ICSI, March 1995.

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

145. Suganthan, P. N. (presenter). Report on performance assessment of multi-objective optimization algorithms, CEC Special Session on the Performance Assessment of real-parameter MOEAs, 2007.

146. Takahama, T. and Sakai, S. Constrained optimization by the ε-constrained differential evolution with gradient based mutation and feasible elites, IEEE Congress Evo. Comp., pp. 1-8, 2006.

147. Tarafder, A., Rangaiah, G. P. and Ray, A. K. Multi-objective optimization of an industrial styrene monomer manufacturing process., Chem. Eng. Sc., *60*, pp. 347-363, 2005.

148. Thibault, J. Net flow and rough set: two methods for ranking the Pareto domain. In: Rangaiah, G.P. (editor), Multi-objective optimization: techniques and applications in chemical engineering, World Scientific, Singapore, 2009.

149. Trautmann, H., Ligges, U., Mehnen, J. and Preuss, M. A convergence criterion for multi-objective evolutionary algorithms based on systematic statistical testing, Lecture Notes in Comp. Sc., Springer, Berlin/ Heidelberg, pp. 825-836, 2008.

150. Turton, R., Bailie, R. C., Whiting, W. B. and Shaeiwitz, J. A. Analysis, synthesis and design of chemical processes, 3rd ed., Prentice Hall, 2009.

151. Van Gerpen, J., Shanks, B., Pruszko, R., Clements, D. and Knothe, G. Bio-diesel production technology, NREL/SR-510-36244, (www.nrel.gov/docs/ fy04osti/36244.pdf, access date: March, 2011), 2004.

152. Van Le, T. A fuzzy evolutionary approach to constrained optimization problems, IEEE Congress on Evo. Comp., pp. 274-278, 1995.

153. Van Veldhuizen, D. A. and Lamont, G. B. Evolutionary computation and convergence to a Pareto front, (citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.42.7224, access date: December 2011), 1998.

154. Villafafila, A. and Mujtaba, I. M. Fresh water by reverse osmosis based desalination: simulation and optimization, Desalination, *155(1)*, pp. 1-13, 2003.

155. Wagner, T. and Trautmann, H. Online convergence detection for evolutionary multi-objective algorithms revisited, IEEE Congress on Evo. Comp., pp.1-8, 2010.

156. Wagner, T., Trautmann, H. and Naujoks, B. OCD: online convergence detection for evolutionary multi-objective algorithms based on statistical testing, Lecture Notes in Comp. Sc., Springer, Berlin/ Heidelberg, pp. 198-215, 2009.

157. Walsh, T. J. and Bungay, H. R. Shallow depth sedimentation of yeast cells, Biotechnology and Bioengineering, *21*, pp. 1081-1084, 1990.

158. Wang, Y. Investigation on large scale global optimization with noise-induced, Report for IEEE CIS Walter Karplus Summer Research Project: Large Scale Optimization with Noise-Induced, 2008.

159. Wang, Y. N., Wu, L. H. and Yuan, X. F. Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy based diversity measure, Soft Computing, *14*, pp. 193-209, 2010.

160. Wang, F. S. and Lin, H. T. Fuzzy optimization of continuous fermentations with cell recycling for ethanol production, Ind. Eng. Chem. Res., *49(5)*, pp. 2306-2311, 2010.

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

162. West, A. H., Posarac, D. and Ellis, N. Assessment of four bio-diesel production process using Hysys, Bioresource Technology, *99*, pp. 6587-6601, 2008.

163. Wieczorek, A. and Michalski, H. Continuous ethanol production by flocculating yeast in the fluidized bed bioreactor, FEMS Microbiology Reviews, *14(1)*, pp. 69-74, 1994.

164. Xue, F., Sanderson, A. C. and Graves, R. J. Pareto based multi-objective evolution, IEEE Congress on Evo. Comp., pp. 862-869, 2003.

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

166. Yingling, B., Li, C., Honglin, W., Xiwen, Y. and Zongchen, Y. Mutli-objective optimization of bio-ethanol production during cold enzyme starch hydrolysis in

very high gravity cassava mash, Bioresource Technology, *102(17)*, pp. 8077-8084, 2011.

167. Zamuda, A. Differential evolution for multi-objective optimization with self-adaptation, CEC Special Session on the Performance Assessment of real-parameter MOEAs, 2007.
168. Zhang, C., Rangaiah, G. P. and Kariwala, V. Design and plant wide control of a bio-diesel plant, In: Rangaiah, G. P. and Kariwala, V. (editors), Plant-wide control: recent developments and applications, Wiley, 2012.
169. Zhang, H. and Rangaiah, G. P. A hybrid global optimization algorithm and its application to parameter estimation problems, Asia Pacific J. of Chem. Eng., *6*, pp. 379-390, 2011.
170. Zhang, H. and Rangaiah, G. P. An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization, Comp. and Chem. Eng., *37*, pp. 74-88, 2012.
171. Zhang, J. and Sanderson, A. C. Self-adaptive multi-objective differential evolution with the directional information provided by archived inferior solutions, IEEE Congress on Evo. Comp., pp. 2801-2810, 2008.
172. Zhang, Q., Zhou, A., Zhano, S., Suganthan, P. N., Liu, W. and Tiwari, S. Multi-objective optimization test instances for the CEC 2009 special session and competition, CEC Special Session on the Performance Assessment of Multi-objective Optimization Algorithms, 2009.
173. Zhang, Y., Dube, M. A., McLean, D. D. and Kates, M. Bio-diesel production from waste cooking oil-1: process design and technological assessment, Bioresource Technology, *89*, pp. 1-16, 2003a.

174. Zhang, Y., Dube, M. A., McLean, D. D. and Kates, M. Bio-diesel production from waste cooking oil-2: economic assessment and sensitivity analysis, Bioresource Technology, *90*, pp. 229-240, 2003b.

175. Zhong, J. H. and Zhang, J. Adaptive multi-objective differential evolution with stochastic coding strategy, Proceeding of 13th Annual Conf. on Genetic and Evo. Comp., pp. 665-672, 2011.

176. Zhou, A., Jin, Y. and Zhang, Q. Combining model-based and generics-based offspring generation for multi-objective optimization using a convergence criterion, IEEE Congress on Evo. Comp., pp. 3234-3241, 2006.

177. Zielinski, K. and Laur, R. Differential evolution with adaptive parameter setting for multi-objective optimization, CEC Special Session on the Performance Assessment of real-parameter MOEAs, 2007.

178. Zitzler, E. and Thiele, L. Multi-objective optimization using evolutionary algorithms: a comparative case study, Parallel Problem Solving from Nature, pp. 292-301, 1998.

179. Zitzler, E. and Thiele, L. Multi-objective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evo. Comp., *3(4)*, 257-271, 1999.

180. Zitzler, E., Deb, K. and Thiele, L. Comparison of multi-objective evolutionary algorithms: empirical results, Evolutionary Computation, *8(2)*, pp. 173-195, 2000.

181. Zitzler, E., Laumanns, M. and Thiele, M. SPEA2: improving the strength Pareto evolutionary algorithm, TIK-Report 103, (http://citeseerx.ist .psu.edu/viewdoc/summary?doi=10.1.1.112.5073, access date: December 2011), 2001.

182. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and Fonseca, V. G. Performance assessment of multi-objective optimizers: an analysis and review, IEEE Transactions on Evo. Comp., *7(2)*, pp. 117-132, 2003.

183. Zuo, G., Wang, R., Field, R. and Fane, A. G. Energy efficiency evaluation and economic analyses of direct contact membrane distillation system using Aspen Plus, Desalination, *283*, pp. 237-244, 2011.

Publications

Shivom Sharma has received his Bachelor's degree in 2005 from R. B. S. College, Agra, affiliated to U. P. Technical University, Lucknow (India). He has finished his Master's degree in Chemical Engineering from I.I.T. Roorkee (India) and T.U. Berlin (Germany) in 2007. After that, he has worked at the University of Petroleum and Energy Studies, Dehradun (India), for almost a year. He has been working towards his Ph.D. degree since 2008, in the Department of Chemical & Biomolecular Engineering, NUS, Singapore. His research interests include multi-objective optimization methods, differential evolution and their applications. Publications during his Ph.D. are listed below:

A. Journal publications in international journals

1. Sharma, S., Rangaiah, G. P. and Chua, Y. C. (2011), Economic and environmental criteria and trade-Offs for recovery processes, Material and Manufacturing Processes, 26, pp. 431-445.
2. Sharma, S., Rangaiah, G. P. and Cheah, K. S. (2012), Multi-objective optimization using MS Excel with an application to design a falling film evaporator system, Food and Bioproducts Processing, 90(2), pp. 123-134.
3. Sharma, S. and Rangaiah, G. P. (2012), Modeling and optimization of a fermentation process integrated with cell recycling and pervaporation for multiple objectives, Ind. Eng. Chem. Res., 51(15), pp. 5542-5551.
4. Sharma, S. and Rangaiah, G. P. (2012), Multi-objective optimization of a bio-diesel production process, Fuel, 103, pp. 269-277.

5. Sharma, S. and Rangaiah, G. P. (2013), An improved multi-objective differential evolution with a termination criterion for constrained optimization problems, Computers and Chemical Engineering, under review.

B. Book chapters

1. Sharma, S. and Rangaiah, G. P. (2013), Multi-objective optimization applications in chemical engineering, In: Rangaiah G. P. and Bonilla-Petriciolet, A. (editors), Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.
2. Sharma, S., Nabavi, S. R. and Rangaiah, G. P. (2013), Performance comparison of jumping gene adaptations of elitist non-dominated sorting genetic algorithm, In: Rangaiah G. P. and Bonilla-Petriciolet, A. (editors), Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.
3. Sharma, S. and Rangaiah, G. P. (2013), Improved constraints handling technique for multi-objective optimization with application to two fermentation processes, In: Rangaiah G. P. and Bonilla-Petriciolet, A. (editors), Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.
4. Sharma, S., Lim, Z. C., and Rangaiah, G. P. (2013), Process design for economic, environmental and safety objectives with an application to cumene process, In: Rangaiah G. P. and Bonilla-Petriciolet, A. (editors), Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.

5. Bonilla-Petriciolet, A., Sharma, S., and Rangaiah, G. P. (2013), Phase equilibrium data reconciliation using multi-objective differential evolution, In: Rangaiah G. P. and Bonilla-Petriciolet, A. (editors), Multi-objective Optimization in Chemical Engineering: Developments and Applications, Wiley.
6. Sharma, S. and Rangaiah, G. P. (2012), Application of hybrid multi-objective optimization approach to solve process engineering problems, under preparation.

C. International/regional conferences

1. Sharma, S. and Rangaiah, G. P. (2010), A hybrid multi-objective optimization algorithm, 5th International Symposium on Design, Operation and Control of Chemical Processes (PSE Asia), Singapore.
2. Sharma, S. and Rangaiah, G. P. (2010), Multi-objective optimization of a bio-diesel production plant, 13th Asia Pacific Confederation of Chemical Engineering Congress (APCChE), Taiwan.
3. Rangaiah, G. P., Zhang, H. and Sharma S. (2012), Stochastic global optimization techniques: progress, challenges and prospects for chemical engineering applications, 14th Asia Pacific Confederation of Chemical Engineering Congress (APCChE), Singapore.
4. Sharma, S. and Rangaiah, G. P. (2012), Multi-objective optimization of a membrane distillation system for desalination of sea water, European Symposium on Computer Aided Chemical Engineering (ESCAPE 22), London.

5. Sharma, S. and Rangaiah, G. P. (2012), Multi-objective optimization of a fermentation process integrated with cell recycling and inter-stage extraction, 11th Symposium on Process System Engineering (PSE 2012), Singapore.

D. Student competition

1. Gim Hoe, O. and Sharma, S. (2012), Design of a biomass power plant, Winner of EURECHA Student Contest 2012 (ESCAPE-22, London).