NEW STRATEGIES FOR GLOBAL OPTIMIZATION OF CHEMICAL ENGINEERING APPLICATIONS BY DIFFERENTIAL EVOLUTION

HAIBO ZHANG

NATIONAL UNIVERSITY OF SINGAPORE

2012

NEW STRATEGIES FOR GLOBAL OPTIMIZATION OF CHEMICAL ENGINEERING APPLICATIONS BY DIFFERENTIAL EVOLUTION ZHANG HAIBO 2012

NEW STRATEGIES FOR GLOBAL OPTIMIZATION OF CHEMICAL ENGINEERING APPLICATIONS BY DIFFERENTIAL EVOLUTION

HAIBO ZHANG

(*B.Tech. (Hons), National University of Singapore*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

To My Parents for Their Inspiration

&

My Wife, Xiang Ying, for Her Constant Support

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. G. P. Rangaiah for his invaluable guidance, continuous support, close supervision and encouragement throughout the period of this work. I would also like to thank him for teaching me the important things: positive attitude, innovative thinking and critical analysis while solving problems during my research work. Besides the above aspects related to work, I will never forget all the moments that I spent with Prof. Rangaiah during the informal get-togethers and dinners that he has arranged for our research group.

I would like to thank my thesis panel members, Assoc. Prof. K. Hidajat and Assistant Prof. D.Y. Lee for teaching me the fundamentals and advanced topics in Chemical Engineering. I received several constructive and helpful comments from them on my thesis proposal. The majority of these comments have been incorporated or addressed in this thesis. I would also like to thank Assoc. Prof. S. B. Chen, Assoc. Prof. M. S. Chiu, Prof. I. A. Karimi, Assoc. Prof. S. Lakshminarayanan and other professors in the Chemical and Biomolecular Engineering Department who have contributed directly or indirectly to this thesis. I would like to thank technical and non-technical staff in the department for their assistance on several laboratory related issues. I would like to thank Prof. A. Bonilla-Petriciolet and Mr. J. A. Fernandez-Vargas from the University of Guanajuato, Mexico, for the collaborations and contributions in some of my publications.

I am thankful to S. Sharma for the thought-provoking discussions on research-related issues. Thanks also to Devid, Luo Kui, Hong Feng, Zhi Wei, Kai Tun, Sathvik, Kai Lun and Zi Long, who Prof. Rangaiah gave me the opportunity to work with for

their final year projects. I would like to thank Xiong Peng from Department of Electrical and Computer Engineering in NUS who helped me in some software issues. Finally, thanks to my lab-mates: Elaine, Suraj, Krishna, Vaibhav, Naviyn, Sumit, Bhargava, Wendou and Shruti for their camaraderie and support that I value.

I am very thankful to my parents who have always been supporting and encouraging me, to overcome the difficulties encountered in all these years. Thanks to my sisters for their everlasting love and generous support. My deepest appreciation to my wife, Xiang Ying, for her endless love, support, patience and understanding and being always by my side.

Finally, I would like to profusely thank NUS for providing the opportunity and funding for my doctoral research.

Table of contents

Summary

Nowadays, optimization is a necessity in almost every field such as business, science and engineering. In real life, most of the optimization problems are highly nonlinear and non-convex. The traditional optimization techniques can be easily trapped at a local optimum. So, global optimization becomes more and more important since it can overcome this difficulty and can find the global optimum. However, there are still many challenges in developing reliable, robust and efficient global optimization methods and using these techniques to solve the difficult and complex application problems. Therefore, a study of global optimization methods and their applications is important and necessary. This thesis focuses on the development of a stochastic global optimization technique with novel strategies for termination and constraints handling, and its application to chemical engineering problems.

First, an overview of various global optimization algorithms together with their categories, advantages and working principles is provided. Then, global optimization applications in thermodynamics, namely, phase equilibrium modeling, calculations and stability analysis, are reviewed. Next, an integrated differential evolution algorithm (IDE) is developed. It combines parameter self-adaption, tabu list, new stopping criterion and local search. The effectiveness of IDE is demonstrated on different sets of benchmark problems and by comparison with the latest DE techniques in the literature. Subsequently, IDE is used to solve many different parameter estimation problems in vapor-liquid equilibrium modeling and in nonlinear dynamic systems. Further, performance of IDE for phase equilibrium and stability problems is studied and compared with other global optimization algorithms.

Many application problems involve equality and inequality constraints. Hence, a new constraint handling method is developed for handling equality and/or inequality constrained problems. It utilizes self-adaptive relaxation of constraints and the feasibility approach for selection. IDE with the proposed constraint handling method is tested for solving benchmark problems and chemical engineering applications with equality and/or inequality constraints. The results show that the proposed constraint handling method is reliable and efficient for solving constrained optimization problems. The pooling problem is an important optimization problem that has not been studied using stochastic global optimization algorithms. Hence, the constraint handling method with IDE is applied to solve the pooling problems. The performance comparison with the recent results by deterministic methods shows that our algorithm is a good alternative method for solving the pooling problems.

Finally, IDE algorithm is modified to handle both discrete and continuous variables. In addition, one-step approach for solving heat exchanger network (HEN) retrofit problems by this modified IDE is proposed. In this approach, HEN structure (integer variables) and retrofitting model parameters (continuous variables) are simultaneously optimized, which not only avoids the algorithm trapping at a local optimum but can also improve the computational efficiency. The performance of the modified IDE algorithm and the proposed one-step approach is compared with the reported state-of-the-art methods for HEN retrofit problems. This shows that our approach is efficient and robust for global optimization of HEN retrofit problems.

Nomenclature

Abbreviation	**Explanation**
ACO	Ant Colony Optimization
ARS	Adaptive Random Search
BARON	Branch and Reduce Optimization Navigator
BBPSO	Bare-Bones PSO
CF	Cash Flow
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
DE	Differential Evolution
DETL	DE with Tabu List
EC	Equality Constraint
EIV	Error-In-Variable
ES	Easom function
FCI	Fixed Capital Investment
GA	Genetic Algorithm
GAMS	General Algebraic Modeling System
GP	Goldstein and Price function
GRG	Generalized Reduced Gradient
GSR	Global Success Rate
GW	Griewank function
H3	Hartmann 3 function
HEN	Heat Exchanger Network
HS	Harmony Search
IDE	Inequality Constraint

IDE_N	Local Exploration-based DE
IEC	Integrates Differential Evolution
LEDE	IDE without Tabu List and Radius
LP	Learning Period
LS	Least Squares
MDSA	Modified Version of Direct Search SA
mHB	Modified Himmelbalu function
MINLP	Mixed Integer Nonlinear Problems
nb	Number of benchmark functions
NFE	Number of Function Evaluations
NPW	Net Present Worth
NR	Number of Rejections
NRTL	Non-Random-Two-Liquid
PBP	Pay-Back Period
PBT	Profit Before Texes
PEC	Phase Equilibrium Calculation
PS	Phase Stability
PSO	Particle Swarm Optimization
rPEC	Reactive PEC
RAS	Rastrigin function
ROS	Rosenbrock function
RTA	Random Tunneling Algorithm
UBBPSO	Unified BBPSO
UNIQUAC	Universal Quasi-Chemical
SaDE	Self-adaptive DE

SA	Simulated Annealing
SC-1	Stopping Criterion 1
SC-2	Stopping Criterion 2
SDE	Shubert function
SH	Stochastic Differential Equations
SQP	Sequential Quadratic Program
SR	Success Rate
SSE	Sum of Squared Errors
TAC	Total Annual Cost
TAV	Total Absolute Violation
TLS	Tabu List Size
TS	Tabu (or taboo) Search
TR	Tabu (or taboo) Radius
TPDF	Tangent Plane Distance Function
VBA	Visual Basic Application
VFSA	Very Fast SA
VLE	Vapor-Liquid Equilibrium
WO	William-Otto process
ZAK	Zakharov function

Symbol	**Explanation**
A	Additional area of retrofitted heat exchanger
A^{L}_{i}	Minimum availability of the i^{th} input raw material
A^{U}_{i}	Maximum availability of the i^{th} input raw material
B_j	Capacity of the j^{th} pool
c	Number of Components for VLE Problems

C_{ij}	Unit cost of the i^{th} input stream into j^{th} pool
Cr	Crossover rate
D	Dimension/ number of variables
$D^{L}{}_{k}$	Minimum demand of the k^{th} product
$D^{U}{}_{k}$	Maximum demand of the k^{th} product
F	Mutation/scaling factor
F_{obj}	Objective function
G	Generations
G_{max}	Maximum number of generations
L	Total number of component qualities
m	Number of dependent variables for VLE problem
N_j	Number of input streams entering pool, j
NP	Population size
n	Number of moles
n_i^k	Total number of moles in component i in phase k
nc	Number of componemts
$ndat$	Number of experiments for VLE problem
$nest$	Number of state variables for VLE problem
$npar$	Number of parameters for VLE models (NRTL, Wilson and UNIQUAC models)
P	System pressure, or Total number of pools
R	Total number of end products
SC_{max}	Maximum number of successive generations without improvement in the best function value
S_k	Unit selling price of the k^{th} product

T	Temperature
U	Trial vector
V	Mutated vector
x	Decision variables, or mole fraction in liquid phase
X	Target vector
X_{ij}	Flow rate of i^{th} input stream entering into the j^{th} pool
y	Mole fraction in vapor phase
Y_{jk}	Flow rate from j^{th} pool to the k^{th} product (Intermediate stream)
z_i	Mole of i^{th} component in the feed
Z	Quality requirement of the product

Greek Letters	**Explanation**
α	Constant / parameter in HEN
β_i	i^{th} component of the decision variables
γ_i	Activity coefficient of component i/ parameter in HEN
θ	Energy interaction parameter vector, or Decision variable in parameter estimation problems
λ	Magnitude of perturbation/ qualities in pooling problem
б	Standard deviation
ρ	Fraction of feasible search space

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides a general introduction to global optimization techniques, categories, challenges, applications and motivation for this study.

1.1 Global Optimization

Nowadays, optimization is a necessity in almost every field such as business, science and engineering. In every area, some quantitative optimization techniques are required in order to improve the performance of applications and processes. To achieve this goal, we need to have a mathematical model for the application and an objective function which depends on decision variables that are subject to relevant conditions or constraints. Most of the optimization problems in real life are nonlinear and non-convex in nature, and so optimization of such problems should find a global optimum rather than a local optimum. However, classical optimization techniques have difficulties in finding the global optimal solution since they can easily be trapped in local minima. Moreover, they cannot generate or even use the global information needed to find the global minimum of a problem with multiple local minima. The global optimization techniques can overcome the disadvantages of the classical optimization techniques. They try to find the values of decision variables to optimize the objective function globally and not just locally.

Interest in global optimization has increased in the last 10-20 years in order to develop effective algorithms for finding global optimal solutions for different kinds of optimization problems. Global optimization refers to finding the best (either maximum or minimum) value of a given non-convex function in the specified feasible region. Some optimization problems involve finding the maximum of an objective function such as profit, production rate, etc. whereas others involve finding the minimum of an objective such as cost, processing time, etc. Often, optimization methods are described for minimization. If the problem is for maximization, it can be transformed to minimization by simply negating the objective function.

A typical global optimization problem features an objective function, equality/inequality constraints and upper/lower bounds on decision variables.

$$\begin{aligned} &\text{Minimize } f(x) \\ &\text{Subject to } h_i(x) = 0, i = 1, 2, ..., m1 \\ &\qquad\qquad g_j(x) \le 0, j = 1, 2, ..., m2 \\ &\qquad\qquad x_k^l \le x_k \le x_k^u, k = 1, 2, ..., n \end{aligned} \tag{1.1}$$

Here, x_k is an n-dimensional vector of decision variables, $f(x)$ is an objective function, $h_i(x) = 0$ and $g_j(x) \le 0$ are respectively $m1$ equality and $m2$ inequality constraints, and x_k^l and x_k^u are respectively the lower and upper bounds of x_k.

1.2 Classification of Global Optimization Techniques

There are many global optimization techniques available currently. However, global optimization is still challenging. Available global optimization methods can be classified in two broad categories (Pardalos et al., 2000; Liberti and Kucherenko, 2005): deterministic and stochastic (or probabilistic) global optimization methods. The commonly used methods are classified and shown in Figure 1.1.

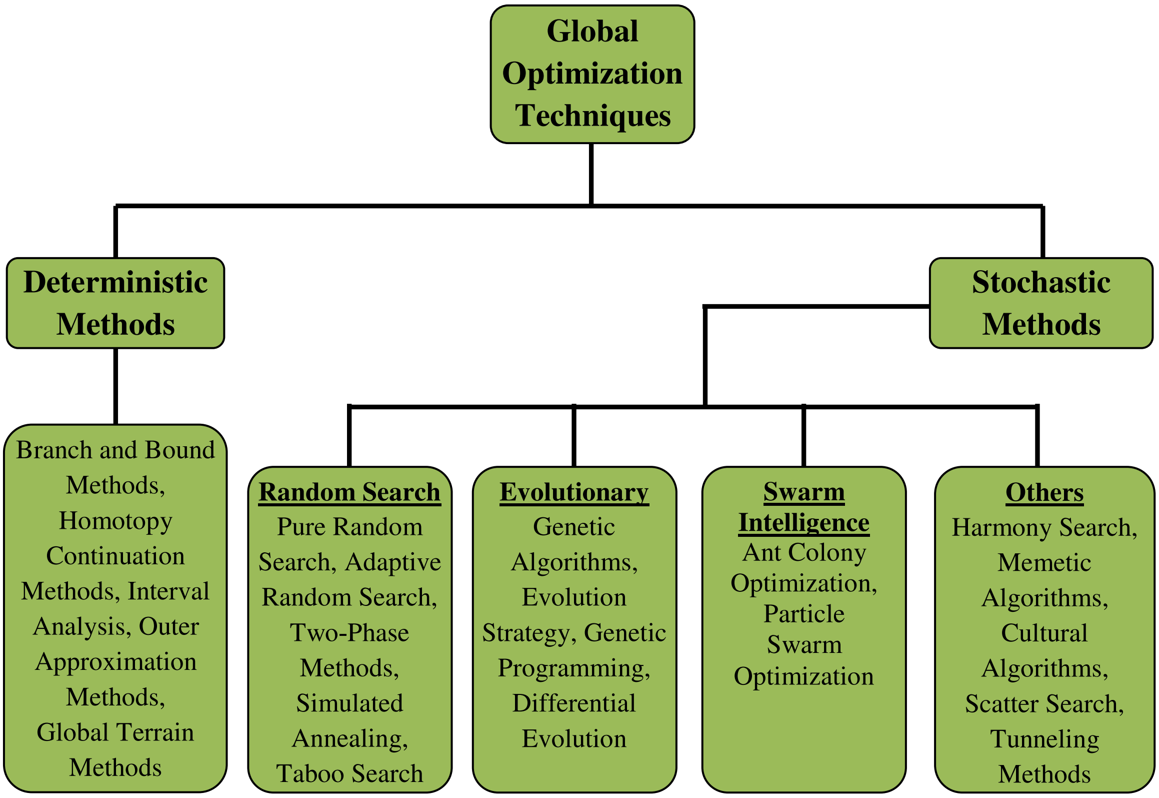

Global Optimization Techniques
Deterministic Methods
Stochastic Methods
Branch and Bound Methods, Homotopy Continuation Methods, Interval Analysis, Outer Approximation Methods, Global Terrain Methods
Random Search
Pure Random Search, Adaptive Random Search, Two-Phase Methods, Simulated Annealing, Taboo Search
Evolutionary
Genetic Algorithms, Evolution Strategy, Genetic Programming, Differential Evolution
Swarm Intelligence
Ant Colony Optimization, Particle Swarm Optimization
Others
Harmony Search, Memetic Algorithms, Cultural Algorithms, Scatter Search, Tunneling Methods

Figure 1.1: Classification of Global Optimization Methods

Deterministic methods include branch and bound methods, homotopy continuation methods, interval analysis, outer approximation methods, global terrain, etc. They are most often used for specific problems and when the relation between the characteristics of the possible solutions and the problem is known (Nocedal and Wright, 1999; Weise, 2008). Deterministic methods can guarantee the global optimality of the final solution under certain conditions such as continuity and convexity. However, no algorithm could solve general global optimization problems with certainty in finite time (Guus et al. 1995; Moles et al., 2003). If the relation between a solution candidate and its “fitness” is not so obvious or too complicated, or the dimensionality of the search space is very high, the global optimization problem becomes harder to solve using deterministic methods (Weise, 2008). For mixed integer nonlinear problems (MINLP), some deterministic methods require solving a relaxed problem or they solve a sequence of NLP with fixed integer values (Exler et al., 2008). In fact, although several classes of deterministic methods (e.g., branch and bound) have sound theoretical convergence properties, the associated computational effort increases very rapidly (often exponentially) with the problem size (Moles et al., 2003).

The well-known stochastic global optimization methods include genetic algorithm, evolutionary strategy, simulated annealing, differential evolution, tabu search, ant colony optimization, particle swarm optimization and scatter search. The recent book by Rangaiah (2010) covers these and their applications in chemical engineering. The most challenging global optimization problems are those without any known structure that can be used, so-called black-box optimization problems (Pardalos et al., 2000; Exler et al., 2008). Stochastic optimization algorithms, whose search is random, are designed to deal with such black-box optimization problems or highly complex optimization problems. They generally require little or no additional assumptions on the optimization problem, are simple to implement and use, and do not require transformation of the original problem. These characteristics are especially useful if the researcher has to link the optimizer with a simulator such as Aspen Plus and Hysys. On the other hand, stochastic algorithms require infinite number of iterations to guarantee global optimality, but they can locate the global optimum with high probability in modest computation times (Moles et al., 2003; Lin and Miller, 2003). Therefore, this thesis focuses on the development and applications of the stochastic global optimization algorithms.

1.3 Motivation and Scope of Work

There are many stochastic global optimization methods which have been developed and applied to application problems in many areas. However, there are still challenges in reliably and efficiently solving global optimization problems by stochastic techniques. These include:

1. Tuning of the algorithm parameters
2. Overcoming the premature condition
3. Balancing the exploration (global search) and exploitation (local search)
4. Lack of good stopping criteria
5. Effective constraint handling methods

Therefore, one of the focuses of the present research is on developing a more reliable, robust and efficient stochastic algorithm for global optimization. The other important issues considered are the parameter estimation in models, phase equilibrium and stability calculations, and pooling problems. Finally, a new mixed-integer nonlinear programming with novel approach for solving heat exchanger network retrofit problems is also proposed as part of this thesis. The motivation for studying these issues, together with relevant background information, is briefly discussed in this section.

1.3.1 An IDE with a Novel Stopping Criterion

The limitation of the global optimization algorithms has been listed in Section 1.3. The proposed algorithm (IDE) integrates differential evolution (DE) with taboo list of taboo search and parameter adaptation. The taboo list/check prevents revisiting the same area, thus increasing the population diversity and computational efficiency. The parameter adaptation strategy reduces the algorithm parameters to be provided and makes the algorithm more robust. Furthermore, a novel stopping criterion based on the number of rejected points is developed, and a local search is employed after the global search for finding the global optimum accurately and efficiently. The effectiveness of the proposed stopping criterion and IDE is assessed on more than 30 benchmark problems with 2 to 30 variables. The performance of IDE is compared with state-of-the-art global optimization algorithms in the literature.

1.3.2 Global Optimization of Parameter Estimation Problems

Parameter estimation is essentially an optimization problem where the unknown values of the parameters in the model are obtained by minimizing a suitable objective function. It plays an important role in developing better mathematical models which can be used to understand and analyze systems. Parameter estimation in thermodynamic models as well as dynamic models have been of great interest in chemical engineering due to its complex nature such as non-linearity, flat objective function in the neighborhood of global optimum, badly scaled model and non-differential term(s) in the equations. In this thesis, IDE with the proposed stopping criterion and local search is used to solve the parameter estimation problems for modeling vapor-liquid equilibrium (VLE) data and chemical engineering applications involving dynamic models. The performance of IDE for benchmark functions and VLE modeling is compared with that of other stochastic algorithms such as DE, DE with tabu list, particle swarm optimization, simulated annealing and a deterministic algorithm, Branch and Reduce Optimization Navigator (BARON).

1.3.3 Global Optimization of Phase Equilibrium and Stability Problems

Phase equilibrium calculations and phase stability analysis play a significant role in the simulation, design and optimization of separation processes in chemical engineering. These are very challenging problems due to the high non-linearity of thermodynamic models. In this study, we introduce two global optimization algorithms developed by our group for phase and chemical equilibrium calculations, namely, IDE and IDE without tabu list and radius (IDE_N), which have fewer parameters to be tuned. The performance of these stochastic algorithms is tested and compared in order to identify their relative strengths for phase equilibrium and phase stability problems. The phase equilibrium problems include both without and with chemical reactions.

1.3.4 Novel Constraint Handling Method

Constrained optimization problems are very important as they are encountered in many engineering applications. Equality constraints in them are challenging to handle due to tiny feasible region. Additionally, global optimization is required for finding global optimum when the objective and constraints are nonlinear. Stochastic global optimization methods can handle non-differentiable and multi-modal objective functions. In this work, a new constraint handling method for use with such methods is proposed for solving equality and/or inequality constrained problems. It incorporates adaptive relaxation of constraints and the feasibility approach for selection. The IDE with this constraint handling technique is tested for solving challenging constrained benchmark problems, and then applied to many chemical engineering application problems with equality and/or inequality constraints.

1.3.5 Global Optimization of Pooling Problems

The pooling problems are the important optimization problems that are frequently encountered in the petroleum refining industries, and they often have multiple optimum. Therefore, pooling problems require a reliable and easy-to-implement optimization method to find the global optimal solution. Recently, many deterministic optimization algorithms have been applied to pooling problems. To the best of our knowledge, the performance of stochastic global optimization algorithms for solving the complex pooling problems has not been reported. In this thesis, IDE with the proposed constraint handling method is applied to solve many pooling problems, and its performance results are compared with those of deterministic methods.

1.3.6 Heat Exchanger Network Retrofitting Using IDE

Heat exchanger network (HEN) synthesis has been a hot topic in the past several decades. HEN retrofitting is more important and challenging than HEN synthesis since it involves the retrofitting existing HEN for improved energy efficiency. Additional factors to be taken into account include spatial constraints, relocation and re-piping costs, reassignment and effective use of existing heat exchangers (Rezaei and Shafiei, 2009). HEN retrofitting is gaining importance in chemical process industries as one of the most effective ways to decrease energy consumption in the current plants. It is generally formulated as a MINLP superstructure model, which contains both discrete and continuous variables. The MINLP model of HEN retrofitting is NP-hard which makes it difficult for deterministic optimization methods, especially for larger size problems (Furman and Sahinidis, 2001). The previous studies using stochastic global optimization algorithms are mainly focused on two-level approach: the first level uses stochastic algorithm for the structure change (discrete variables), and the second level uses either stochastic or deterministic algorithm for optimizing the continuous variables (Rezaei and Shafiei, 2009; Bochenek, and Jezowski, 2010). In this study, we propose one-step approach, where IDE algorithm developed above handles both discrete and continuous variable together. Thus, HEN structure and retrofitting model parameters are simultaneously optimized, which not only avoids the algorithm trapping at a local optimum but also can improve the computational efficiency. Application of the one-step approach using IDE to HEN retrofitting is tested on several examples.

1.4 Organization of the Thesis

This thesis comprises nine chapters. The next chapter presents an overview of both deterministic and stochastic global optimization techniques together with their applications in phase equilibrium modeling and calculations. Development of IDE algorithm along with a novel stopping criterion based on the number of rejection points, and its evaluation are presented in Chapter 3. Application of IDE to solve the parameter estimation in chemical engineering applications is described in Chapter 4. The evaluation of IDE algorithm for solving phase equilibrium and stability problems is presented in Chapter 5. The subsequent chapter presents a novel constraint handling method which uses self-adaptive relaxation method with feasibility approach for constrained global optimization. The first attempt to solve the pooling problems with a large number of equality and inequality constraints using a stochastic global optimization is presented in Chapter 7. Next, modified IDE is developed to handle both continuous and discrete variables, and applied for solving HEN retrofitting problems by one-step approach in Chapter 8. The conclusions and recommendations for future works are finally outlined in the last chapter. Note that Chapters 2 to 8 are based on published journal papers or submitted manuscripts, which are edited in order to minimize repetition. However, some material in Chapters 2 to 8 was repeated with the sole intention of making the concerned chapters easier to follow.

Chapter 2

Literature Review *

2.1 Introduction

The phase equilibrium modeling for multi-component systems is essential in the design, operation, optimization and control of separation schemes. Novel processes handle complex mixtures, severe operating conditions, or even incorporate multi-functional unit operations (e.g., reactive distillation and extractive distillation). Therefore, phase behavior of multi-component systems has significant impact on process design including equipment and energy costs of separation and purification strategies (Wakeham and Stateva, 2004). Phase equilibrium calculations are usually executed thousands of times in process simulators, and are especially important in chemical, petroleum, petrochemical, pharmaceutical and other process industries where separation units are the core of process performance. Hence, these calculations must be performed reliably and efficiently, to avoid uncertainties and errors in process design.

Global optimization problems abound in the modelling and analysis of phase equilibrium for both reactive and non-reactive systems. Specifically, several thermodynamic calculations can be formulated as global optimization problems, and they include three applications: a) phase stability analysis, b) Gibbs free energy minimization and c) estimation of parameters in thermodynamic models. Formally, the optimization problems of these applications can be stated as follows: minimize $F_{obj}(\boldsymbol{u})$ subject to $h_j(\boldsymbol{u}) = 0$ for $j = 1, 2, \ldots, m$ and $\boldsymbol{u} \in \Omega$ where $\boldsymbol{u}$ is a vector of n

* This chapter is based on the paper: Zhang, H., Bonilla-Petriciolet, A. and Rangaiah, G.P., A review on global optimization methods for phase equilibrium modeling and calculations. *The Open Thermodynamics J.,* vol. 5, pp.71-92, 2011.

continuous variables in the domain $\Omega \in \Re^n$, m is the number of equality constraints arising from the specific thermodynamic application, and $F_{obj}(\boldsymbol{u}) : \Omega \Rightarrow \Re$ is a real-valued function. The domain Ω is defined by the upper and lower limits of each decision variable.

The major challenge of solving global optimization problems for phase equilibrium modeling and analysis is because $F_{obj}(\boldsymbol{u})$ is generally non-convex and highly non-linear with many decision variables. Thus, the objective functions involved in phase equilibrium modeling and calculations may have several local optima including trivial and non-physical solutions especially for multi-component and multi-phase systems. Therefore, traditional optimization methods are not suitable for solving these thermodynamic problems because they are prone to severe computational difficulties and may fail to converge to the correct solution when good initial estimates are not available (Teh and Rangaiah, 2002; Wakeham and Stateva, 2004). In general, finding the global optimum is more challenging than finding a local optimum, and the location of this global optimum for phase equilibrium problems is crucial because only it corresponds to the correct and desirable solution (Floudas, 1999; Wakeham and Stateva, 2004).

The development and evaluation of global optimization methods had played and continue to play a major role for modeling the phase behavior of multi-component systems (Floudas, 1999; Teh and Rangaiah, 2002; Wakeham and Stateva, 2004). Until now, many deterministic and stochastic global optimization methods have been used for phase equilibrium calculations and modeling. Studies on the use of deterministic methods for phase equilibrium problems have been focused on the application of branch and bound optimization, homotopy continuation method and interval-Newton/generalized bisection algorithm. The stochastic optimization techniques applied for solving phase equilibrium problems include point-to-point, population-based and hybrid stochastic methods.

There have been significant developments in global optimization and their applications to phase equilibrium problems. But, to the best of our knowledge, there is no review in the literature that focuses on the global optimization methods for phase equilibrium modeling and calculations. Therefore, use of both deterministic and stochastic global optimization methods to solve phase equilibrium problems in multi-component systems is reviewed in this chapter. In particular, we focus on applications of global optimization for phase stability analysis, Gibbs free energy minimization in both reactive and non-reactive systems, and parameter estimation in phase equilibrium models. The performance and capabilities of many global optimization methods for these thermodynamic calculations are discussed. The remainder of this review is organized as follows. The formulation of optimization problems for phase equilibrium modeling and calculations is presented in Section 2.2. In Section 2.3, we briefly describe the deterministic and stochastic optimization methods used for solving the optimization problems outlined in Section 2.2. Section 2.4 reviews the phase equilibrium modeling and calculations using global optimization algorithms. Finally, concluding remarks are given in Section 2.5.

2.2 Phase Equilibrium Modeling and Calculations

This section introduces the basic concepts and description of phase equilibrium problems considered in this review. Specifically, a brief description of the global optimization problems including the objective function, decision variables and constraints, for phase stability, physical and chemical equilibrium, and phase equilibrium modeling is given in the following sections.

2.2.1 Phase Stability

Phase stability analysis is a fundamental step in phase equilibrium calculations. This analysis allows identification of the thermodynamic state that corresponds to the global minimum of Gibbs free energy (globally stable equilibrium). Additionally, the results of stability analysis can be used to begin phase-split calculations. According to the Gibbs criterion, a mixture at a fixed temperature *T*, pressure *P* and overall composition is stable if and only if the Gibbs free energy surface is at no point below the tangent plane to the surface at the given overall composition (Michelsen, 1982; Wakeham and Stateva, 2004). This statement is a necessary and sufficient condition for global stability. Generally, stability analysis is performed using the tangent plane distance function (TPDF). So, the phase stability of a non-reactive mixture with *c* components and overall composition in mole fraction units, at constant *P* and *T*, requires the global minimization of TPDF. Physically, TPDF is the vertical distance between the Gibbs free energy surface and the tangent plane constructed to this surface. For more details on the explanation, derivation and implications of TPDF, see the work of Michelsen (1982).

To perform stability analysis, TPDF must be globally minimized with respect to a trial composition *y* subject to an equality constraint and bounds on decision variables. The decision variables in phase stability problems are the mole fractions. If the global minimum of $TPDF(y) < 0$, the mixture under analysis is unstable; else, it is a globally stable system. Note that the constrained problem can be transformed into an unconstrained problem by using new decision variables β_i instead of y_i as the decision vector (Rangaiah, 2001; Srinivas and Rangaiah, 2007a and 2007b). As an alternative to the optimization procedure, phase stability can also be determined by finding all solutions of the stationary conditions of TPDF. If TPDF at any of the solutions obtained by this set of equations is negative, then the given mixture is unstable and will exhibit phase-split. Note that the trivial solution is always present in this thermodynamic problem, and it corresponds to the global minimum of TPDF for the case of stable mixtures. As suggested by Michelsen (1982), the stability criterion is also applicable to chemically equilibrated phases, and consequently almost any method proposed for stability analysis of non-reactive systems can be extended to reactive mixtures.

2.2.2 Phase Equilibrium Calculations

In the phase equilibrium/split calculations, main objectives are to establish the correct number and types of phases at equilibrium as well as the composition and quantity of each phase (Wakeham and Stateva, 2004). At constant temperature *T* and pressure *P*, a *c* multi-component and π multi-phase non-reactive system achieves equilibrium when its Gibbs free energy is at the global minimum. There are two main approaches for performing phase equilibrium calculations: a) equation solving approach and b) Gibbs free energy minimization approach (Teh and Rangaiah, 2002). The former involves solving a set of non-linear equations arising from mass balances and equilibrium relationships, whereas the latter involves the direct minimization of Gibbs free energy function. Although the first approach seems to be faster and simple, the solution obtained may not correspond to the global minimum of free energy function. Moreover, it needs a priori knowledge of phases existing at equilibrium (Teh and Rangaiah, 2002). Therefore, minimization of Gibbs free energy is a natural approach for calculating the equilibrium state of a mixture.

In a non-reactive system with *c* components and π phases, the thermodynamic function for phase equilibrium calculations is expressed as a linear combination of the chemical potential of each component in each phase. The expression for Gibbs free energy and its mathematical properties depend on the thermodynamic equation(s) chosen to model each of the phases that may exist at equilibrium. For a non-reactive system, Gibbs free energy must be minimized with respect to the set of decision variables subject to mass balance constraints. One can use new variables instead of original decision variables in the above optimization problem. Introduction of the new variables eliminates the restrictions imposed by material balances, reduces problem dimensionality and the optimization problem is transformed to an unconstrained one. For multi-phase non-reactive systems, real variables $\beta_{ij} \in (0, 1)$ are defined and employed as new decision variables. For Gibbs energy minimization, the number of phases existing at the equilibrium is usually assumed to be known *a priori*, and the number of decision variables in the unconstrained approach is $c\ (\pi - 1)$ for non-reactive systems.

Reactive phase equilibrium calculations, also known as chemical equilibrium, are performed if a reaction is possible in the system under study. Note that reactions increase the complexity and dimensionality of phase equilibrium problems, and so phase split calculations in reactive systems are more challenging due to non-linear interactions among phases and reactions. The phase distribution and composition at equilibrium of a reactive mixture are determined by the global minimization of Gibbs free energy subject to mass balances and chemical equilibrium constraints. Based on the handling of material balance constraints, available strategies can be classified as either stoichiometric or non-stoichiometric (Stateva and Wakeham, 1997).

For reactive phase equilibrium, the mass balance restrictions and non-negativity requirements are usually formulated using the conservation of chemical elements in the components (Seider and Widagdo, 1996). Therefore, to determine the phase equilibrium compositions in reactive systems using this approach, it is necessary to solve the constrained global optimization problem. The constrained global minimization of Gibbs free energy is with respect to $c\pi$ decision variables. For modeling reactive systems, the chemical equilibrium condition can be evaluated from either Gibbs free energy data or chemical equilibrium constants determined experimentally. Accordingly, we can use different objective functions for the constrained minimization of Gibbs energy function. In addition, this thermodynamic problem can be also formulated using transformed composition variables. For more details on different objective functions using both conventional and transformed composition variables as the decision vector for Gibbs free energy minimization in reactive systems, see the recent study by Bonilla-Petriciolet et al. (2011).

In particular, the constrained Gibbs free energy minimization using conventional composition variables is better in terms of computer time and numerical implementation, for reactive phase equilibrium calculations (Bonilla-Petriciolet et al., 2011). For a c multi-component and π multi-phase system subject to r independent chemical reactions, the objective function for reactive phase equilibrium calculations can be defined, using reaction equilibrium constants. The constrained global optimization problem can be solved by minimizing Gibbs free energy with respect to $c\ (\pi - 1) + r$ decision variables. In this formulation, the mass balance equations are rearranged to reduce the number of decision variables of the optimization problem and to eliminate equality constraints. For more details on the development of equations, see the recent study of Bonilla-Petriciolet et al. (2011).

2.2.3 Phase Equilibrium Modeling

The estimation of parameters in thermodynamic models is an important requirement and a common task in many areas of chemical engineering because these models form the basis for synthesis, design, optimization and control of process systems. In the case of separation processes, thermodynamic models play a major role with respect to energy requirements, phase equilibrium and equipment sizing. The parameter estimation problem refers to determining values of model parameters that provide the best fit to a set of measured data such as vapor-liquid or liquid-liquid equilibrium. In particular, estimation of parameters in non-linear thermodynamic models for vapor-liquid equilibrium (VLE) modeling has been of great interest in the chemical engineering literature. VLE data modeling using thermodynamic equations is generally based on classical least squares or maximum likelihood approaches (Englezos and Kalogerakis, 2001).

Consider a set of observations q_{ij} of $i = 1, \ldots, nd$ dependent/response variables from $j = 1, \ldots, ne$ experiments are available for the system, where the responses can be expressed by an explicit model $q_{ij} = f_i\ (\ \boldsymbol{r}_j,\ \boldsymbol{\theta}\ )$, with independent variables $\boldsymbol{r}_j = (r_{1,j},\ldots,r_{nd,j})^T$ and $npar$ parameters $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_{npar})^T$. Measurement errors in $\boldsymbol{r}_j$ can either be treated or neglected; depending on this choice, we can have either least squares (when errors in independent variable are neglected) or maximum likelihood formulation (when independent variables have measurement errors). For modeling VLE data (i.e., x–y–P at constant T, or x–y–T at constant P), excess Gibbs energy equations are widely employed. Therefore, the objective function commonly used for VLE data fitting is based on activity coefficients. Thus, the global minimization of LS objective function can be done as an unconstrained optimization problem using local composition models. On the other hand, if we assume that there are measurement errors in all the variables (which include both independent and response variables) for the experiments of the system to be modeled, the minimization problem to be solved is the error-in-variable (EIV) formulation of the form. In the EIV formulation, there is a substantial increase in the dimensionality of the optimization problem, which depends on the number of experiments.

2.3 Global Optimization Methods

As stated, global optimization problems involved in the modeling and calculation of phase equilibrium are very challenging. This is because the objective functions are multivariable, non-convex and highly non-linear. For example, global minimization of TPDF and Gibbs free energy are difficult tasks and require robust numerical methods, since these objective functions often have unfavorable attributes such as discontinuity and non-differentiability (e.g., when cubic equations of state or asymmetric models are used for modeling thermodynamic properties). Additional complexities arise near the phase boundaries, in the vicinity of critical points or saturation conditions, and when the same model is used for determining the thermodynamic properties of the mixture (Teh and Rangaiah, 2002; Wakeham and Stateva, 2004). Consequently, TPDF and Gibbs free energy may have several local minima including trivial and non-physical solutions.

Parameter estimation problems can be very difficult to solve reliably even for simple thermodynamic models (Gau et al., 2000; Bollas et al., 2009; Bonilla-Petriciolet et al., 2010). Specifically, a number of pitfalls and difficulties may be faced in parameter estimation for VLE modeling; these include convergence to a local minimum, flat objective function in the neighborhood of the global optimum, badly scaled model functions and non-differentiable terms in thermodynamic equations. In addition, the number of decision variables can be very large, especially for EIV problems. Failure to find the globally optimal parameters for a thermodynamic model and using locally optimal parameters instead, can have significant consequences in phase equilibrium calculations and predictions, may cause errors and uncertainties in equipment design and erroneous conclusions about model performance. Recent studies have shown that using the locally optimal parameters may result in incorrect predictions of the azeotropic states with local composition models and in qualitative discrepancies of the phase behavior such as prediction of spurious phase split and modeling of homogeneous azeotropes as heterogeneous (Gau et al., 2000; Bollas et al., 2009). In summary, several studies have demonstrated the challenging nature of global optimization problems for phase equilibrium modeling and calculations, and they have highlighted the need for reliable numerical techniques to overcome these difficulties.

Global optimization methods can be classified into two broad categories: deterministic and stochastic methods (Rangaiah, 2010). The former methods can provide a guaranteed global optimum but they require certain properties of objective function and constraints such as continuity and convexity. In some cases, problem reformulation is needed depending on the characteristics of the model under study. The stochastic methods generally require little or no assumption on the characteristics of the optimization problem, and yet provide a high probabilistic convergence to the global optimum. Further, stochastic methods are easy to understand, implement and use. They can often locate the global optimum in modest computational time compared to deterministic methods (Blum and Roli, 2003). This section provides the basic concepts and description of deterministic and stochastic methods used for global optimization in phase equilibrium calculations and modeling.

2.3.1 Deterministic Methods

Deterministic optimization methods are those which exploit analytical properties of the problem to generate a deterministic sequence of points (finitely or infinitely) converging to a global optimum (Pardalos et al., 2000). These methods include branch and bound global optimization, homotopy continuation methods, Lipschitz optimization and interval analysis (Floudas, 1999). In the following sections, we briefly summarize different deterministic global optimization methods applied to phase equilibrium calculations and modeling.

2.3.1.1 Branch and Bound Global Optimization

Branch and bound algorithms are a variety of adaptive partition strategies that have been proposed to solve global optimization problems (Floudas, 1999). These methods are based upon partitioning, sampling, and subsequent lower and upper bounding procedures. These operations are iteratively applied to the collection of active (i.e., candidate) subsets within the feasible set *D*. Branch and bound methods are non-heuristic, in the sense that they maintain provable upper and lower bounds on the globally optimal objective value; they terminate with a certificate that the optimal point found is ε-suboptimal.

Branch and bound methods include many specific approaches, and allow for a variety of implementations. These methods typically rely on some a priori knowledge of objective function characteristics and in developing proper structures (i.e., convex terms) of the optimization problem. The general branch and bound methodology is applicable to broad classes of global optimization problems. In general, these optimization algorithms are often slow and require a significant numerical effort that grows exponentially with problem size (Nichita et al., 2002a; Wakeham and Stateva, 2004).

2.3.1.2 Homotopy Continuation Methods

A Homotopy continuation method is considered as a global method since it has the capability of finding all roots of a set of nonlinear equations. In brief, homotopy continuation methods provide a smooth transition between an approximate solution (often linear or nearly linear) and the true solution(s) of a nonlinear equation system, $\boldsymbol{f}(\boldsymbol{u}) = \mathbf{0}$ by gradually introducing the nonlinearities through the use of a scalar homotopy parameter, t (Riggs, 1994; Jalali et al., 2008). These methods are global methods for finding the zeros of nonlinear functions. For global optimization, $\boldsymbol{f}(\boldsymbol{u})$ is a system of non-linear equations obtained from the stationary conditions of the optimization problem. Newton homotopy is usually used in the literature, and it has the form:

$$\boldsymbol{H}(\boldsymbol{u}, t) = t\,\boldsymbol{f}(\boldsymbol{u}) - (1 - t)\,\boldsymbol{g}(\boldsymbol{u}) = \mathbf{0} \tag{2.1}$$

where $\boldsymbol{f}\ (\boldsymbol{u})$ is the system of equations to be solved, $\boldsymbol{g}(\boldsymbol{u})$ is a simple system of equations for which a solution is known or easily found and t is a scalar homotopy parameter, which is gradually varied from 0 to 1 as the path is tracked from the starting point to the true solution.

Note that starting at t = 0, $\boldsymbol{H}(\boldsymbol{u}, 0) = 0$ is trivial to solve given any initial vector, $\boldsymbol{u}^0$. A homotopy path is generated as t increases to unity, where the true solutions occur. A predictor-corrector method can be applied to trace the homotopy paths by integrating along their arc lengths. Beginning on the homotopy path, a tangent vector is computed and a step is taken along the direction of its arc length (Euler's method). The algorithm calculates tangent vectors by solving an initial-value problem. The resulting homotopy paths resemble the solution diagrams obtained through parameterization. When a unique and continuous path exists for $\boldsymbol{H}(\boldsymbol{u}, t)$ from $t = 0$ to $t = 1$, the Newton homotopy-continuation algorithm guarantees global convergence to a single solution; however, it does not guarantee global convergence to multiple solutions. Note that success in finding all solutions along a single path has only been demonstrated for simple polynomials when all variables are relaxed from the real to the complex domain. Therefore, continuation methods can be implemented in both real and complex search spaces (Golubitsky and Schaeffer, 1985).

2.3.1.3 Interval Analysis

The interval analysis method is a general-purpose computational method to solve nonlinear equations to find all solutions lying within the variable bounds (Schnepper and Stadtherr, 1996; Hua et al., 1998). Specifically, consider the solution of a nonlinear equation system, $\boldsymbol{f}(\boldsymbol{u}) = \mathbf{0}$ where $\boldsymbol{u} \in \boldsymbol{U}^0$ and the goal is to enclose, within very narrow intervals, all roots of the equation system in $\boldsymbol{U}^0$. The algorithm is applied to a sequence of intervals, beginning with the initial interval vector $\boldsymbol{U}^0$ specified by the user. For an interval $\boldsymbol{U}^k$ in the sequence, the first step in the solution procedure is the function range test. The interval extension $\boldsymbol{F}(\boldsymbol{U}^k)$ of $\boldsymbol{f}(\boldsymbol{u})$ over the current interval $\boldsymbol{U}^k$ is computed and tested to determine whether it contains zero. If not, then clearly there is no root of $\boldsymbol{f}(\boldsymbol{u}) = \mathbf{0}$ in this interval and can be discarded. If $\boldsymbol{U}^k$ passes the function range test, then the next step is the interval Newton test. This step requires an interval extension of the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{u})$ and involves setting up and solving the interval Newton equation (a system of linear interval equations) for a new interval, which is usually referred as the image. Comparison of this image to the current interval being tested provides an existence and uniqueness test for roots of the equation system. Note that the initial interval should be wide enough so that the interval Newton method provides all solutions of local minima and maxima, saddle points and global minimum for the optimization problem under study (Hansen, 1992; Gecergormez and Demirel, 2005).

2.3.2 Stochastic Methods

Stochastic optimization methods involve probabilistic elements and use random sequences in the search for the global optimum (Gecergormez and Demirel, 2005). These methods employ heuristics for exploring (diversification) and exploiting (intensification) the search space, and learning strategies are used to find quickly near-optimal solutions (Blum and Roli, 2003). The balance between diversification and intensification is important to equilibrate between reliability and computational efficiency (i.e., improve the effectiveness) of finding the global optimum by the stochastic algorithm. Stochastic optimization methods manipulate a single (i.e., point-to-point methods) or a collection of solutions (i.e., population-based methods) at each iteration or objective function evaluation. They include random search, simulated annealing, particle swarm optimization, tabu search, genetic algorithms, differential evolution, ant colony optimization and harmony search. In the following sections, we describe the general characteristics of several stochastic methods used in phase equilibrium modeling and calculations.

2.3.2.1 Random Search

The original random search method is pure random search (PRS) which was first defined by Brooks (1958). It is the simplest algorithm among the random search methods, and consists of generating a sequence of uniformly distributed points in the feasible region, while keeping track of the best point that was already found. PRS offers a probabilistic asymptotic guarantee that the global minimum will be found with probability one as the sample size grows to infinity. Among the random search methods, a direct search algorithm (also called adaptive random search, ARS) proposed by Luus and Jaakola has found many applications in chemical engineering;

it uses random search points and systematic region reduction for locating the global optimum (Luus and Jaakola, 1973).

The ARS algorithm begins with a feasible initial point and region size vector *r*. Then, it generates a number of random points *R* around the initial point. The feasibility of each randomly chosen point is checked. The objective function values of such feasible points are found, and the best point is recorded. In the next iteration, *R* random points are generated around the best point found so far and the same procedure is repeated. After each iteration, the region size is reduced by a certain factor. Iterations are continued until the termination criterion is satisfied. Pseudo-code of ARS is shown in Algorithm 1, and details of this optimization procedure can be found in (Luus, 2001 and 2010). There are several versions of ARS, which have been applied to different chemical engineering application problems (Ali et al., 1997; Luus and Brenek, 1989; Lee et al., 1999; Jezowski et al., 2005; Jezowski et al., 2010).

Algorithm 1 Pseudo-code of Adaptive Random Search

```
Set region size vector $r$ and initialize $x_{\text{best}}$
Give a feasible initial point $x_0$ within the search space
While the stopping criterion is not satisfied
    Randomly generate $R$ points, $x_i$ around $x_0$
    Check the feasibility of each $x_i$
    For $i$ = 1 to $R$
        If $x_i$ is feasible then
            Evaluate objective function at $x_i$
        End if
    End for
    Update $x_{\text{best}}$ based on the objective function value and let $x_0 = x_{\text{best}}$
    Reduce the region size by a certain factor
End while
```

2.3.2.2 Simulated Annealing

Simulated annealing (SA), which was developed by Kirkpatrick et al. (1983), is a stochastic method inspired by the analogy to annealing of metals. In the physical process of annealing, a metal is first heated to its molten state and then slowly cooled to solid state in order to reach thermal equilibrium with minimum energy. This process of slow, controlled cooling scheme of the melted metal to obtain the desired crystalline structure is simulated in SA. It starts from an initial point in the search space and a given high temperature *T*. A new point is randomly created in the neighborhood of the initial point, and its energy (objective function) is evaluated. If this new point has lower energy than the previous one, it is accepted; otherwise, the new point is accepted with probability, $P = \exp(-\Delta E/K_B T)$ where ΔE is the difference in the energy of these two points, K_B is the Boltzmann constant. Generation of new points and their evaluation/acceptance are repeated *N* times at the same temperature to ensure the system is in thermal equilibrium at this *T*. After that, *T* is reduced according to the cooling schedule and the same procedure is repeated until the termination criterion is satisfied. The probability of acceptance, *P* decreases as the search progresses because of lower *T*.

From mathematical point of view, SA can be viewed as a randomization device that allows wrong-way movements during the search for the optimum through an adaptive acceptance/rejection criterion. Based on this concept, SA not only accepts the point with better value but also accepts a point with worse value with some probability, which decreases as search progresses. The main control parameter in the cooling schedule is the temperature, *T*. The main role of *T* is to let the probability of accepting a new move be close to 1 in the early stage of the search and to make it almost zero in the final stage of the search. Convergence to an optimal solution can theoretically be guaranteed after an infinite number of iterations controlled by the procedure of cooling schedule. Pseudo-code of SA is shown in Algorithm 2, and more details of this optimization method are available in Chibante (2010). Various versions of SA have been proposed and applied to chemical engineering problems (Cordero et al., 1997; Hanke and Li, 2000; Zhu et al., 2000; Bonilla-Petriciolet et al., 2007).

Algorithm 2 Pseudo-code of Simulated Annealing

```
Choose an initial point $x_o$
While the stopping criterion is not satisfied
    For $i = 1$ to $N$
        Randomly generate $x_{new}$ around $x_0$
        $\Delta E = f(x_{new}) - f(x_o)$
        If $\Delta E < 0$ then
            $x_o = x_{new}$
        Else
            If random(0,1) < $\exp(-\Delta E/K_B T)$ then
                $x_0 = x_{new}$
            End if
        End if
    End for
    Reduce $T$ according to the cooling schedule
End While
```

2.3.2.3 Genetic Algorithm

Genetic algorithm (GA), developed by Holland (1975), is inspired by the evolutionary process occurring in nature. The main ideas of this algorithm are the ‘survival of the fittest’, and crossover and mutation operations for generating a new solution. GA starts with initializing a population of individuals or trial solutions, which are generated randomly within the feasible region. Objective function value of these individuals is evaluated. The individuals undergo three main operations, namely, reproduction, crossover and mutation. Reproduction creates a mating pool in which the individuals with good fitness will have more copies than the ones with lower fitness value. Crossover is an operation which allows the algorithm to explore the entire search space and to escape from the local minima. In this operation, new strings (individuals) are formed by exchanging the information among parents of the mating pool. Mutation operation involves making changes in each individual directly.

Mutation is exploitative; it can create random small diversions, thereby staying near the parent. After the mutation, the new population is created.

The new population enters into the next generation and the same process of reproduction, crossover and mutation is repeated until the stopping criteria are satisfied. Since the selection of the population for the mating pool is based on the survival of the fittest, the solutions will converge towards its optimal point. GA is probably the most widely known stochastic algorithm, and has found many applications in chemical engineering (e.g., Rangaiah, 2001; Alvarez et al., 2008; Babu et al., 2009; Bonilla-Petriciolet et al., 2011). Pseudo-code of GA is shown in Algorithm 3, and more details of this stochastic method can be found in Younes et al. (2010).

Algorithm 3 Pseudo-code of Genetic Algorithm

```
Initialization:
        Randomly generate $NP$ individuals within the search space
        Evaluate objective function of each of the individuals generated
While the termination criterion is not satisfied
        Reproduction: Create a mating pool of parents
        Crossover: New individuals formed from parents
        Mutation: Randomly modify the new individuals
        Selection: Offspring created by crossover and mutation replaces the original
                   parent population based on its fitness
End While
```

2.3.2.4 Tabu Search

Tabu (or taboo) Search (TS) was developed by Glover in 1989 (Glover and Laguna, 1997). Tabu means that the things must be left alone and should not be visited or touched. Accordingly, the main idea of TS is that the points searched by the algorithm should not be re-visited. This procedure enhances the searching capability of the solution space economically and effectively. Initially, a set of candidate solutions is evaluated and then stored in a taboo list. Then, each new solution generated is compared with the solutions in the taboo list. If the new solution is near to any point in the taboo list, then it will not be evaluated and discarded right away. The length of taboo list is defined by the user. If a new solution enters into the taboo list, the oldest solution in the taboo list will be removed to keep the specified length of the taboo list. After a number of iterations, several promising areas containing the global optimum solution will be found. Then, the intensive search is carried out from these areas to find the global optimum. See the pseudo-code of TS given in Algorithm 4. For more details on this stochastic optimization method, see the book by Glover and Laguna (1997) and the book chapter by Sim et al. (2010). TS has been successfully applied to a wide range of optimization problems (e.g., Teh and rangaiah, 2003; Lin and Miller, 2004; Srinivas and Rangaiah, 2007a; Exler et al., 2008; Mashinchi et al., 2011).

Algorithm 4 Pseudo-code of Tabu Search

```
Randomly generate $N$ initial points, $X_i$ within the search space
Evaluate objective function of all these points, and send them to tabu list
While the termination criterion is not satisfied
    For $i$ =1 to $N$
        Generate a new point $X_{i,new}$
        If $X_{i,new}$ is near any point in the tabu list then
            Discard $X_{i,new}$
        End if
    End for
    Evaluate the objective function at all the remaining points $X_{i,new}$
    Find and save the best point found so far
    Update tabu list
End while
```

2.3.2.5 Differential Evolution

Storn and Price (1997) proposed differential evolution (DE). The main idea behind it is taking the difference between two individuals and adding it to another individual to produce a new individual. It contains four steps similar to GA, namely, initialization of population, mutation, crossover and selection (see the pseudo-code in

Algorithm 5). The main difference between DE and GA is that the search is guided by mutation in the former whereas it is governed by the crossover in the latter. DE algorithm starts with a randomly generated initial population within the search region. For each (target) individual in the population, three other individuals are randomly selected, and the weighted difference between two of them is added to the third individual in order to produce a mutant individual. This operation is called as mutation. Elements of the mutant individual thus obtained are copied to the target individual using crossover constant/probability to produce a trial individual, in the crossover operation. In the selection operation, the better one between the trial and target individuals is selected based on the objective function values, for the next generation. This selection of the fittest individual causes the individuals to improve over the generations, finally converging to an optimum. DE has been successfully applied to a wide range of optimization problems (Chen et al., 2010). More details of DE can be found in Price et al. (2005).

Algorithm 5 Pseudo-code of Differential Evolution

```
Initialization:
    Randomly generated N individuals ($x_i$) within the search space
    Evaluate the objective function of all these individuals, and find the best, $x_{Best}$
While the termination criterion is not satisfied
    For $i$ =1 to N
        Randomly choose 3 individuals ($x_{r1} \neq x_{r2} \neq x_{r3}$ ) from the current population
        Mutation to find mutant individual: $v_i = x_{r1} + F\ (x_{r2} - x_{r3})$
        Crossover: For $j$ = 1 to $D$
                    If rand(0,1) ≥ Cr then
                        $u_{i,j} = v_{i,j}$
                    Else
                        $u_{i,j} = x_{i,j}$
                    End If
                End For
        Find the objective function of the new (trial) individual
        Between $u_i$ and $x_i$, the better one goes to next generation
        Update $x_{Best}$
    End For
End While
```

2.3.2.6 Particle Swarm Optimization

Particle swarm optimization (PSO), developed by Eberhart and Kennedy (1995), exploits swarm intelligence (i.e., the behavior of a biological social system like a flock of birds or a school of fish) for finding the global optimum. This search algorithm is also a population-based stochastic optimization technique. The swarm in PSO consists of a number of particles, each of which represents a potential solution in the search space. Each particle moves to a new position according to certain velocity and the previous position of the particle.

PSO algorithm starts with a randomly generated initial population of particles in the search space. Unlike other evolutionary optimization methods, particles in PSO do not recombine genetic material directly between individuals during the search, but work according to the social behavior of swarms instead. Therefore, PSO finds the global best solution by simply adjusting the moving vector of each individual according to the personal best and the global best positions of particles in the entire swarm at each time step (generation). In other words, the search process allows particles to stochastically return toward previously successful regions in the search space. Recent developments and applications of PSO can be found in Schwaab et al. (2008), Skolpap et al. (2008), Bonilla-Petriciolet and Segovia-Hernandez (2010), and Zhang et al. (2011). Pseudo-code PSO is presented in Algorithm 6, and more details of this method can be found in Kennedy et al. (2001).

Algorithm 6 Pseudo-code of Particle Swarm Optimization

```
Initialization:
    Randomly generate $N$ particles ($x_i$), velocities ($v_i$) and positions ($pbest_i$)
    Evaluate objective function of all these particles
    Set the global best particle to $gbest$
While the termination criterion is not satisfied
    for $i = 1$ to $N$
        $v_i = wv_i + c_1 rand(0,1)(pbest_i - x_i) + c_2 rand(0,1)(gbest - x_i)$
        $x_i = x_i + v_i$
        Evaluate the objective function of the new particle
        If $x_i$ better than $pbest_i$
            $pbest_i = x_i$
        End if
        If $x_i$ better than $gbest$
            $gbest = x_i$
        End if
    End for
End While
```

2.3.2.7 Random Tunneling Algorithm

The tunneling method was first introduced by Levy and Montalvo (1985). It is composed of a sequence of cycles, where each cycle has two phases: a local minimization phase and a tunneling phase. In the first phase, a minimization algorithm such as gradient descent or Newton's method is used to minimize the given objective function, $f(x)$ to locate the first local minimum, x^*. In the second phase, the method searches for the zeros of the tunneling function such that $x^0 \neq x^*$ but $f(x^0) = f(x^*)$ Then, the zero point is used as the starting point of the next cycle, and the two phases are repeated sequentially until a stopping criterion such as failure to find a zero within the prescribed CPU time is met.

Jiang et al. (2002) developed one of the tunneling algorithms, namely, random tunneling algorithm (RTA). It is a stochastic algorithm based on the concepts of sub-energy transformation and terminal repeller in the terminal repeller and unconstrained sub-energy tunneling (TRUST) algorithm of Cetin et al. (1993). RTA consists of two phases: a global search phase and a local optimization phase. The global phase perturbs the system randomly from the last local minimum and solves a system of differential equations from the perturbed point to explore new regions of attraction. Then, the local phase employs a local optimization method (e.g., Quasi-Newton method) to find an improved point in the new region. The two phases are repeated until the specified termination criterion is met.

Srinivas and Rangaiah (2006) implemented RTA differently, as in Algorithm 7; it starts with setting parameter values and randomly generating an initial point within the search space. A local optimization is performed from this point to find the local optimum in this area. Then, tunneling phase is started from this local minimum which comprises of three steps. The first step is random perturbation from the current local minimum, and the second step involves tunneling from the perturbed point in a random direction using uniform grid search until it hits the boundary. The third step consists of 1D tunneling from the perturbed point along each coordinate axis.

Algorithm 7 Pseudo-code of Random Tunneling Algorithm

```
Initialization:
    Randomly generate a point, $x$ within the search space
While termination criterion is not satisfied
      Local phase: Local search starts from $x$ and optimum found is $x^*$
      Tunneling: Do
                    Random perturbation from the best local minimum $x^*$
                    Perform tunneling from perturbed point along a random direction
                    1D tunneling from perturbed point along each coordinate axis
                    If any point is better than $x^*$ then
                      Exit Do and go to Local phase
                    End if
                 Until maximum number of perturbations exceed
                 Set last perturbed point as new initial guess $x$
End while
Local search starts from $x$ and optimum found is $x^*$
```

The three steps of tunneling phase are repeated until the number of perturbations reaches the maximum number or a better point is found. If a better point is found, the tunneling phase will be terminated, and this point will be the new initial guess for the local minimization phase; else, the last perturbed point will be the new initial guess. In the local minimization, a new local minimum is found and compared with the previous local minima, and the best minimum is taken as the current local minimum for the subsequent tunneling phase. The cycle of local minimization and tunneling is repeated until the number of tunneling phases reaches the maximum specified number. The algorithm then terminates declaring the last/best local minimum as the global minimum. More details about this RTA algorithm can be founded in Srinivas and Rangaiah (2006).

2.3.2.8 Ant Colony Optimization

Ant colony optimization (ACO) is a novel meta-heuristic that mimics foraging behavior of real ant colonies. Dorigo et al. (1996) developed the first ant algorithm, and since then several improvements of the ant system have been proposed (Jayaraman et al., 2010). It is an evolutionary approach where several generations of artificial ants search for good solutions in a co-operative way. These ants deposit pheromone on the ground for making some favorable paths that should be followed by other members of the colony. Note that the indirect communication between the ants is performed by means of pheromone trails which enable them to find short paths between their nest and food sources. This characteristic of real ant colonies is exploited in ACO algorithms in order to solve optimization problems. On the other hand, pheromone evaporation is a process of decreasing the intensities of pheromone trails over time. This process is used to avoid local convergence and to explore more in the search space. The meta-heuristic of classical ACO consists of three basic components, and its pseudo-code is given in Algorithm 8.

Algorithm 8 Pseudo-code of Ant Colony Optimization

```
While termination conditions not met, do
    Schedule activities
        Ants generation and activity
        Pheromone evaporation
        Daemon actions
    End Schedule activities
    Evaluate objective function
End while
```

Ants find solutions, starting from an initial value and moving to feasible neighbor regions, in the step of *Ants generation and activity.* During this step, information collected by ants is stored in the so-called pheromone trails. An agent-decision rule, made up of the pheromone and heuristic information, guides the ant's search toward neighbor regions stochastically. Objective function values of candidate solutions are usually used to modify the pheromone values in a way that is deemed to bias future sampling towards high quality solutions. However, due to *pheromone evaporation*, later generations of ants have smaller influence of the pheromone values than earlier. Ants use this information and make their decisions according to the probability distribution determined by the relative size of the pheromone values corresponding to the possible outcomes of the decision variables. Finally, *Daemon actions* are optional for ACO, and they can be used to implement centralized actions which cannot be performed by single ants. Examples are the application of local search methods to the constructed solutions, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a non-local perspective. Details of this stochastic optimization method can be found in Jayaraman et al. (2010).

2.3.2.9 Harmony Search

Harmony search (HS) is a music-inspired meta-heuristic algorithm, which has been introduced by Geem et al. (2001). This stochastic optimization method was developed in analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony. Specifically, musicians may perform the following steps to improvise: playing an existing score from memory, performing variations on an existing piece, or creating an entirely new composition. In the optimization context, each musician is replaced with a decision variable, and the possible notes in the musical instruments correspond to the possible values for the decision variables. So, the harmony in music is analogous to the vector of decision variables, and the musician's improvisations are analogous to local and global search schemes in optimization techniques. HS combines heuristic rules and randomness to imitate this music improvisation process.

Briefly, HS involves three stochastic operators to perform both diversification and intensification stages: a) memory consideration, b) pitch adjustment, and c) random selection. Pseudo-code of HS is given by Algorithm 9. The diversification is controlled by the pitch adjustment and random selection operators, while memory consideration is generally associated with the intensification. The proper combination of these operators is important to favor the performance of HS in global optimization. This iterative procedure is repeated until the convergence criterion is satisfied. Recently, some modifications have been proposed in the literature to improve the convergence performance of the original HS. According to Geem (2009), variations proposed for HS may involve: a) mechanisms for the proper initialization of HS parameters, b) mechanisms for the dynamic adaptation of HS parameters during optimization, and c) the application of new or modified HS operators that include hybrid methods using other meta-heuristics such as SA or DE. For example, two typical and promising variants of HS are the Improved HS and the Global-Best HS. For more details on this meta-heuristic, consult the book by Geem (2009).

Algorithm 9 Pseudo-code of Harmony Search

```
While termination conditions are not met, do
    Perform Improvisation (i.e., generate a new solution)
        Memory consideration
        Pitch adjustment
        Random selection
    End Improvisation
    Evaluate objective function
End while
```

2.3.2.10 Hybrid Methods

In recent years, many hybrid methods have been proposed and studied. A judicious combination of effective concepts of different meta-heuristics can provide a better algorithm for dealing with real world and large scale problems (Talbi, 2002). The hybrid algorithm usually provides several advantages such as better solution using less computational time and handle large or difficult problems (Srinivas and rangaiah, 2007b; Balsa-Canto et al., 2008; Jourdan et al., 2009; Liu and Wang, 2009; Zhang and Rangaiah, 2011). We focus here on hybrid algorithms that have been applied to phase equilibrium modeling and calculations.

Chaikunchuensakun et al. (2002) presented a combined algorithm based on nonlinear parametric optimization (NLQPB) routines. It solves the Kuhn-Tucker conditions by minimizing a quadratic sub-problem with linearized equality and inequality constraints. The solution vector of the quadratic sub-problem is used as a search direction until sufficient decrease of a merit function is found. The approximate Hessian matrix is updated for each quadratic sub-problem by the quasi-Newton algorithm. Mitsos and Barton (2007) proposed a hybrid method which combines CPLEX and BARON solvers in GAMS. The upper bound of the problem is solved with CPLEX, and lower bound of the problem is solved through BARON in order to enhance its reliability. Srinivas and Rangaiah (2007b) proposed a hybrid method which combines DE and tabu list of TS. The tabu list used in DE can avoid re-visiting the same area, increase the diversity of the population, avoid unnecessary function evolutions, enhance global exploration and prevent premature convergence. The proposed method was shown to be more reliable and efficient compared to many other stochastic algorithms (Bonilla-Petriciolet et al., 2010a and 2011). Pereira et al. (2010) combined three solvers in GAMS, where BARON is used for global optimization, MINOS is used as a nonlinear solver and CPLEX is used for linear problems. The proposed algorithm can solve challenging optimization problems.

Srinivas and Rangaiah (2010) proposed two versions of DE with tabu list, referred as DETL-G (wherein the tabu list is implemented in the generation step) and DETL-E (wherein the tabu list is implemented in the evaluation step). These two algorithms combine the good reliability of DE with the computational efficiency of TS. Recently, Zhang et al. (2011) proposed a novel bare-bones PSO for parameter estimation of vapor-liquid data modeling problems. The proposed method combines the mutation strategy of DE with bare-bones PSO for a good balance between the exploration and exploitation to enhance the global search ability.

Besides the above hybrid methods, one common approach is to use stochastic algorithm for global search followed by a local optimizer for intensifying search. Accordingly, a local optimizer has been combined with stochastic optimization algorithms such as GA, SA, PSO, DE and HS (Rangaiah, 2001; Srinivas and Rangaiah, 2007a; Bonilla-Petriciolet et al., 2006, 2007 and 2010b; Lin and Chen, 2007; Staudt et al., 2009; Bonilla-Petriciolet and. Segovia-Hernandez, 2010; Fernandez-Vargas, 2011).

2.4 Applications of Global Optimization Methods to Phase Equilibrium Modeling and Calculations

The following sections summarize studies, mainly from the year 2000, on application and evaluation of deterministic and/or stochastic global optimization methods to phase equilibrium modeling (in Section 2.4.1), phase stability analysis (in Section 2.4.2) and phase equilibrium calculations (in Section 2.4.3).

2.4.1 Applications to Phase Equilibrium Modeling

Deterministic and stochastic global optimization methods have been applied to parameter estimation in VLE modeling, and these are summarized in Table 2.1. In comparison to phase equilibrium calculations, there are fewer studies on the solution of parameter estimation problems for phase equilibrium modeling using global optimization methods.

Table 2.1. Application of Global Optimization Methods to Modeling Vapor-liquid Equilibrium Data

Method (Reference)	Problem formulation	Thermodynamic models
Branch and Bound (Esposito and Floudas, 1998)	Error-in-variable	Local composition model and ideal gas
Interval Analysis (Gau et al., 2000; Dominguez et al., 2002)	Least squares	Local composition models and ideal gas
Interval Analysis (Gau and Stadtherr, 2002)	Error-in-variable	Local composition model and ideal gas
Simulated Annealing (Costa et al., 2000)	Least squares	Equation of state
Simulated Annealing (Bonilla-Petriciolet et al., 2007)	Least squares and Error-in-variable	Local composition models and ideal gas
Random Tunneling (Srinivas and Rangaiah, 2006)	Error-in-variable	Local composition model and ideal gas
Genetic Algorithm (Alvarez et al., 2008)	Least squares	Local composition models and equation of state
Differential Evolution (Kundu et al., 2008)	Least squares	Equation of state
Particle Swarm Optimization, Differential evolution, Simulated Annealing, Genetic Algorithm, Differential Evolution with tabu list (Bonilla-Petriciolet et al., 2010)	Least squares and Error-in-variable	Local composition model and ideal gas
Particle Swarm Optimization (Lazzus, 2010)	Least squares	Local composition models and ideal gas
Bare bone particle swarm optimization (Zhang et al., 2011)	Least squares and Error-in-variable	Local composition models and ideal gas
Harmony Search (Bonilla-Petriciolet et al., 2010)	Least squares	Local composition models and ideal gas
Ant Colony Optimization (Fernandez-Vargas, 2011)	Least squares and Error-in-variable	Local composition models and ideal gas

Specifically, Esposito and Floudas (1998) have reformulated the optimization problem in terms of convex under-estimating functions and then used a branch-and-bound procedure to solve parameter estimation problems. This method provides a mathematical guarantee of global optimality but, in general, it may be necessary to perform problem reformulation and develop convex under-estimators specific to each new application. Gau et al. (2000) and Dominguez et al. (2002) have used an interval analysis approach and classical least square formulation for modeling VLE data. These studies indicated that several sets of parameter values of local composition models published in the DECHEMA VLE Data Collection correspond to local optima. These authors also showed that these locally optimal parameters affect the predictive capability of thermodynamic models for phase equilibrium modeling. Later, Gau and Stadtherr (2002) applied an interval-Newton approach for the reliable solution of EIV parameter estimation problems in VLE modeling of binary systems. This approach can be used for both parameter estimation and data reconciliation.

With respect to stochastic methods, several meta-heuristics have been used to solve the parameter estimation problems in phase equilibrium modeling, and they include SA, GA, RTA, DE, DE with tabu list (DETL), PSO, HS, bare-bones PSO (BBPSO) and ACO. Specifically, Costa et al. (2000) reported the application of SA for parameter estimation in the modeling of vapor-solid equilibrium with supercritical carbon dioxide as the solvent. Results of data fitting using SA were compared with those obtained using the Powell method, and the authors concluded that SA may offer a better performance. Steyer and Sundmacher (2004) used an evolutionary optimization strategy for the simultaneous fitting of VLE and liquid-liquid equilibrium (LLE) data for ternary systems. Bonilla-Petriciolet et al. (2007) also studied the performance of SA for parameter estimation in VLE modeling using both least squares and maximum likelihood formulations. This study concluded that SA is a robust method for non-linear parameter estimation in thermodynamic models. However, in difficult problems (e.g., EIV problems with several decision variables), it may converge to a local optimum of the objective function. Srinivas and Rangaiah (2007a) used a RTA for VLE modeling using the error-in-variable approach. This method was able to solve reliably the two modeling problems having 18 and 34 decision variables, and with a global minimum not comparable to a local minimum. Alvarez et al. (2008) applied and compared two versions of GA for VLE modeling using local composition models and equations of state and LS approach. DE was successfully applied to modeling the equilibrium solubility of CO_2 in aqueous alkanolamines (Kundu et al., 2008).

Recently, the performance of SA, GA, DE, DETL and PSO has been compared for VLE modeling using experimental data for binary systems and both least squares and maximum likelihood criteria (Bonilla-Petriciolet et al., 2010a). This comparison shows that DE and DETL perform better than other algorithms tested in terms of reliability for parameter estimation in VLE data modeling. Further, DETL offers a significant reduction in the computational time. Lazzus (2010) also reported the application of PSO to modeling vapor-liquid equilibrium in binary systems using UNIQUAC and NRTL local composition models. Zhang et al. (2011) studied the performance of PSO and variants of BBPSO algorithms for parameter estimation in VLE modeling problems based on LS and EIV approaches. The reliability of BBPSO proposed by Zhang et al. (2011) is shown to be better than or comparable to other stochastic global optimization methods tested; in addition, it has fewer parameters to be tuned. Preliminary studies have also been performed for parameter estimation in VLE modeling using both HS and ACO (Bonilla-Petriciolet et al., 2010b; Fernandez-

Vargas, 2011). In particular, HS is reliable for solving parameter estimation problems using LS approach but its performance is poor for finding the global optimum using EIV formulation. On the other hand, ACO appears to be a competitive stochastic method for VLE modeling especially using EIV formulation.

The above review indicates that several researchers have studied the parameter estimation in VLE modeling problems using stochastic optimization methods instead of deterministic methods. In particular, stochastic optimization methods may offer reduced computational time and easier implementation than the deterministic approaches. The former methods usually show robust performance for solving parameter estimation problems but, in some challenging problems, they may fail to locate the global optimum especially using fewer function evaluations and for optimization problems with many decision variables (e.g., EIV problems). In addition, the performance of many stochastic methods is significantly dependent on the stopping condition used.

2.4.2 Applications to Phase Stability Analysis

With the introduction of the tangent plane criterion for phase stability analysis, many researchers have studied the solution of this optimization problem using different computational methods. These studies using deterministic and stochastic optimization methods are summarized in Tables 2.2a and 2.2b respectively.

Table 2.2a. Application of Deterministic Optimization Methods to Phase Stability Analysis

Method (Reference)	Problem formulation	Thermodynamic models
Homotopy Continuation (Sun and Seider ,1995)	Tangent plane distance function	SRK and PR
Branch and Bound (Harding and Floudas,2000)	Tangent plane distance function	SRK, PR and van der Waals
Interval Newton/Generalized Bisection (Tessier et al., 2000)	Excess Gibbs energy	NRTL and UNIQUAC
Branch and Bound (Zhu and Inoue, 2001)	Tangent plane distance function	NRTL activity coefficient equation
Interval Newton/Generalized Bisection (Xu et al., 2002)	Volume-based formulation using the Helmholtz energy	Statistical associating fluid theory
Tunneling Method (Nichita et al., 2002)	Tangent plane distance function	SRK and PR
Terrain Method (Lucia et al., 2005)	Projected Gibbs energy and the norm of chemical potentials	PR
Interval Newton method (Gecegormez and Demirel, 2005)	Tangent plane distance function	NRTL
Tunneling Method (Nichita et al., 2006)	Tangent plane distance function in terms of Helmholtz free energy	SRK and PR
CPLEX and BARON (Mitsos and Barton, 2007)	Tangent plane distance function	NRTL and UNIQUAC
Homotopy Continuation (Jalali et al., 2008)	Michelsen criteria	NRTL
Tunneling Method (Nichita et al., 2008)	Tangent plane distance function	Perturbed-chain statistical association fluid theory
Dividing Rectangles (Saber and Shaw, 2008)	Tangent plane distance function	PR and SRK
Tunneling Method (Nichita and Gomez, 2009)	Tangent plane distance function	PR and SRK

Table 2.2b. Application of Stochastic Optimization Methods to Phase Stability Analysis

Method (Reference)	Problem formulation	Thermodynamic models
Genetic Algorithm and Simulated Annealing (Rangaiah, 2001)	Tangent plane distance function	NRTL, UNIQUAC and SRK
Stochastic Sampling and Clustering Method (Balogh et al., 2003)	Modified tangent plane distance function	SRK
Simulated Annealing (Henderson et al., 2004)	Modified tangent plane distance function	SRK and PR
Simulated Annealing, very fast SA, a modified direct search SA and stochastic differential equations (Bonilla-Petriciolet et al., 2006)	Tangent plane distance function	SRK
Differential Evolution and Tabu Search (Srinivas and Rangaiah, 2007a)	Tangent plane distance function	NRTL, UNIQUAC and SRK
Adaptive Random Search (Junior et al., 2009)	Tangent plane distance function	SRK, PR and Perturbed Chain – Statistical associating fluid theory
Repulsive Particle Swarm Optimization (Rahman et al., 2009)	Tangent plane distance function	NRTL and UNIQUAC
Particle Swarm Optimization and its Variants (Bonilla-Petriciolet and Segovia-Hernandez, 2010)	Tangent plane distance function	NRTL, SRK, Wilson, UNIQUAC, ideal solution and gas
Differential Evolution, Simulated Annealing and Tabu Search (Bonilla-Petriciolet et al. 2010)	Tangent plane distance function with reaction	NRTL, Wilson and UNIQUAC

Sun and Seider (1995) introduced homotopy continuation method for phase stability problem, in order to locate all stationary points of the tangent plane distance function (TPDF). However, their technique requires several initial estimates for finding all stationary points of TPDF. Harding and Floudas (2000a) studied the phase stability of three cubic equations of state: SRK, Peng-Robinson and van der Waals, based on analytical findings and the principles of αBB (branch and bound) global optimization framework. In this study, stability problems with several decision variables (≤ 8) have been analyzed.

Tessier et al. (2000) introduced an interval Newton/generalized bisection technique for solving phase stability problems involving excess Gibbs energy models. The proposed technique is independent of initialization, immune to rounding errors, and provides both mathematical and computational guarantees that all stationary points of TPDF are enclosed. Zhu and Inoue (2001) developed a general quadratic under-estimating function based on branch and bound algorithm by the construction of a rigorous under-estimator for TPDF involving NRTL model, and showed its effectiveness for phase stability analysis of three ternary mixtures with up to 2-3 phases. Xu et al. (2002) studied the phase stability criterion involving the statistical associating fluid theory equation of state model. They introduced an interval Newton/generalized bisection algorithm and a volume-based formulation for the Helmholtz energy, and then applied them successfully to non-associating, self-associating, and cross-associating systems.

Nichita et al. (2002b) proposed the tunneling method for phase stability analysis with cubic equations of state by minimization of the TPDF on a variety of representative systems. Their results show that the proposed method is very robust even for the very difficult systems. Lucia et al. (2005) incorporated some new ideas within the terrain methods and applied them to phase stability and equilibrium of n-alkanes mixtures. This method provides global knowledge for understanding the solution structure, saddle points and other information. Gecegormez and Demirel (2005) introduced interval Newton method for phase stability analysis of binary systems and ternary systems modeled by NRTL, to locate all the stationary points. Their results confirm that the interval Newton method is able to locate all the stationary points of TPDF. Mitsos and Barton (2007) reinterpreted the Gibbs tangent plane stability criterion via a Lagrangian duality approach, as the solution of the dual problem of a primal problem that minimizes Gibbs free energy subject to material balances. Then, this optimization problem was solved using CPLEX and BARON in GAMS. Nichita et al. (2008a) used the tunneling method to solve the non-convex optimization problem that results from the TPDF in terms of the Helmholtz free energy.

Jalali et al. (2008) studied homotopy continuation method for phase stability analysis in the complex domain using Michelsen criteria (Michelsen, 1982). However, this approach is not possible if the equations cannot be converted into complex variables. Nichita et al. (2008b) applied the tunneling method to solve the phase stability problem for more complex equation of state like perturbed-chain statistical association fluid theory. Calculations were performed for several benchmark problems and for binary and multi-component mixtures of non-associating molecules. Saber and Shaw (2008) tested dividing rectangles (DIRECT) global optimization algorithm for optimizing TPDF with SRK equation of state for multi-component mixtures and near critical-point systems, and showed that this algorithm has better robustness and efficiency compared to Lipschitz method, interval Newton method, tunneling method, very fast simulated annealing, stochastic differential equations and/or modified direct search annealing. Nichita and Gomez (2009) applied the tunneling method to perform stability analysis of various systems modeled by PR and SRK equation of state.

Besides the application of deterministic methods outlined in the above paragraphs, stochastic methods have been studied by many researchers for phase stability problems. Rangaiah (2001) applied GA and SA to phase stability problems of various systems. The results show that the former is more efficient and reliable than the latter. Balogh et al. (2003) introduced a stochastic sampling and clustering method, and applied it to a modified TPDF with an equation of state as the thermodynamic model. This method was able to solve small to moderate size problems efficiently and reliably. Henderson et al. (2004) formulated the phase stability optimization problem with a slight modification of the Gibbs tangent plane criterion, and used SA to solve it. Bonilla-Petriciolet et al. (2006) compared four algorithms: SA, very fast SA, a modified direct search SA and stochastic differential equations, on several phase stability problems. Their results show that SA is the most reliable among the methods tested for minimization of TPDF for both reactive and non-reactive mixtures.

Srinivas and Rangaiah (2007a) investigated solution of phase stability problems with DE and TS, and reported that the former has better reliability but less computational efficiency compared to the latter. Junior et al. (2009) applied a hybrid adaptive random search method to solve the phase stability problems for three different equation of state models. Their results show that the proposed method outperforms the classical adaptive random search, quasi-Newton and DIRECT methods. Rahman et al. (2009) tested a repulsive PSO for phase stability problems. This optimization algorithm uses the propagation mechanism to determine new velocity for a particle. Consequently, it can prevent the swarm from being trapped in a local minimum. Ferrari et al. (2009) used SA and PSO for modeling liquid-liquid equilibrium data of binary and multi-component systems. They concluded that both algorithms are robust for estimating the model parameters in these applications. Bonilla-Petriciolet and Segovia-Hernandez (2010) performed a comparative study of different variants of PSO algorithms for phase stability of multi-component mixtures. Their results indicate that the classical PSO with constant cognitive and social parameters is reliable and offers the best performance for global minimization of TPDF in both reactive and non-reactive systems.

Srinivas and Rangaiah (2010) proposed two versions of DE with tabu list, referred as DETL-G and DETL-E, and applied to phase stability problems. Their results show that the overall performance of DETL-G and DETL-E is better than that of DE and TS. Bonilla-Petriciolet et al. (2010c) studied phase stability and equilibrium calculations in reactive systems using DE, SA and TS, and showed that DE and TS are better than SA in terms of efficiency but not so in terms of reliability. In these and other studies (Rangaiah, 2001; Srinivas and Rangaiah, 2007a and 2010; Bonilla-Petriciolet and Segovia-Hernandez, 2010), a local optimization technique was used after the global search for efficiently and accurately finding the (global) minimum. Among the many stochastic methods tested and compared for solving the phase stability problems, DETL has shown better performance.

It is clear that both stochastic and deterministic methods can be used for reliably solving phase stability problems in multi-component system. Overall, finding all stationary points of TPDF is not an easy task because a search over the entire composition space is required and the number of these stationary points is also unknown. Hence, it is better to find the global optimum of TPDF during phase stability analysis. Several studies indicate that optimization methods tested may fail to find the global optimum in phase stability analysis when there are comparable minima (i.e., the difference in function values at the global minimum and at a local minimum is very small). The reduction of CPU time taken by global optimization methods is one of the major challenges in phase stability analysis of multi-component systems. This improvement would allow us to extend the application of these strategies for performing phase equilibrium calculations in more complex systems.

2.4.3 Applications to Phase Equilibrium Calculations

Both deterministic and stochastic global optimization methods have been applied for phase equilibrium calculations of different systems with and without chemical reactions; these investigations are summarized in Tables 2.3a and 2.3b. For example, Lucia et al. (2000) introduced unique initialization strategies and successive quadratic programming for phase equilibrium calculations. The overall algorithmic framework is based on using a combination of binary tangent plane analyses, bubble point calculations and dimensionless Gibbs free energy minimization for solving a sequence of sub problems (i.e., VLE, LLE, and VLLE). Chaikunchuensakun et al. (2002) applied a combined algorithm, NLQPB, for the calculation of multi-phase equilibrium conditions at fixed temperature, pressure and overall composition. Although global solutions cannot be guaranteed, NLQPB can find equilibrium compositions accurately for multi-phase mixtures by the minimization of the Gibbs free energy of the system. Cheung et al. (2002) developed a branch-and-bound algorithm, which incorporates tight convex under-estimators and bounds on the dependent variables approach, and applied it to determine the global minimum potential energy for the solvent-solute interactions in phase equilibrium.

Table 2.3a. Application of Deterministic Optimization Methods to Phase Equilibrium Calculation

Method (Reference)	Problem formulation	Thermodynamic models
Successive Quadratic Programming (Lucia et al. 2000)	Gibbs free energy	NRTL, UNIQUAC, UNIFAC, RK, PolyNRTL, HOC, SRK and PolySRK
Nonlinear parametric optimization (NLQPB) (Chaikunchuensakun et al., 2000)	Gibbs free energy	UNIQUAC, PR and van der Waals
Branch and Bound (Cheung et al., 2002)	Potential energy	van der Waals and Coulombic
Tunneling Method (Nichita et al., 2002)	Gibbs free energy	SRK and PR
Interval Analysis (Scurto et al., 2003)	Gibbs energy surface	PR and van der Waals
Tunneling method (Nichina et al., 2004)	Gibbs free energy	SRK and PR
CONOPT in GAMS (Rossi et al., 2010)	Gibbs free energy	NRTL and Wilson
Duality based optimization (BARON, MINOS and CPLEX) (Pereira et al., 2010)	Helmholtz free energy	Augmented van der Waals

Table 2.3b. Application of Stochastic Optimization Methods to Phase Equilibrium Calculations

Method (Reference)	Problem formulation	Thermodynamic models
Enhanced Simulated Annealing (Zhu et al. 2000)	Gibbs free energy	PR and SRK
Genetic Algorithm, Simulated Annealing and hybrid GA (Teh and Rangaiah, 2001)	Gibbs free energy	SRK, PR, NRTL and UNIFAC
Enhanced Tabu Search (Teh and Rangaiah, 2003)	Gibbs free energy	SRK, PR, NRTL and UNIFAC
Random Tunneling Algorithm (Srinivas and Rangaiah, 2006)	Gibbs free energy	SRK, PR, NRTL and ideal gas
Differential Evolution and Tabu Search (Srinivas and Rangaiah, 2007a)	Gibbs free energy	SRK, PR, NRTL and UNIFAC
Differential Evolution with Tabu List (Srinivas and Rangaiah, 2007b)	Gibbs free energy	SRK, PR, NRTL and UNIFAC
Hybrid Artificial Immune System (Lin and Chen, 2007)	Gibbs free energy with reaction	NRTL and UNIQUAC
Simulated Annealing (Bonilla-Petriciolet et al, 2009)	Gibbs free energy with its orthogonal derivatives	NRTL, Wilson and ideal gas
Hybrid Genetic Algorithm with Interior Point Method (Staudt and Soares, 2009)	Gibbs free energy	NRTL, SRK and PR
Genetic Algorithm and Differential Evolution with Tabu List (Bonilla-Petriciolet et al, 2011)	Gibbs free energy with reaction	NRTL, Wilson, UNIQUAC and Margules solution

Nichita et al. (2002a) tested the tunneling method for multi-phase equilibrium calculation by direct minimization of Gibbs free energy of a variety of multi-component systems. Their results suggest that tunneling method is a robust and efficient tool for solving phase equilibrium problems even for extremely difficult cases. However, it requires feasible and improved initial estimates for reliability and computational efficiency respectively. Scurto et al. (2003) applied interval analysis methodology to predict the behavior of high-pressure solid-multiphase equilibrium systems using cubic equations of state with co-solvents, where the likelihood of formation of more than two phases is great. Nichita et al. (2004) too used the tunneling method to directly minimize Gibbs free energy in multi-phase equilibrium calculations. Rossi et al. (2010) applied convex analysis method to chemical and phase equilibrium of closed multi-component reactive systems. This method employs the CONOPT solver in GAMS. The optimization is to minimize Gibbs free energy of systems at constant pressure and temperature, and constant pressure and enthalpy. The proposed method can solve the phase equilibrium problems with high efficiency and reliability but it requires the convexity of the model. Pereira et al. (2010) proposed a duality-based optimization for phase equilibrium where the volume-composition space is converted from Gibbs free energy to Helmholtz free energy. They used BARON for global optimization, MINOS as the nonlinear local solver and CPLEX for linear problems. The method is applicable to the calculation of any kind of fluid phase behavior (e.g., VLE, LLE and VLLE). The method proposed by Pereira et al. (2010) can guarantee the global optimum but it requires a differentiable objective function.

Beside deterministic methods for solving the phase equilibrium problems reviewed above, many stochastic methods have been used to solve them. Zhu et al.

(2000) introduced enhanced SA for phase equilibrium calculations of multi-component systems at high pressure, which include ternary, quaternary and five component mixtures. Although the proposed algorithm requires slightly more computational time compared to two algorithms in the literature (MULPRG and HOMPEQ), it provides comparable reliability, is self-starting and simple. Rangaiah (2001) evaluated the performance of GA, SA and hybrid GA for phase stability problems of several mixtures. The results show that GA is more efficient and reliable than SA, and that hybrid GA outperforms both GA and SA in terms of reliability but its main limitation is the significant increase in the CPU time. Teh and Rangaiah (2003) tested enhanced continuous TS for phase equilibrium calculations via Gibbs free energy minimization, of VLE, LLE and VLLE systems. Their results indicate that TS is more efficient than GA but both require further improvement for 100% reliability.

Srinivas and Rangaiah (2006) evaluated RTA on a number of medium sized problems including VLE, LLE and VLLE problems. This algorithm can locate the global optimum for most of the examples tested but its reliability is low for problems having a local minimum comparable to the global minimum. Srinivas and Rangaiah (2007a) compared DE and TS algorithms for phase equilibrium calculations of various VLE, LLE and VLLE systems. Subsequently, Srinivas and Rangaiah (2007b) introduced DETL algorithm for phase equilibrium calculation. Their results show that this hybrid algorithm performs better than both DE and TS. Lin and Chen (2007) proposed a hybrid method for chemical reaction and phase equilibrium calculation. This method was constructed by making use of the advantages of artificial immune system and sequential quadratic programming. The results show that the hybrid method is better than the artificial immune system method alone.

Staudt and Soares (2009) proposed a hybrid global optimization method for the minimization of Gibbs free energy for multi-phase equilibrium calculation. The proposed method uses GA for the global search and interior point method for refinement after the global search. Bonilla-Petriciolet et al. (2011) applied GA and DETL for phase equilibrium calculations in reactive systems by Gibbs free energy minimization; two approaches – constrained and unconstrained, were tried for solving these problems. The results show that unconstrained free energy minimization involving transformed composition variables requires more computational time compared to constrained minimization, and that DETL has generally better performance for free energy minimization in reactive systems. Among the stochastic methods, hybrid methods often provide better performance in terms of reliability and efficiency.

In summary, the literature indicates that the major difficulties of Gibbs free energy minimization using both deterministic and stochastic methods arise in phase equilibrium calculations for highly non-ideal mixtures. For some conditions, difference of objective function value at the global minimum and at a local optimum (i.e., at trivial solutions and unstable phase equilibria) is also very small. In fact, trivial solutions present a significant region of attraction for numerical strategies, and may cause convergence problems. Many of the studies and tests assume that the correct number of phases at equilibrium is known *a priori*. However, the number and type of phases, at which Gibbs free energy function achieves the global minimum, are unknown in phase equilibrium problems and, as a consequence, several calculations must be performed using different phase configurations (adding or removing phases) to identify the stable equilibrium state. Hence, it is desirable to develop more effective deterministic and stochastic methods for the global Gibbs free energy minimization in both reactive and non-reactive systems.

2.5 Concluding Remarks

Optimization problems involved in phase equilibrium modeling and calculations are complex and difficult to solve using traditional local optimization methods due to (a) the presence of several local minima, (b) the objective function may be flat and/or with discontinuities in some regions of solution domain, (c) wide range of decision variables in modeling problems, and (d) presence of trivial solutions in some problems. In fact, these optimization problems are generally non-convex, constrained, and highly non-linear. Hence, solution of these important and common problems requires reliable and efficient global optimization methods able to handle different problem characteristics. To date, a number of deterministic and stochastic global optimization methods have been developed and evaluated for solving phase equilibrium modeling and calculation problems. These methods have been widely applied to solve phase stability and Gibbs free energy minimization problems in non-reactive systems. However, fewer attempts have been made in the application of these methods to reactive phase equilibrium calculations and modeling, compared to those reported for non-reactive systems.

Even though research in the application of global optimization methods for phase equilibrium modelling and calculations has grown significantly over the last decade, results reported in the literature indicate that both deterministic and stochastic global optimization methods require further improvement for solving, robustly and efficiently, these application problems. One of the major limitations of deterministic global optimization methods is the significant computational time required for solving high dimensional problems, which grows exponentially with the number of decision variables. This aspect limits the application of these strategies to model multi-component and multi-phase systems and the use of complex thermodynamic models for predicting mixture properties. Therefore, further research should be performed to improve the performance of available optimization algorithms and to develop general purpose and effective deterministic methods for solving phase equilibrium problems in multi-component systems.

Compared to deterministic optimization methods, stochastic optimization techniques involve simple concepts, do not require any assumptions and can be used for any type of problem. Hybridization to synergize selected features of different stochastic algorithms is a promising approach for developing highly effective algorithms since reported results show that the performance of pure algorithms is almost always inferior to that of hybrid algorithms. Therefore, further studies should be focused on the development of hybrid strategies to improve the reliability of stochastic optimization methods using fewer NFE. In addition, alternative termination criteria should be studied and tested for reliably determining the global convergence of stochastic optimization methods for phase equilibrium modeling and calculations. It is also desirable that these methods should have no or fewer tuning parameters. Despite the many advances in this area, research in global optimization for phase equilibrium modeling and calculations will continue to be an active field in chemical engineering, in order to develop and evaluate effective global optimization methods, in the foreseeable future. Further, promising deterministic and stochastic methods need to be compared carefully and comprehensively for solving phase equilibrium modeling and calculation problems.

Chapter 3

An Integrated Differential Evolution with a Novel Stopping Criterion*

3.1 Introduction

Global optimization is a fast growing area in the recent few decades due to its applications in many areas such as mathematics, science and engineering. Available global optimization methods can be classified into two broad categories (Pardalos et al., 2000; Moles et al., 2003): deterministic methods (Horst and Tuy 1996; Pinter 1996; Floudas 1999; Esposito and Floudas 2000; Lucia, Gattupalli et al. 2008) and stochastic methods (Guus et al. 1995; Zhigljavsky and Zilinskas 2007). Deterministic algorithms are most often used for specific problems and when the clear relation between the characteristics of the possible solutions and the problem is available (Nocedal and Wright. 2006; Weise 2008). They can guarantee the global optimality of the final solution under certain conditions such as continuity and convexity (Moles 2003; Teh and Rangaiah 2003). However, no algorithm can solve general global optimization problems with certainty in finite time (Weise 2008). If the relation between a candidate solution and the objective function is too complicated or the dimensionality of the search space is very high, it becomes harder to solve an optimization problem deterministically (Moles, et al., 2003; Liberti and Kucherenko 2005; Weise 2008).

* This chapter is based on the paper - Zhang, H., Rangaiah, G.P. and Bonilla-Petriciolet, A., Integrated Differential Evolution for Global Optimization and its Performance for Modeling Vapor-Liquid Equilibrium Data. *IECR*, vol.50, pp.10047-10061, 2011.

The most challenging global optimization problems in real life are those without any known structure that can be used, the so-called black-box optimization problems (Pardalos et al., 2000; Exler et al., 2008). Stochastic optimization algorithms, whose search and outcome are random, are designed to deal with such problems or highly complex problems. They generally require little or no additional assumptions on the optimization problem, are simple to implement and use, and do not require transformation of the original problem, which can be treated as a black box. This characteristic is especially useful if the researcher has to link the optimizer with a simulator such as Aspen Plus and Hysys. On the other hand, stochastic algorithms cannot guarantee global optimality (except for infinite iterations), but they can locate the global optimum with high probability in modest computation times (Moles et al., 2003; Lin and Miller 2004; Liberti and Kucherenko 2005).

Many stochastic global optimization methods have been developed and used for diverse applications. The well-known stochastic global optimization methods include GA, evolutionary strategy, SA, DE, TS and PSO. Our recent book covers these and their applications in chemical engineering (Chen et al., 2010). Among these, DE is a promising population-based stochastic optimization algorithm proposed by Storn and Price (Storn and Price 1997; Price et al., 2005). It has been gaining popularity due to its simplicity, faster convergence and capability to handle non-differentiable and multi-model problems (Price et al., 2005; Babu and Munawar 2007). It has found many applications in chemical engineering (Chen et al., 2010). In general, users need to choose suitable values of DE parameters (namely, population size NP, mutation/scaling factor F and crossover rate Cr), and proper mutation strategy for different problems in order to find the global optimum. These selections require effort and expertise as an improper choice can result in computational inefficiency. In order to overcome these difficulties, recently, researchers have been studying strategies to adapt the algorithm parameters of DE according to the performance in the previous generations (Brest et al., 2006 and 2008; Soliman and Bui 2008; Omran et al., 2009; Qin et al., 2009; Liao, 2010). Of these, Qin et al. (2009) developed a comprehensive self-adaptive DE (SaDE) to adapt mutation strategy, F and Cr, and showed it to be better than the conventional DE and three recent adaptive DE variants on more than 20 benchmark optimization problems having up to 30 decision variables and bounds on variables but no constraints. In parallel to these works, there have been a number of attempts to improve DE (Lampinen, 2002; Angira and Babu, 2006; Srinivas and Rangaiah, 2007b and 2007c; Yang et al., 2008; Zhang et al., 2008; Ali and Kajee, 2009; Pant et al., 2009). The work of Srinivas and Rangaiah (2007b and 2007c) improved the computational efficiency of DE by including the taboo list of taboo search (DETL) which avoids revisits during the search. Results show that DETL outperforms the classic DE and modified DE of Angira and Babu (2006).

Although there are many stochastic global optimization methods with good performance, there are still some challenges. Limitations of stochastic optimization algorithms include tuning of algorithm parameters, lack of good stopping criterion and difficulty to overcome the premature convergence. The main objective of the present work is to develop an efficient and reliable DE for practical applications that can overcome these disadvantages. Motivated by the desirable features and performance of DETL and parameter adaptation, both these are integrated together with a novel termination criterion to stop the global search reliably and use a local optimizer for finding the minimum accurately and efficiently. The resulting algorithm, IDE, is tested on many challenging benchmark problems, and the effectiveness of the novel termination criterion is compared with another used earlier (Bonilla-Petriciolet et al., 2006; Babu and Munawar, 2007; Srinivas and Rangaiah, 2007b and 2007c; Zielinski and Laur 2007; Bonilla-Petriciolet et al., 2010c). Note that an effective stopping criterion is critical for efficiently and reliably solving application problems where the global minima are unknown, and yet it has not received much attention in the literature on stochastic global optimization.

The remainder of this chapter is organized as follows. Classical DE is outlined in Section 3.2, and development and description of the IDE are presented in Section 3.3. The performance of IDE on benchmark functions is reported and discussed in Section 3.4. The effect of NR$_{\max}$ is studied in Section 3.5. Finally, Section 3.6 concludes this chapter.

3.2 Differential Evolution Algorithm

DE is a population-based, real-coded direct search algorithm. It can be used to solve the following type of optimization problems:

$$Min \;\; F_{obj}(x^j) \;\; j = 1, 2, ..., D \;\; \text{subject to} \;\; x^j_{\min} \leq x^j \leq x^j_{\max} \qquad (3.1)$$

Here, F_{obj} denotes the objective function, x^j is the j^{th} decision variable, D denotes the number of decision variables in (i.e., dimension of) the problem, and x^j_{min} and x^j_{max} are respectively the lower and upper bound on each decision variable. DE uses a population of *NP D*-dimensional vectors, which are also called individuals. It has four main steps: initialization, mutation, crossover and selection, as outlined in the pseudo-code in Fig. 3.1. Mutation and crossover steps generate new individuals, and they together with the selection step constitute one generation or iteration of the DE algorithm.

```
Step 1 Initialization

    Initialize the generation number as $G = 0$.
    Randomly initialize $NP$ individuals (target vectors $X_{i,G} = \{x_{i,G}^{1}, ..., x_{i,G}^{D}\}$ for $i = 1, 2, ..., NP$) within the search range [$X_{min}$,$X_{max}$], where $X_{\min} = \{x_{\min}^{1}, ..., x_{\min}^{D}\}$ and $X_{\max} = \{x_{\max}^{1}, ..., x_{\max}^{D}\}$.

    FOR $i$ = 1 to $NP$
        FOR $j$ = 1 to $D$
            $x_{i,G}^{j} = x_{\min}^{j} + rand(0,1) * (x_{\max}^{j} - x_{\min}^{j})$
        END FOR
    END FOR

Step 2 Generation

    WHILE stopping criterion is not satisfied
    •   Mutation Step: /generate a mutation vector $V_{i,G} = \{v_{i,G}^{1}, ..., v_{i,G}^{D}\}$
        FOR $i$ = 1 to $NP$
                $V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$
        END FOR   /where $F$ is the mutation factor and subscripts: $r_1$, $r_2$ and $r_3$ are randomly chosen integers from 1 to $NP$ such that $r_1 \neq r_2 \neq r_3 \neq i$
    •   Crossover Step: / generate a trial vector $U_{i,G} = \{u_{i,G}^{1}, ..., u_{i,G}^{D}\}$
        FOR $i$ = 1 to $NP$
            FOR $j$ = 1 to $D$
                $u_{i,G}^{j} = \begin{cases} v_{i,G}^{j} & if\ rand(0,1) \leq Cr\ or\ j = j_{rand} \\ x_{i,G}^{j} & otherwise \end{cases}$
            END FOR / where $Cr$ is the crossover rate and $j_{rand}$ is a random integer from 1 to $D$
        END FOR
    •   Selection Step: / select trial or target vector with better objective value as the individual for the next generation.
        FOR $i$ = 1 to $NP$
            $X_{i,G+1} = \begin{cases} U_{i,G} & if\ F_{obj}(U_{i,G}) \leq F_{obj}(X_{i,G}) \\ X_{i,G} & otherwise \end{cases}$
        END FOR
        Increment the generation number $G = G + 1$
    END WHILE
```

Figure 3.1. Pseudo-code of the Classic DE Algorithm

3.2.1 Initialization

The search space is bounded by the minimum bound $X_{\min} = \left\{ x_{\min}^{1}, \ldots, x_{\min}^{D} \right\}$ and maximum bound $X_{\max} = \left\{ x_{\max}^{1}, \ldots, x_{\max}^{D} \right\}$. The decision variables of each individual are generated randomly (using uniformly distributed random numbers) within the search space by $x^{j} = x_{\min}^{j} + rand(0,1) * (x_{\max}^{j} - x_{\min}^{j})$. So, an individual is expressed by the D-dimensional genes $X = \left\{ x^{1}, \ldots, x^{D} \right\}$. Thus, the initial population of NP individuals is produced.

3.2.2 Mutation

One of the vectors (say, i^{th} vector, $X_{i,G}$) in the population is selected as the target vector for possible replacement. Then, the mutation operation is performed by randomly selecting two individuals in the population, calculating the difference vector between them, and then adding the difference vector multiplied by the mutation factor, *F* to another individual to produce a mutation vector, $V_{i,G}$.

$$V_{i,G} = X_{r1,G} + F \bullet (X_{r2,G} - X_{r3,G}) \qquad (3.2)$$

Here, *G* denotes the generation number and subscripts: r_1, r_2 *and* r_3 are randomly chosen integers from 1 to NP (number of individuals in population) such that $r_1 \neq r_2 \neq r_3 \neq i$. So, at generation G, its mutation vector is $V_{i,G} = \left\{ v_{i,G}^{1}, \ldots, v_{i,G}^{D} \right\}$. When the population converges to an optimum, any randomly chosen difference vector will become smaller in magnitude. Eventually when all members converge to a single solution, the difference vector will become zero and mutation operation (Eq. 3.2) will be nearly disabled. Thus, mutation operation is not only determined by F but also by the population diversity.

3.2.3 Crossover

Once the mutation vector is created, it will undergo a binomial crossover between target and mutant vectors ($X_{i,G}$ and $V_{i,G}$) to generate a trial vector. Each gene of trial vector is generated using a random number between 0 and 1; if the random number is lower than crossover rate (*Cr*), the variable of the target vector is chosen, otherwise the variable of the mutation vector is chosen. So the variable of trial vector is

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

Thus, the trial vector is $U_{i,G} = \{u_{i,G}^{1},...,u_{i,G}^{D}\}$. The condition $j = j_{rand}$ (where j_{rand} is a random integer in the range 1 to NP) is introduced to ensure that the trial vector will be different from its corresponding target vector by at least one decision variable.

3.2.4 Selection

After the crossover operation, there is a need to check the boundary violation of the trial vector. If any variable of the trial vector has crossed the upper or the lower bound, it is reinitialized randomly within the decision variable's bounds. After evaluating the objective function value at the resulting trial vector, selection operation is performed. DE uses a greedy selection criterion based on the objective function value. The comparison is performed between the objective function values of the trial vector, $f(U_{i,G})$ and the corresponding target vector, $f(X_{i,G})$. The vector with better objective function value is selected for the next generation. The selection operation is expressed as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G,} & if \ f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G,} & otherwise \end{cases} \tag{3.4}$$

Mutation, crossover and selection steps are repeated for each and every individual in the population as the target vector. This completes one generation. The generations will be repeated until the specific termination criteria are satisfied.

3.3 Integrated Differential Evolution Algorithm

There have been many developments on DE with regards to initialization, mutation, crossover and selection operations as well as hybridization with other methods. Developments up to the year 2002 can be found at http://www2.lut.fi/~jlampine/debiblio.htm, and subsequent developments are reviewed by Chen et al. (2010). These are not repeated here for brevity, and only two relevant developments are outlined here. Hybrid methods combine DE with another optimization method to enhance the performance of DE (Srinivas and Rangaiah, 2007b; Yang et al., 2008; Pant et al., 2009; Liao, 2010). DETL proposed by Srinivas and Rangaiah (2007b and 2007c), was tested on benchmark functions and application problems, and shown to have better performance. In this, the taboo check is implemented after mutation and crossover steps; it is performed by measuring the Euclidean distance between the trial individual and each individual in the taboo list (TL). If the Euclidean distance is less than the taboo radius, the trial individual is rejected and another trial individual is produced by mutation and crossover operations. This procedure is repeated until the Euclidean distance between the trial individual and each individual in the TL is greater than the taboo radius. This operation significantly avoids revisiting the same area, increases the diversity of the population and avoids unnecessary objective function evaluations. Thus, the ability of global exploration is greatly enhanced. On the other hand, several researchers investigated self-adaptive strategies to tune the parameters in DE (Brest et al., 2008;

Nobakhti and Wang, 2008; Omran et al., 2009; Qin et al., 2009). Among these, SaDE, proposed by Qin et al. (2009) was tested on many benchmark functions and shown to be better than nine variants of DE.

Typically, DE requires values for algorithm parameters, uses the stopping criterion of maximum number of generations and has scope to improve its efficacy. Hence, several useful strategies are integrated into DE to develop IDE for finding the global optimum reliably and efficiently. These are: adaptation of mutation parameter and strategy, crossover parameter, inclusion of tabu list and tabu check as in DETL, a novel stopping criterion and a local optimizer after the global search. These additions into the classic DE are briefly discussed in the following sub-sections.

3.3.1 Adaptation of Mutation Strategy

A particular mutation strategy performs differently for solving different optimization problems, and an inappropriate choice of strategies and parameters may lead to premature convergence.(Lampinen, 2002; Price, Storn et al. 2005) Therefore, the mutation strategy candidate pool should contain distinct capabilities for dealing with specific problems at different stages of evolution. Recently, many researchers have studied adapting mutation strategy and parameters in DE`(Lampinen, 2002; Brest et al. 2006; Soliman and Bui, 2008; Omran et al., 2009; Qin et al., 2009; Liao, 2010). Of these, Qin et al.(2009) developed a self-adaptive DE to adapt mutation strategy, *F* and *Cr*, and showed it to be better than the conventional DE and three recent adaptive DE variants on more than 20 benchmark optimization problems having up to 30 decision variables. Hence, the adaptation schemes of Qin et al. (2009) are chosen for developing IDE. Accordingly, the following four mutation strategies are selected for the candidate pool in IDE.

1) DE/rand/1

$$V_{i,G} = X_{r1,G} + F \bullet (X_{r2,G} - X_{r3,G}) \tag{3.5}$$

2) DE/rand/2

$$V_{i,G} = X_{r1,G} + F \bullet (X_{r2,G} - X_{r3,G}) + F \bullet (X_{r4,G} - X_{r5,G}) \tag{3.6}$$

3) DE/rand-to-best/2

$$V_{i,G} = X_{i,G} + F \bullet (X_{best,G} - X_{i,G}) + F \bullet (X_{r1,G} - X_{r2,G}) + F \bullet (X_{r3,G} - X_{r4,G}) \tag{3.7}$$

4) DE/current-to-rand/1

$$U_{i,G} = X_{i,G} + rand(0,1) \bullet (X_{r1,G} - X_{i,G}) + F \bullet (X_{r2,G} - X_{r3,G}) \tag{3.8}$$

Each of the mutation strategies 1, 2 and 3 (Eqs. 3.5, 3.6 and 3.7) will be followed by the binomial crossover operation to produce the trial vector. Mutation strategy 4 (Eq. 3.8) directly produces the trial vector and does not need crossover operation.

During evolution in each generation, one mutation strategy is selected from the candidate pool according to the probability, P_k, k = 1, 2, ..., K, where K is the total number of strategies in the pool. Initially, the probability of each strategy is 0.25 since K = 4 here. In the subsequent generations, the probability of selecting a mutation strategy is based on its success rate in the previous LP number of generations. The number of trial vectors generated by k^{th} strategy that are successfully selected in a generation, g is recorded as $ns_{k,g}$; otherwise, it is recorded as $nf_{k,g}$. After LP generations, the probability of choosing k^{th} strategy in generation G is given by

$$S_{k,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{k,g}}{\sum_{g=G-LP}^{G-1} ns_{k,g} + \sum_{g=G-LP}^{G-1} nf_{k,g}} + \varepsilon \qquad \text{for } k = 1, 2, \ldots, K \tag{3.9}$$

$$p_{k,G} = \frac{S_{k,G}}{\sum_{k=1}^{K} S_{k,G}} \qquad \text{for } k = 1, 2, \ldots, K \tag{3.10}$$

Here, ε = 0.1 is used to prevent the possibility of null success rate. So, the larger success rate of k^{th} mutation strategy in the previous LP generations will result in higher probability of the corresponding mutation strategy to be selected in the current generation *G*.

Stochastic universal sampling method (Baker, 1987), which provides zero bias and minimum spread, is employed to choose one of these four strategies for mutation, based on probability. The probabilities of these strategies are calculated at each generation, using Eqs. 3.9 and 3.10. This method for four strategies and 6 individuals is illustrated in Fig. 3.2. Here, the four strategies are mapped in black/continuous line such that size of each strategy is equal to its probability (e.g., 0.18, 0.31, 0.24 and 0.27 in Fig. 3.2). The equally spaced pointers (individuals) are placed in green line to indicate the strategy that they belong to. The first individual position is given by a randomly generated number in the range [0, 1/6]. For example, the first individual is randomly generated at 0.1, and the strategies are assigned to individuals as follows: strategy 1 to individual 1, strategy 2 to individuals 2 and 3, strategy 3 to individual 4 and strategy 4 to individuals 5 and 6 (Fig. 3.2). It is obvious that the higher probability of a mutation strategy (due to higher success of the trial individuals produced by this strategy in the previous generations) leads to more chances for producing the new trial individuals by this strategy. The probabilities of mutation strategies are calculated once in each generation, which results in adaptive learning to choose the more suitable strategy for the particular problem being solved. This in turn makes the algorithm more reliable and robust.

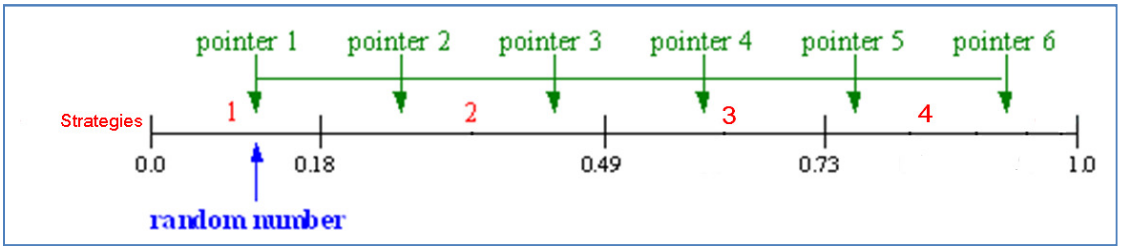


Figure 3.2. Stochastic Universal Sampling Method for Selecting 4 Strategies with 6 Individuals

3.3.2 Adaptation of F and Cr

Mutation factor (*F*) is very important for DE as it is closely related to the convergence speed. So, in order to maintain the balance between exploration and exploitation, *F* is randomly chosen using normal distribution with mean of 0.5 and standard deviation of 0.3. Crossover rate (*Cr*) is another important parameter in DE, which significantly affects the reliability of the algorithm. Cr_k value follows the normal distribution with mean of Crm_k (where *k* is the mutation strategy used) and standard deviation of 0.1. The initial value of Crm_k is set at 0.5, and the successful Cr_k values of each strategy are stored into $CRMemory_k$. After *LP* generations, the median of values stored inside $CRMemory_k$ will be used for Crm_k. Thus, Cr_k value is gradually changed by learning from the previous generations. This option will lead to promising Cr_k value for different kinds of problems and to more reliability (Qin et al., 2009).

3.3.3 DE with Tabu List

The hybrid of classic DE with TL was proposed and tested on benchmark functions by Srinivas and Rangaiah (2007b), and subsequently applied to nonlinear and mixed-integer nonlinear programming problems (Srinivas and Rangaiah, 2007c). The tabu check is implemented after the mutation and crossover steps. In this, if the Euclidean distance between the new trial individual and any of the individuals in the TL is smaller than the specified tabu radius value (which indicates that the trial individual is similar to an individual already visited), the trial individual is rejected and another trial individual is generated by the mutation and crossover steps. This procedure is repeated until the Euclidean distance between the trial individual and each of the individuals in the tabu list is greater than the tabu radius. The procedure makes the individuals in the population more diverse, enhances the exploration of the search space and avoids unnecessary evaluation of the objective function.

Tabu radius is one of the parameters used in the algorithm, and it prevents revisit of the same neighbourhood. So, the higher the value of tabu radius, the larger area will be forbidden for revisiting and vice versa. Obviously, the higher value of tabu radius will cause larger number of rejected points and result in faster termination of the global search by the rejection-based stopping criterion. For problems having a local optimum very near to the global optimum, it is better to choose a smaller tabu radius for higher reliability. The tuning of the tabu radius and nominal values are reported in Srinivas and Rangaiah (2007b).

The objective function is evaluated at the trial individual only if it is away from all the points in the tabu list. After each evaluation, the tabu list is updated dynamically to keep the latest points in the list by replacing the earliest entered point(s). Thus, the new point is added to the list by rejecting the oldest point in the list so that the recently added points are retained (i.e., first-in first-out basis). The tabu check avoids revisiting the same area, increases the diversity of the population and avoids unnecessary function evaluations. Thus, the ability of global exploration is greatly enhanced. The tabu check will require extra computational effort but this is negligible in application problems where objective function evaluation is computationally intensive.

3.3.4 Stopping Criteria

One of the disadvantages of stochastic optimization algorithms is lack of proper stopping criteria. The improper stopping of the algorithm will lead to the final solution either at a local optimum (stop too early) or waste of the computational resources (stop too late). The commonly used stopping criteria for stochastic optimization algorithm in the literature can be classified as: (1) closeness of the optimum found to the exact/known optimum (Brest et al., 2006; Qin et al., 2009); (2) maximum number of function evaluations (fe_{max}) or generations (G_{max}) (Moles et al., 2003; Price et al. 2005; Omran et al., 2009; Liao, 2010); and (3) maximum number of successive iterations without improvement in the best objective function value found so far, which is also known as an improvement-based stopping criterion ($\mathrm{SC}_{\mathrm{max}}$) (Bonilla-Petriciolet et al., 2006; Babu and Munawar, 2007; Srinivas and Rangaiah 2007b and 2007c; Zielinski and Laur 2007; Bonilla-Petriciolet et al., 2010c).

The first type is only suitable for benchmark functions where the optimum is known. It is not applicable for real-world optimization problems where the optimum is not known *a priori*. The second one can be applied to these problems but it is difficult to choose a proper fe_{max} or G_{max} value. Firstly, it is highly dependent on the optimization problem being solved. Secondly, the stochastic optimization algorithm contains some randomness, and so the same problem solved at a different time may need a different fe_{max} or G_{max}. Wrong choice of fe_{max} or G_{max} will lead to a local solution or excessive computation time. The third criterion has been used in the application problems (Angira and Babu, 2006; Bonilla-Petriciolet et al., 2006; Srinivas and Rangaiah 2007b and 2007c; Bonilla-Petriciolet et al., 2010). This is consistent with the conclusion of Zielinski and Laur (2007) that it is better to use a stopping criterion which considers knowledge from the state of optimization run.

Here, a novel stopping criterion specifically for IDE is used after observing detailed results on a number of benchmark functions. It is based on the number of rejected individuals when producing a new trial individual for a target individual in a generation. As the generations proceed, individuals in the population congregate together owing to the nature of DE. When they are close enough, the new trial individual is more likely to be rejected by the tabu check, which is performed using the Euclidean distance between the new trial individual and every individual in the tabu list. If the Euclidean distance is smaller than the tabu radius, then the new trial individual is rejected and another trial individual is generated. So, many rejections of the new trial individual for a target individual in a generation indicate convergence of the algorithm. The stopping criterion based on number of rejections monitors the positions of individuals rather than the objective function values. Thus, it is more robust than the stopping criteria mentioned previously ($\mathrm{SC}_{\mathrm{max}}$ and G_{max}). The stopping criterion based on number of rejections is appropriate for IDE which includes tabu check. In this work, effectiveness of this stopping criterion is compared with $\mathrm{SC}_{\mathrm{max}}$ used earlier.

3.3.5 Hybridize with Local Optimization

There are different ways of hybridization of global optimization with local optimization methods (Talbi, 2002). The traditional hybridization methods are based on the global search followed by a local search. In this approach, switching time between global and local searches is very important. Early switching will increase the probability of trapping in a local optimum. Conversely, a late switching time will waste computational resources.(Miettinen et al., 2006) In this chapter, results are presented later to show that the tabu check in DETL and the stopping criterion based on the number of rejections can determine the switching time efficiently (i.e., the global search is expected to terminate in the valley containing the global optimum). In order to find a precise optimum efficiently, a local optimizer is used after completing the global search, as in our previous works (Srinivas and Rangaiah, 2007b and 2007c). Some researchers have used a local optimizer during the iterations of the global algorithm (Dumas et al., 2009; Tong et al., 2009; Yuan and Qian, 2010). However, this increases computational effort, its benefit is unclear and so this approach is not chosen for the IDE algorithm.

The IDE algorithm is implemented in MS Excel spreadsheet environment using VBA (Visual Basic Application) language because of its ready availability and used by researchers and practitioners in diverse fields. Solver tool in Excel is used as the local optimizer; this can solve different types of optimization problems, and employs Generalized Reduced Gradient (GRG) method for solving nonlinear problems. GRG method is an efficient local optimizer, and uses finite difference approximation for numerical derivatives of the objective function. A user-friendly interface is developed in a worksheet for coding the objective function and calling the optimization algorithm by anyone who can use the spreadsheet. Decision variables in the problem given by the user are normalized between 0 and 1, inside the program.

3.3.6 Description of IDE Algorithm

IDE begins with the setting of parameter values: population size (NP), learning period (LP), taboo list size (TLS), taboo radius (TR), maximum number of generations (G_{max}) and maximum number of rejections (NR); see the flowchart in Figure 3.3. In the first/initialization step, a population of NP individuals is generated using uniformly distributed random numbers within the search space. The objective function of each individual is evaluated and sent to the taboo list. The best individual is saved.

During each generation, a strategy for each target individual is selected with probability P_{kG} using stochastic universal sampling method. The crossover rate ($Cr_{i,k}$) for each trial individual is calculated based on normal distribution with mean of Crm_k and standard deviation of 0.1. Note that the initial probability of each mutation strategy, P_{kG} is 0.25 and median value of crossover rate for each strategy, Crm_k is 0.5.

When the generation number is higher than the learn period (LP), P_{kG} is updated and median of values stored inside $CRMemory_k$ is used for Crm_k.

For generating a trial individual, mutation factor, F is calculated based on normal distribution with mean of 0.5 and standard deviation of 0.3, and then a trial individual is produced according to assigned mutation strategy (based on stochastic universal sampling and P_{kG}), F and $Cr_{i,k}$. A boundary violation check is performed to make sure the decision variables of the generated trial individual is within the search space. If any bound is violated, the corresponding decision variable of the trial individual is replaced by a randomly generated value within its bounds. The trial individual is then compared with the points in the taboo list. If it is near to any point in the taboo list (i.e., Euclidean distance between the two points in the search space is less than TR), the trial individual is rejected and another point is generated through the mutation and crossover operations.

If the number of rejections for the same trial individual is greater than the specified number, NR, then it means that the individuals in the current population are very close. This indicates the algorithm has either converged to the approximate global optimum or trapped at a local optimum. So, running the global search for more generations is unlikely to improve the solution significantly. Hence, the global search is terminated and the local optimizer is started from the best point found so far. In this study, the Solver tool in Excel is used as the local optimizer. This can solve different types of optimization problems, and employs Generalized Reduced Gradient (GRG) method for solving nonlinear problems. GRG method is an efficient local optimizer and uses finite difference approximation for numerical derivatives of the objective function.

If the number of rejections for the same trial individual is not greater than the specified NR, the global algorithm will continue. After evaluating the objective function of the trial individual produced, the selection step is performed based on the fitness of objective function value. If the trial individual is selected, it replaces the target individual in the population immediately, and $Cr_{i,k}$ is stored into Crm_k and the success of the corresponding mutation strategy is updated as $ns_{k,G} = ns_{k,G} + 1$. Otherwise, the target individual remains in the population and the failure of the corresponding mutation strategy is recorded as $nf_{k,G} = nf_{k,G} + 1$. The objective function is evaluated at the trial individual only if it is away from all the points in the taboo list. After each evaluation, the taboo list is updated dynamically to keep the latest points in the list by replacing the earliest entered point(s). Then, NR is reset to 0 for generating the trial individual for the next target individual until all NP target individuals are covered. The updating of P_{kG}, calculation of Crm_k, mutation, crossover and selection operations are repeated for the next generation until the maximum number of function evaluations or generations, $\mathrm{G}_{\mathrm{max}}$ or the specified stopping criterion is satisfied. Then, the best point obtained over all generations is refined using the local optimizer.

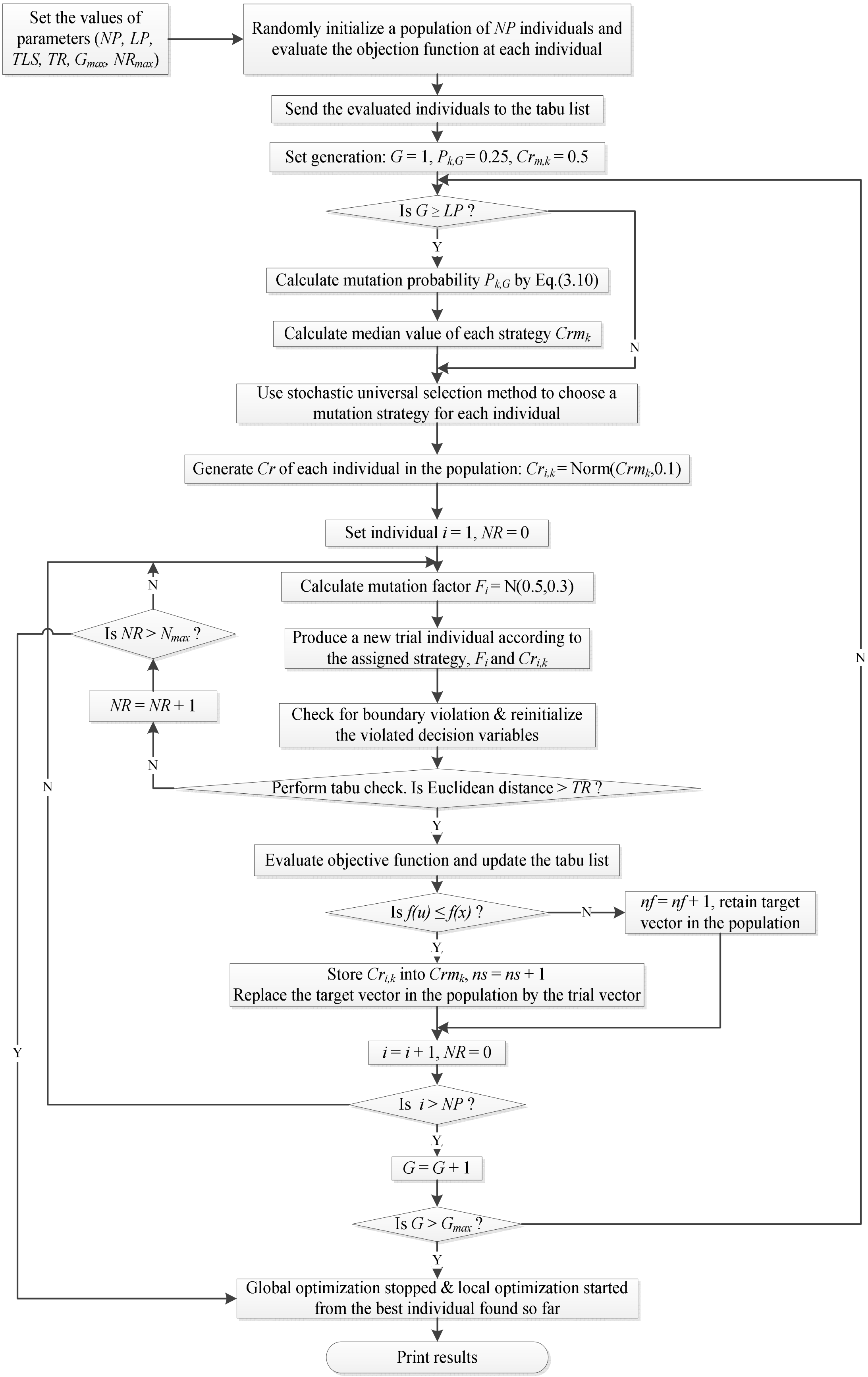

Set the values of parameters (*NP, LP, TLS, TR, G_{max}, NR_{max}*)
Randomly initialize a population of *NP* individuals and evaluate the objection function at each individual
Send the evaluated individuals to the tabu list
Set generation: *G* = 1, $P_{k,G}$ = 0.25, $Cr_{m,k}$ = 0.5
Is *G* ≥ *LP* ?
Y
N
Calculate mutation probability $P_{k,G}$ by Eq.(3.10)
Calculate median value of each strategy Crm_k
Use stochastic universal selection method to choose a mutation strategy for each individual
Generate *Cr* of each individual in the population: $Cr_{i,k}$ = Norm(Crm_k,0.1)
Set individual *i* = 1, *NR* = 0
Calculate mutation factor F_i = N(0.5,0.3)
Produce a new trial individual according to the assigned strategy, F_i and $Cr_{i,k}$
Check for boundary violation & reinitialize the violated decision variables
Perform tabu check. Is Euclidean distance > *TR* ?
N
NR = *NR* + 1
Is *NR* > N_{max} ?
N
Y
N
Y
Evaluate objective function and update the tabu list
Is *f(u)* ≤ *f(x)* ?
N
nf = *nf* + 1, retain target vector in the population
Y
Store $Cr_{i,k}$ into Crm_k, *ns* = *ns* + 1
Replace the target vector in the population by the trial vector
i = *i* + 1, *NR* = 0
Is *i* > *NP* ?
Y
G = *G* + 1
Is *G* > G_{max} ?
N
Y
Global optimization stopped & local optimization started from the best individual found so far
Print results

Figure 3.3. Flowchart of IDE Algorithm.

3.4 Evaluation of IDE on Benchmark Functions

In this section, the proposed algorithm with the novel stopping criterion is tested on many benchmark functions which include shifted, rotated and composite functions. The performance of IDE is compared with some recent state-of-the-art global optimization algorithms in the literature. Finally, the analysis of the proposed stopping criterion is presented.

3.4.1 Comparison of IDE with DETL

3.4.1.1 Benchmark Functions and Evaluation Procedure

The performance of the IDE algorithm is tested on common benchmark functions involving a few to thousands of local optima and 2-20 variables (Srinivas and Rangaiah, 2007b). The details of these benchmark functions are summarized in Table 3.1. As shown in this table, some of these functions are considered to be moderate while the rest as difficult to solve (Srinivas and Rangaiah, 2007b). Out of nine benchmark functions in Table 3.1, five functions (GP, H3, ROS, ZAK and mHB) can be solved by BARON and four functions (ES, SH, RAS and GW) cannot be solved by BARON because of cosine terms in the objective function.

The following parameters were used for all benchmark functions: population size, NP = 30, learning period, LP = 10, tabu list size, TLS = 50, tabu radius, TR = $10^{-3}D$ and $10^{-6}D$ for moderate and difficult functions respectively. Stopping criterion is the satisfaction of either the maximum number of rejections, NR_{max} = 20 or maximum number of function evaluations, G_{max} = 30D and 60D for moderate and difficult functions respectively. All these parameter values are the same as those in Srinivas and Rangaiah (2007b) except NP and LP. The G_{max} is used as another stopping criterion to avoid infinite loops. IDE both with an improvement-based stopping criterion (SC*max*) and the new number of rejections based stopping criterion (*NR*max), will be evaluated.

Table 3.1. Benchmark Problems

Function	Dimension	Domain	Global minimum
Goldstein and Price (GP)[a] $F_{obj} = [1 + (x_1 + x_2 + 1)^2(-14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$-2 \le x_i \le 2$	3.0 at x={0,1}
Easom(ES)[a] $F_{obj} = -\cos(x_1)\cos(x_2)\exp[-((x_1 - \pi)^2 + (x_2 - \pi)^2)]$	2	$-100 \le x_i \le 10$	-1.0 at x={ π , π }
Shubert (SH)[a] $F_{obj} = \left\{\sum_{j=1}^{5} j\cos[(j+1)x_1 + j]\right\}\left\{\sum_{j=1}^{5} j\cos[(j+1)x_2 + j]\right\}$	2	$-10 \le x_i \le 10$	-186.7309 x={0.0217, -0.9527}

Hartmann 3 (H3)[a]
$F_{obj} = -\sum_{j=1}^{4} c_j \exp[-\sum_{i=1}^{n} a_{ji}(x_i - p_{i,j})]$
	3	$0 \le x_i \le 1$	-3.86278 at
			x={0.1146,0
			.5556,
			0.8525}

Rosenbrock (ROSn)[a] $F_{obj} = \sum_{i=1}^{n} [100(x_i^2 - x_{i+1})^2 (x_i - 1)^2]$	2,5,10, 15,20	$-5 \le x_i \le 10$	0.0 at x={1,…,1}
Zakharov (ZAKn)[a] $F_{obj} = (\sum_{i=1}^{n} x_i^2) + (\sum_{i=1}^{n} 0.5ix_i^2)^2 + (\sum_{i=1}^{n} 0.5ix_i^2)^4$	2,5,10, 15,20	$-5 \le x_i \le 10$	0.0 at x={0,…,0}
Modified Himmelbalu (mHB)[b] $F_{obj} = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + 0.1((x_1 - 3)^2 + (x_2 - 2)^2)$	2	$-6 \le x_i \le 6$	0.0 at x={3,2}
Rastrigin (RASn)[b] $F_{obj} = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	2,5,10, 15,20	$-600 \le x_i \le 60$	0.0 at x={0,…,0}
Griewank (GWn)[b] $F_{obj} = \sum_{i=1}^{n} (\frac{x_i^2}{D}) - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	5,10, 15,20	$-600 \le x_i \le 60$	0.0 at x={0,…,0}

[a] Moderate functions; [b] Difficult functions

Since IDE is a stochastic optimization algorithm, convergence speed (computational efficiency) and reliability of finding the global optimum may be affected by random numbers. Hence, 100 independent runs, each time starting from a different random number seed, were performed on each of the benchmark functions. A successful run means the algorithm has found the objective function value very close to the known global optimum value, f(x*). Here, a run of an algorithm is considered to be successful if the objective function value found is ≤ [f(x*) +1e-5] for all benchmark functions. Convergence speed and reliability are assessed using average number of (objective) function evaluations (NFE) and success rate (SR) respectively. NFE for local optimizer is not included here since it is not provided by the Solver tool in Excel; however, our experience shows that local optimization contributes only a small percentage to the total NFE for finding accurate final solutions (Srinivas and Rangaiah, 2007b). Note that NFE is a good indicator of computational efficiency since function evaluation involves extensive computations in application problems. Further, it is independent of the computer and software platform used, and so it is useful for comparison by researchers. The reliability of the algorithm is measured in terms of SR, which is the number of times the algorithm located the global optimum to the specified accuracy out of 100 runs. SR and NFE of the benchmark functions are summarized and discussed in the next section.

3.4.1.2 Results and Discussion

First, IDE and DETL are compared using the same stopping criterion (namely, SC_{max} = 7D for moderate functions and SC_{max} = 12D for difficult functions) used by Srinivas and Rangaiah (2007b) (see Table 3.2). For an overall comparison of algorithms, we report the global success rate (GSR) defined as the mean success rate for all benchmark functions tested (nb):

$$GSR = \sum_{i=1}^{nb} \frac{SR_i}{nb} \qquad (3.11)$$

where SR$_i$ is the success rate for function *i*. The results in Table 3.2 for SC$_{\max}$ criterion show that, out of 21 benchmark functions tested in this study, IDE achieved 100% SR for 16 benchmark functions compared to 100% SR for 10 benchmark functions by DETL. GSR of all benchmark functions listed in Table 3.2 is respectively 97.6% and 99.3% for DETL and IDE. It is obvious that IDE is more reliable than DETL for the benchmark functions. Total NFE used for all the 21 benchmarks is 114,235 and 128,638 for DETL and IDE respectively; thus, total NFE used for IDE is increased by around 12.6%. This may be partly because DETL used different parameter settings for different sets of benchmark functions whereas IDE used the same parameter settings along with parameter adaptation for all benchmark functions tested. In any case, using SC$_{\max}$ criterion, IDE with higher reliability at the expense of slight increase in NFE is preferable to DETL.

Next, IDE is tested using the NR_{max} = 20 stopping criterion. These results in Table 3.2 show that IDE with *NR* stopping criterion has achieved 100% SR for 18 out of 21 benchmark functions tested, and that GSR of IDE with *NR* criterion is the highest (99.8%). Further, NFE taken by IDE with *NR* criterion is less than the other two algorithms for 11 out of 21 problems tested, and total NFE required by the former is 14.7% less than that by DETL. Thus, IDE with $NR_{\max}$ is more reliable and efficient than DETL and IDE with SC$_{\max}$. Comparing the performance of IDE with $NR_{\max}$ and SC$_{\max}$, it can be concluded that the NR_{max} stopping criterion is better in stopping the global search effectively to achieve high reliability with less computational effort. Further, it is more robust than SC$_{\max}$ since the latter required different values for different problems (2007b). Additionally, IDE has fewer parameters to be chosen compared to DETL since two parameters (*F* and *Cr*) which are very sensitive in DE are adapted during the iterations. All these make the IDE algorithm more reliable and efficient for application problems.

Table 3.2. Performance Results of DETL and IDE for Benchmark Functions

Functions	DETL with SC$_{\text{max}}$		IDE with SC$_{\text{max}}$		IDE with NR_{max}	
	SR	NFE	SR	NFE	SR	NFE
GP	100	**636**	100	1084	100	862
ES	95	1409	96	**1306**	100	1482
SH	99	**684**	100	990	100	1279
ROS2	100	**678**	100	883	100	1230
ZAK2	100	**400**	100	912	100	1161
H3	100	719	100	1364	100	**589**
ROS5	99	2671	100	2615	100	**931**
ZAK5	100	**1071**	100	1980	100	3071
ROS10	97	4913	98	5442	99	**1269**
ZAK10	100	**2194**	100	2169	100	5862
ROS20	98	9507	97	11490	98	**3043**
ZAK20	100	4894	100	4748	100	**4378**
mHB	93	1473	100	1308	100	**1279**
RAS5	100	**3592**	100	6469	100	5862
RAS10	95	**7544**	100	10210	100	9234
RAS15	93	12532	97	14530	100	**12155**
RAS20	83	19156	97	**18765**	99	20361
GW5	98	5889	100	8582	100	**3043**
GW10	100	11580	100	12517	100	**4960**
GW15	99	11105	100	11505	100	**6483**
GW20	100	11588	100	9769	100	**7800**
GSR	97.6		99.3		99.8	
Total NFE Used		114235		128638		97388
Total NFE Reduced Compared to DETL(%)				-12.6		14.7

Note: The least NFE required for solving each problem by an algorithm is in bold.

3.4.2 Comparison of IDE with SaDE

3.4.2.1 *Benchmark Functions and Evaluation Procedure*

In this section, the proposed algorithm with the novel stopping criterion is tested on many benchmark functions in Table 3.3. These include shifted, rotated and composite functions which are more challenging to solve. In the shifted functions, position of the global optimum is shifted to a new random position (i.e., global optimum solution has different values for decision variables) but keep the objective function value unchanged.

Table 3.3: Basic Details of 26 Benchmark Functions

Function	Name	Dimension	Global optimum x*	f(x*)	Search range
f1	Shifted Sphere	10 & 30	***o***	0	$[-100,100]^D$
f2	Shifted Schwefel 1.2	10 & 30	***o***	0	$[-100,100]^D$
f3	Rosenbrock	10 & 30	(1,…,1)	0	$[-100,100]^D$
f4	Shifted Schwefel 1.2 with niose	10 & 30	***o***	0	$[-100,100]^D$
f5	Shifted Ackley	10 & 30	***o***	0	$[-32,32]^D$
f6	Shifted rotated Ackley	10 & 30	***o***	0	$[-32,32]^D$
f7	Shifted Griewank	10 & 30	***o***	0	$[-600,600]^D$
f8	Shifted rotated Griewank	10 & 30	***o***	0	$[-600,600]^D$
f9	Shifted Rastrigin	10 & 30	***o***	0	$[-5,5]^D$
f10	Shifted rotated Rastrigin	10 & 30	***o***	0	$[-5,5]^D$
f11	Shifted non-continuous Rastrigin	10 & 30	***o***	0	$[-5,5]^D$
f12	Schwefel	10 & 30	(420.96,…,420.96)	0	$[-500,500]^D$
f13	Composition 1	10 & 30	***o***	0	$[-5,5]^D$
f14	Composition 2	10 & 30	***o***	0	$[-5,5]^D$
f15	Schwefel 2.22	30	(0,…,0)	0	$[-10,10]^D$
f16	Schwefel 2.21	30	(0,…,0)	0	$[-100,100]^D$
f17	Generalized penalized 1	30	(-1,…,-1)	0	$[-50,50]^D$
f18	Generalized penalized 2	30	(1,…,1)	0	$[-50,50]^D$
f19	Kowalik	4	(0.1928,0.1908,0.1231,0.1358)	0.00031	$[-5,5]^D$
f20	Six-hump camel-back	2	(0.8983,-0.7126);(-0.08983,0.7126)	-1.03163	$[-5,5]^D$
f21	Branin	2	(3.142,2.275);(9.425,2.425)	0.398	$[-5,10]^D$
f22	Hartman 1	3	(0.114,0.556,0.853)	-3.86	$[0,1]^D$
f23	Hartman 2	6	(0.201,0.150,0.447,0.275,0.331,0.657)	-3.32	$[0,1]^D$
f24	Shekel's family 1	4	(4,4,4,4)	-10.2	$[0,10]^D$
f25	Shekel's family 2	4	(4,4,4,4)	-10.4	$[0,10]^D$
f26	Shekel's family 3	4	(4,4,4,4)	-10.5	$[0,10]^D$

Remark: ***o*** *is the shifted vector.*

We fixed values of the algorithm parameters for solving all the problems, for fair performance comparison between the proposed IDE algorithm and SaDE algorithm. The following parameter values are used: population size, NP = 50, learning period, LP = 50, taboo list size = 50, The maximum number of function evaluations is just used as a second stopping criterion to prevent infinite iterations. Most of the parameter values used in this study are referring Qin et al. (2009) and Srinivas and Rangaiah (2007b) for fair comparison purpose. In order to make a reliable evaluation of the IDE algorithm, we performed 30 independent runs on each benchmark function. The solution quality, success rate and number of function evaluations (NFE) in these 30 runs are summarized and compared.

3.4.2.2 Results and Discussion

Table 3.4 gives the summary of the results obtained by IDE for the 26 benchmark functions in Table 3.3. These are compared with those of SaDE using the convergence to the known global optimum (i.e., best objective function value ≤ f(x*) +1e-5) as the stopping criterion, given in Qin et al· (2009). SaDE is chosen for comparison because it outperforms the recent self-adaptive DE such as ADE, SDE and jDE (2008). Note that the termination criteria of IDE are NR_{max} and NFE_{max} stated above, and do not require the global optimum of the problem in advance. The stopping criterion of convergence to the known global optimum is very efficient for problems with known global optimum; but it is not practicable and useful for application problems where the global optimum is unknown.

Table 3.4 Comparison of Results by IDE and SaDE for Benchmark Functions

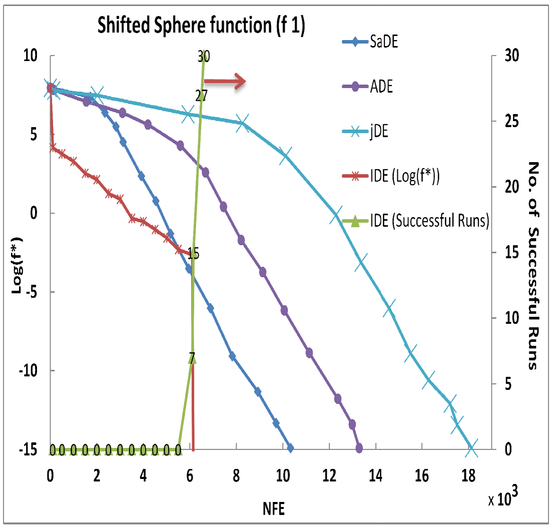
			SaDE		IDE		Reduction in
Dimension	Function	Name	SR	NFE	SR	NFE	NFE (%)
D=10	f1	Shifted Sphere	100	8375	100	6744	19
	f2	Shifted Schwefel 1.2	100	14867	100	17299	-16
	f3	Rosenbrock	100	42446	100	16626	61
	f4	Shifted Schwefel 1.2 with niose	100	15754	0	19317	-
	f5	Shifted Ackley	100	12123	100	5954	51
	f6	Shifted rotated Ackley	100	12244	100	6256	49
	f7	Shifted Griewank	100	35393	100	31867	10
	f8	Shifted rotated Griewank	20	-	47	81924	-
	f9	Shifted Rastrigin	100	23799	100	19207	19
	f10	Shifted rotated Rastrigin	0	100000	0	92122	8
	f11	Shifted non-continuous Rastrigin	100	26945	100	21287	21
	f12	Schwefel	100	16663	100	15722	6
	f13	Composition 1	100	9740	100	5316	45
	f14	Composition 2	80	-	100	18854	-
D=30	f1	Shifted Sphere	100	20184	100	16779	17
	f2	Shifted Schwefel 1.2	100	118743	100	88641	25
	f3	Rosenbrock	90	-	93	36485	-
	f4	Shifted Schwefel 1.2 with niose	0	300000	0	168427	44
	f5	Shifted Ackley	100	26953	100	14542	46
	f6	Shifted rotated Ackley	100	33014	100	15407	53
	f7	Shifted Griewank	80	-	100	22671	-
	f8	Shifted rotated Griewank	40	-	37	42614	-
	f9	Shifted Rastrigin	100	58723	100	55484	6
	f10	Shifted rotated Rastrigin	0	300000	0	224941	25
	f11	Shifted non-continuous Rastrigin	100	77920	100	68266	12
	f12	Schwefel	100	44283	100	45626	-3
	f13	Composition 1	100	19031	100	12227	36
	f14	Composition 2	0	-	0	17140	-
30	f15	Schwefel 2.22	100	25137	100	11707	53
30	f16	Schwefel 2.21	100	88934	100	35653	60
30	f17	Generalized penalized 1	100	18742	100	17516	7
30	f18	Generalized penalized 2	100	19390	100	17125	12
4	f19	Kowalik	100	6426	100	8074	-26
2	f20	Six-hump camel-back	100	2076	100	441	79
2	f21	Branin	100	2614	100	1930	26
4	f22	Hartman 1	100	802	100	79	90
6	f23	Hartman 2	100	3080	100	3022	2
4	f24	Shekel's family 1	100	4947	100	5805	-17
4	f25	Shekel's family 2	100	4173	100	4136	1
4	f26	Shekel's family 3	100	4267	100	4380	-3

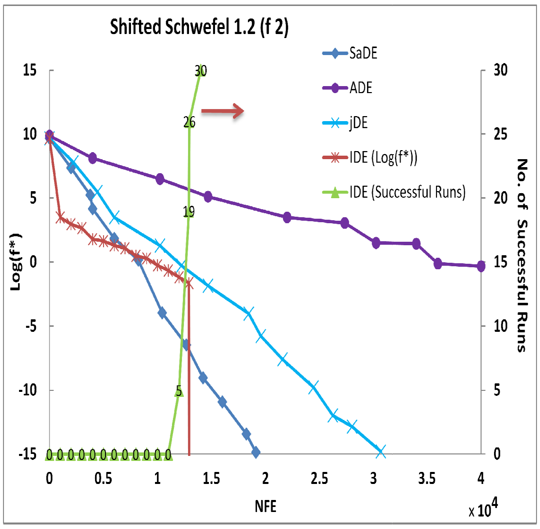
Average reduction in NFE (%) by IDE compared to SaDE **25**

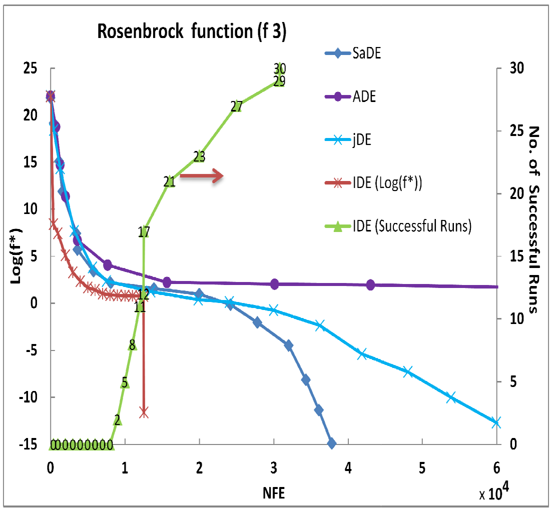
Note: "-" Results are not available for SaDE for these functions.

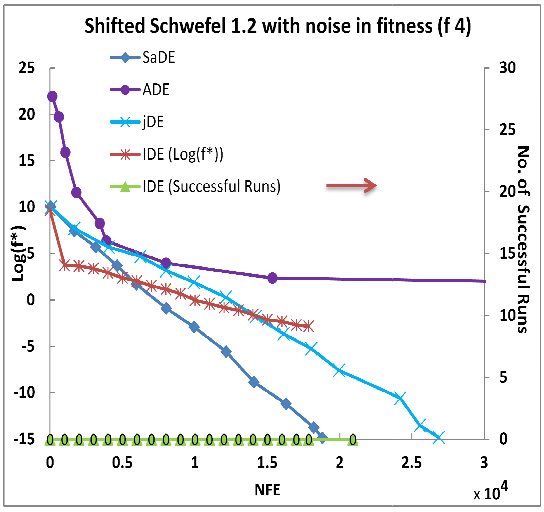
It is obvious from Table 3.4 that IDE often requires fewer NFE compared to SaDE even without knowing the global optimum in advance; it requires slightly more NFE for functions f2, f4, f19 and f24 (i.e., in 4 out of 40 cases studied). The average reduction in NFE by IDE is by 25%, and SR is also slightly better than SaDE. For f4, SR for IDE is shown as 0 in Table 2; this is because local optimizer fails to find the precise minimum due to noise. However, the final value obtained by the global search is near the global minimum in all runs. In effect, SR of IDE can be considered as 100% for f4 as well.

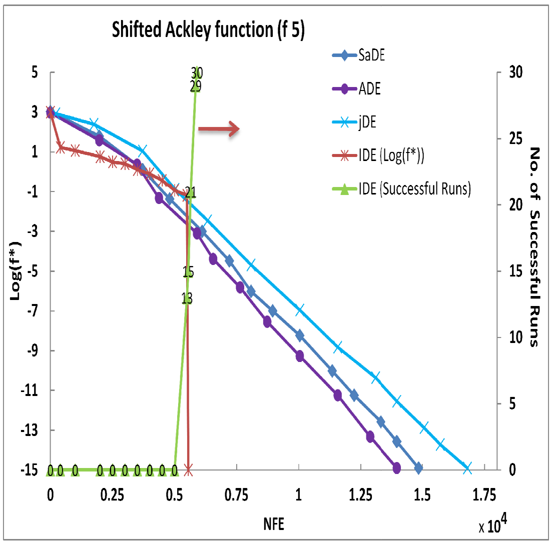
To further confirm the improved performance of IDE, Figure 3.4 shows the convergence characteristics of IDE, SaDE, ADE and jDE on 10D benchmark functions f1-f14. It illustrates the convergence profile in terms of best objective function value of the median run of each algorithm for each function. The data for SaDE, ADE and jDE are taken from Qin et al. (2009). From Figure 3.4, it can be concluded that IDE has faster convergence in early stages, and the convergence speed slows down in the later generations; the former is because of TL and taboo check which increases exploration of new regions. After terminating the global search by the proposed stopping criterion, the local optimizer (Solver) is started to find the precise solution. In Figure 3.4, for functions f1, f2, f3, f6, f8, f13 and f14, IDE shows the fastest convergence over SaDE, ADE and jDE, throughout most of the search. For function f4, SaDE shows the best convergence profile among the four algorithms except in the initial stage where IDE is faster. For functions f7, f9, f11 and f12, ADE shows the fastest convergence among the four algorithms, but its reliability is lower than IDE for some of these functions.

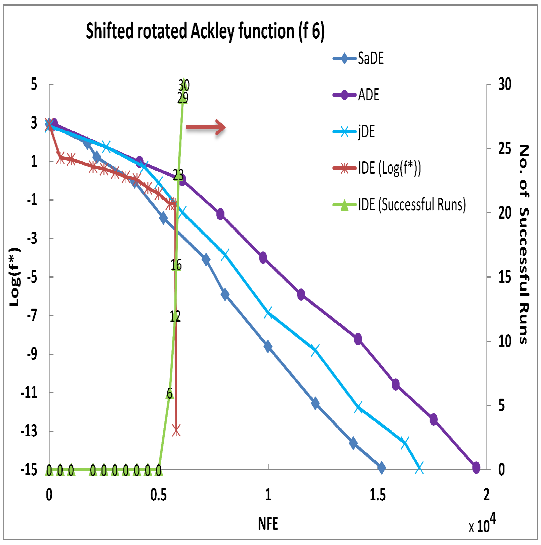

Shifted Sphere function (f 1)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
No. of Successful Runs
NFE
x 10^3
10
5
0
-5
-10
-15
0
2
4
6
8
10
12
14
16
18
30
25
20
15
10
5
0
30
27
15
7
0 0 0 0 0 0 0 0 0 0 0 0

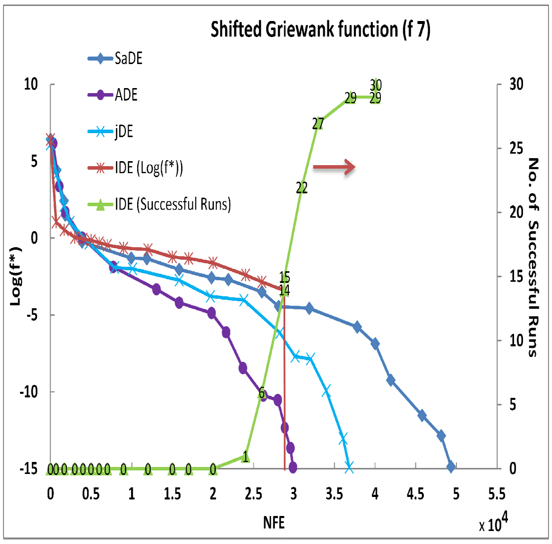

Shifted Schwefel 1.2 (f 2)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
No. of Successful Runs
NFE
x 10^4
15
10
5
0
-5
-10
-15
0
0.5
1
1.5
2
2.5
3
3.5
4
30
25
20
15
10
5
0
30
26
19
5
0 0 0 0 0 0 0 0 0 0 0 0

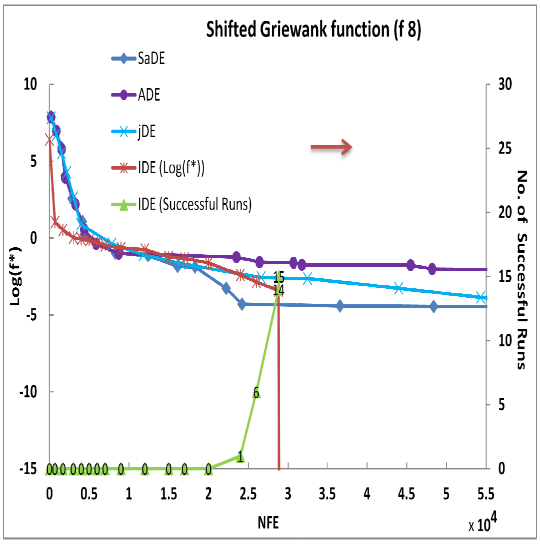

Rosenbrock function (f 3)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
No. of Successful Runs
NFE
x 10^4
25
20
15
10
5
0
-5
-10
-15
0
1
2
3
4
5
6
30
25
20
15
10
5
0
30
29
27
23
21
17
12
11
8
5
2
000000000

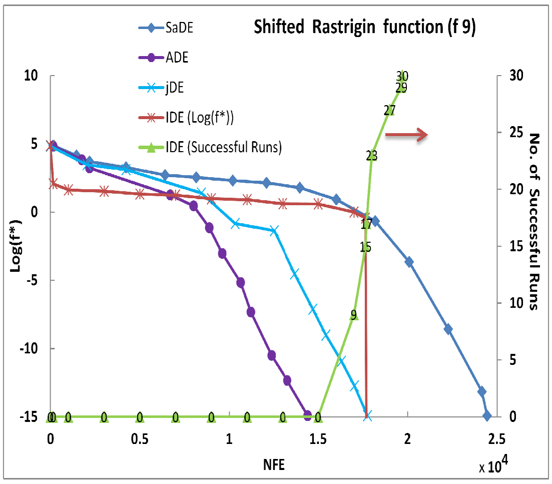

Shifted Schwefel 1.2 with noise in fitness (f 4)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
No. of Successful Runs
NFE
x 10^4
25
20
15
10
5
0
-5
-10
-15
0
0.5
1
1.5
2
2.5
3
30
25
20
15
10
5
0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

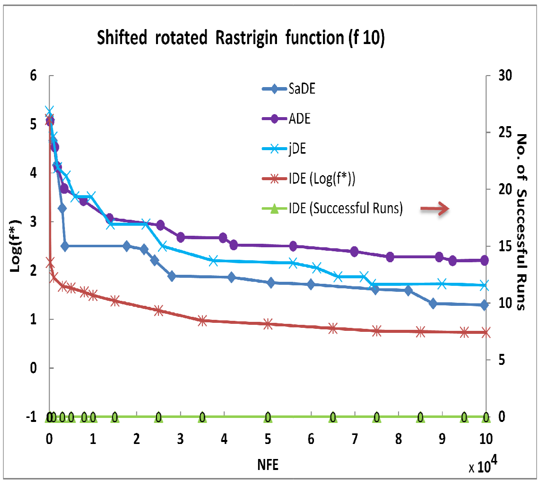

Shifted Ackley function (f 5)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
No. of Successful Runs
NFE
x 10^4
5
3
1
-1
-3
-5
-7
-9
-11
-13
-15
0
0.25
0.5
0.75
1
1.25
1.5
1.75
30
25
20
15
10
5
0
30
29
21
15
13
0 0 0 0 0 0 0 0 0 0 0

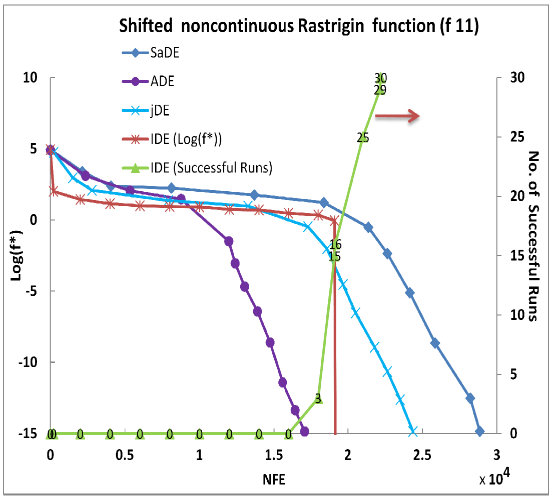

Shifted rotated Ackley function (f 6)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
No. of Successful Runs
NFE
x 10^4
5
3
1
-1
-3
-5
-7
-9
-11
-13
-15
0
0.5
1
1.5
2
30
25
20
15
10
5
0
30
29
23
16
12
6
0 0 0 0 0 0 0 0 0 0

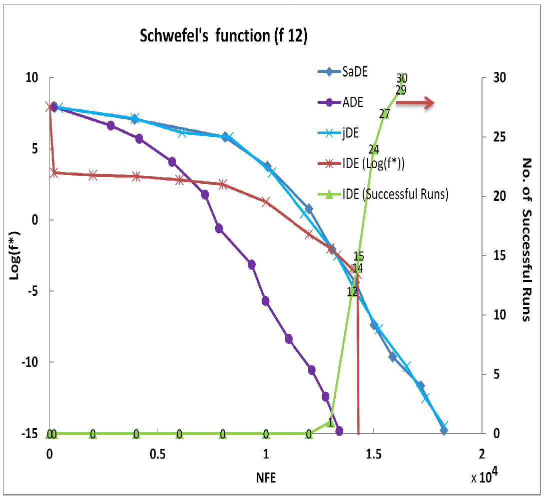

Shifted Griewank function (f 7)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
10
5
0
-5
-10
-15
No. of Successful Runs
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
NFE
x 10^4
30
29
29
27
22
15
14
6
1
0 0 0 0 0 0 0 0 0 0 0 0 0


Shifted Griewank function (f 8)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
10
5
0
-5
-10
-15
No. of Successful Runs
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
NFE
x 10^4
15
14
6
1
0 0 0 0 0 0 0 0 0 0 0 0 0


Shifted Rastrigin function (f 9)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
10
5
0
-5
-10
-15
No. of Successful Runs
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
NFE
x 10^4
30
29
27
23
17
15
9
0 0 0 0 0 0 0 0 0 0


Shifted rotated Rastrigin function (f 10)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
6
5
4
3
2
1
0
-1
No. of Successful Runs
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
NFE
x 10^4
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0


Shifted noncontinuous Rastrigin function (f 11)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
10
5
0
-5
-10
-15
No. of Successful Runs
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
NFE
x 10^4
30
29
25
16
15
3
0 0 0 0 0 0 0 0 0 0


Schwefel's function (f 12)
SaDE
ADE
jDE
IDE (Log(f*))
IDE (Successful Runs)
Log(f*)
10
5
0
-5
-10
-15
No. of Successful Runs
30
25
20
15
10
5
0
0
0.5
1
1.5
2
NFE
x 10^4
30
29
27
24
15
14
12
1
0 0 0 0 0 0 0 0

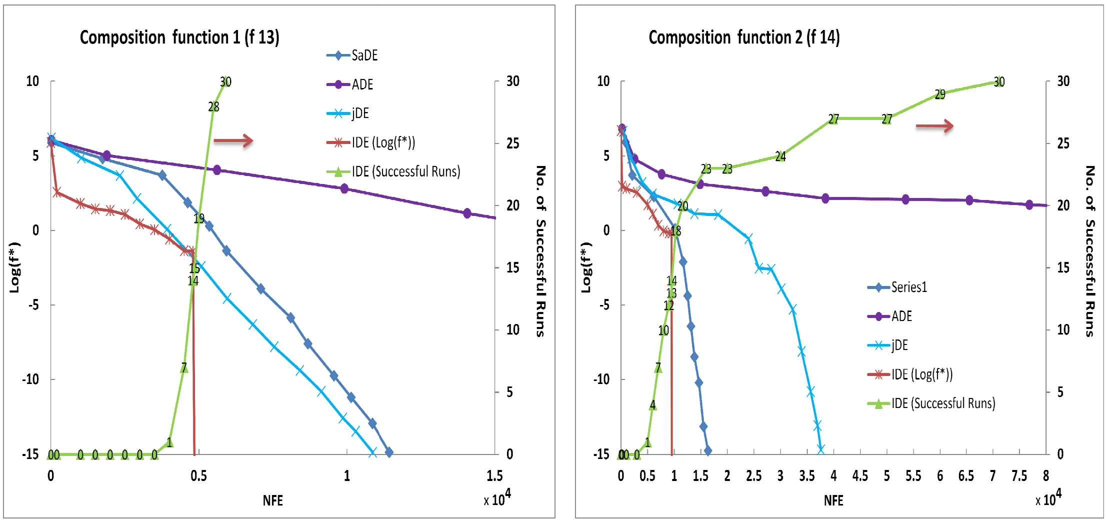


Figure 3.4: Convergence Profiles of IDE, SaDE, ADE and jDE for the Median Trial on 10D Benchmark Functions f1-f14. Results for SaDE, ADE and jDE are Taken from Qin et al. (2009).

From Figure 3.4, it is clear that the local optimizer can significantly increase the convergence speed to the exact optimum. So, the overall convergence speed of IDE is much faster than the other three algorithms compared. The additional line ('IDE (Successful Runs)') in each plot in Figure 3.4 shows the performance of IDE in all 30 runs for each function. It shows the number of successful runs at different NFE. This facilitates better understanding of algorithm's reliability to find the global optimum.

3.4.3 Comparison of IDE with CMA-ES

3.4.3.1 Benchmark Functions and Evaluation Procedure

In order to evaluate the stopping criterion of the proposed algorithm and compare with the state-of-the-art global optimization algorithm other than DE variants, IDE is compared with covariance matrix adaptation evolution strategy (CMA-ES) proposed by Hansen and Kern (2004). CMA-ES is readily available in different programming languages. The 10 commonly used benchmark functions listed in Omran et al. (2009) are used to evaluate and compare the optimization algorithms.

These functions are different from Table 3.1 except Rosenbrock and Camel-back functions, and include five unimodal and five multi-modal functions. All these functions are considered with 30 dimensions except for Camel-back function with 2 variables. IDE uses the termination criterion based on the number of rejections. CMA-ES refers to the updated cmaes.m version 3 (2008) in http://www.lri.fr/~hansen/cmaes_inMATLAB.html, with default settings for the termination.

3.4.3.2 Result and discussion

The results reported in Table 3.5 are SR, mean and standard deviations over 30 trials. The significantly better results are shown in bold in this table. The results in Table 3.5 clearly show that the proposed stopping criterion can achieve 100% success rate for all the 10 test functions. CMA-ES can achieve 100% success rate on only 5 test functions, namely, Sphere, Schwefel 2.22, Rotated hyper-ellipsoid, Griewank and Camel-back; of these, Sphere, Schwefel 2.22 and Rotated hyper-ellipsoid are unimodal functions whereas Griewank and Camel-back are multi-modal functions. For the Step function, CMA-ES has only 17% success rate; it fails to find the global optimum for Schwefel 2.26 and Rastrigin functions. It is clear that the IDE algorithm has better reliability and that the proposed stopping criterion is very robust compared to those in CMA-ES. On the other hand, CMA-ES has faster convergence rate compared to IDE, but it is easily trapped at a local optimum point. It is desirable to have higher reliability to find the global optimum since computational resources are often readily available.

Table 3.5 Comparison of Results by CMA-ES and IDE

	CMA-ES			IDE		
Functions	Mean (*std*)	SR	NFE	Mean (*std*)	SR	NFE
Sphere	0 (0)	100	**6101**	0 (0)	100	14553
Schwefel 2.22	0 (0)	100	10020	0 (0)	100	10634
Rosenbrock	0.1329 (0.7279)	97	46235	**0 (0)**	**100**	**22498**
Step	3.567 (2.725)	17	305	**0 (0)**	**100**	12352
Rotated hyper-ellipsoid	0 (0)	100	**19213**	0 (0)	100	80583
Schwefel 2.26	-118.359 (0)	0	6363	**-12569.5 (0)**	**100**	41475
Rastrigin	18.0419 (4.2029)	0	7065	**0 (0)**	**100**	50586
Ackley	0.031 (0.170)	97	**8970**	**0 (0)**	**100**	12715
Griewank	0 (0)	100	**7100**	0 (0)	100	18190
Camel-back	-1.0316 (0)	100	528	-1.0316 (0)	100	**401**

3.5 Studies on the Effect of NR_{max}

The proposed stopping criterion is based on the number of trial individuals rejected, NR_{max}, which indicates the clustering of the population. Since NR is monitoring the positions of the population instead of the objective function value, it is expected to be more robust and independent of problem dimension and type. In order to explore this and also find suitable value of NR_{max} for the proposed IDE, experiments were performed. Higher NR_{max} will result in higher reliability but lower efficiency, and lower NR will cause lower reliability but higher efficiency. Therefore, a good value of NR_{max} should give high reliability as well as reasonable efficiency. Experiments with NR_{max} = 3, 5, 8, 10 and 13 were conducted on five functions; dimensions of these are 10 and 30.

Table 3.6 Effect of NR_{max} on the Performance of IDE

Function with	3		5		8		10		13	
D in brackets	SR	NFE	SR	NFE	SR	NFE	SR	NFE	SR	NFE
Sphere (10)	100	4826	100	6069	100	6572	100	6850	100	7047
Rosenbrock(10)	100	9820	100	12804	100	16162	100	20065	100	23803
Rastrigin (10)	33	15653	83	17994	100	18690	100	20149	100	21100
Ackley (10)	100	4511	100	5463	100	5840	100	6358	100	6613
Griewank(10)	90	26481	97	28825	100	31463	100	36141	100	42800
Sphere (30)	100	12650	100	13487	100	13996	100	14303	100	14713
Rosenbrock(30)	90	18079	93	20844	100	22753	100	23615	100	24373
Rastrigin (30)	40	41369	93	48852	100	50552	100	50801	100	50934
Ackley (30)	100	10937	100	11954	100	12679	100	12739	100	12953
Griewank (30)	90	16755	93	17227	100	18424	100	18915	100	19282
Total NFE	161081		183519		197131		209936		223618	

The results in Table 3.6 and Figure 3.5 show that when NR_{max} = 3 and 5, the reliability of the algorithm is low. When NR_{max} = 8, 10 and 13, the reliability is as high as 100%, but NR_{max} = 10 and 13 require more NFE (lower efficiency) compared with NR_{max} = 8. Thus, NR_{max} = 8 is used and recommended as the stopping criterion in IDE for problems with up to 30 variables. In stochastic global optimization methods, different parameter settings will affect convergence of speed, and this in turn can affect NR_{max} value. Although NR_{max} can be affected by other parameters, it will be a robust stopping criterion if a higher NR_{max} value is used especially for application problems where the global optimum is unknown.

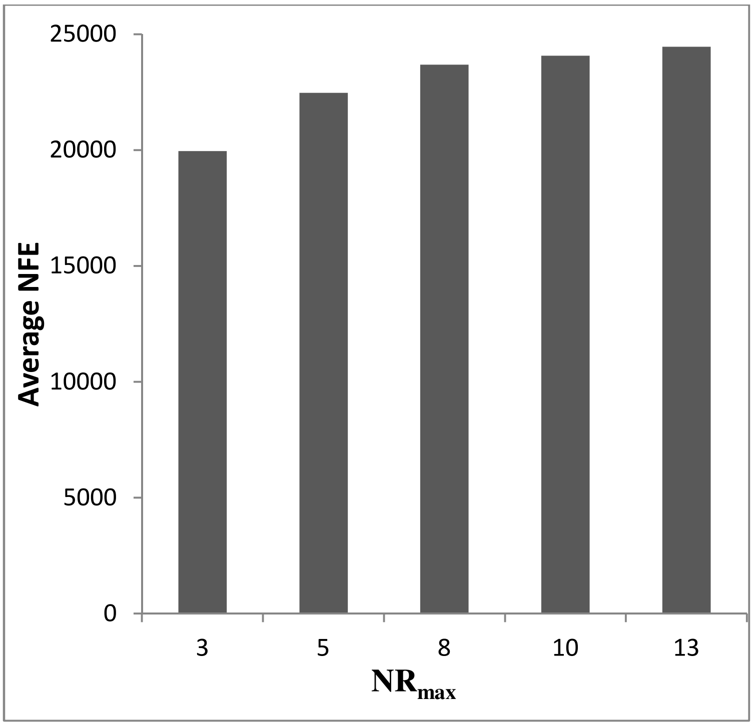


Figure 3.5: Average NFE for the Five Benchmark Functions Versus the Number of Rejected Points (NR$_{max}$)

3.6 Conclusions

The IDE algorithm integrates classic DE with parameter adaptation, tabu list, a new stopping criterion and local optimization after global search. In IDE, where the two control parameters (F and Cr) of DE are automatically tuned and mutation strategies for DE are assigned suitable probability based on the previous generations. The strong feature of TS is integrated with DE in order to prevent re-visiting the same place, to increase the diversity of population and consequently increase the reliability of the algorithm. A new novel stopping criterion based on the number of rejection points is incorporated in IDE to enhance its performance, and a local optimizer is used to efficiently refine the solution obtained by global search. The results obtained on the benchmark functions show that IDE algorithm is superior to the recent algorithms (DETL, SaDE and CMA-ES). In addition, the NR$_{max}$ stopping criterion is effective in terminating the global search at the right time to start the local search. In summary, the proposed IDE integrating DE, taboo check, parameter self-adaptation strategy and the novel stopping criterion is robust, efficient and suitable for application problems.

Chapter 4

Global Optimization of Parameter Estimation Problems*

4.1 Introduction

Parameter estimation in models plays a very important role in developing a better mathematical model which can be used for understanding and analyzing a physical, chemical and/or biochemical system. Parameter estimation in a (thermodynamic) model refers to determining the values of model parameters that best fit the model predictions to the given experimental data. Mathematically, it is formulated as the minimization of a suitable objective function subject to constraints arising from the model equations. For many mathematical models, the objective function and constraints are multi-modal (non-convex). Hence, it is necessary to estimate the model parameters by using a global optimization method because traditional gradient-based optimization methods often provide a local solution only (Gau et al., 2000; Bonilla-Petriciolet et al., 2010).

Many deterministic and stochastic algorithms have been proposed for finding the global optimum, particularly in the past two decades. The former methods can guarantee convergence to the global optimum but they usually require certain properties such as continuity; in some cases, reformulation of the problems is needed depending on the characteristic of the thermodynamic models. In contrast, stochastic methods are quite simple to implement and use, and they do not require any assumptions or transformation of the original problems (Bonilla-Petriciolet et al.,

* This chapter is based on the paper - Zhang, H. and Rangaiah, G.P., A hybrid global optimization algorithm and its application to parameter estimation problems. *Asia Pac. J. Chem. Eng.,* vol. 6, pp. 379-390, 2011.

2010). The stochastic algorithms include simulated annealing, genetic algorithms, tabu search, differential evolution and particle swarm optimization.

Several researchers have studied the parameter estimation in vapor-liquid-equilibrium (VLE) data modeling problems. The challenges and difficulties identified in solving these problems are convergence to a local optimum, flat objective function, badly scaled model functions, non-differential terms in thermodynamic equations and/or large number of optimization variables (Gau et al., 2000 and 2002; Alvarez, 2008; Bonilla-Petriciolet et al., 2010a and 2010b). Thus, it is necessary to use a reliable global optimization method in order to overcome these difficulties.

In the present work, IDE is evaluated on parameter estimation problems in dynamic models in chemical engineering problems such as first-order chain reaction, catalytic cracking of gas oil, methanol-to-hydrocarbon process and Lotka-Volterra problems. Then, it is further applied to parameter estimation in vapor-liquid equilibrium (VLE) data modeling using both least squares and error-in-variable approaches; these application problems involve three different thermodynamic models: NRTL, Wilson, and UNIQUAC. Results obtained by IDE are compared with both deterministic (BARON) and other stochastic methods, and discussed.

The remainder of this chapter is organized as follows. Classical DE is outlined in Section 4.2, and development and description of the IDE are presented in Section 4.3. The performance of IDE on benchmark functions is reported and discussed in Section 4.4. The application of IDE to VLE data modeling problems and performance comparison of IDE with other stochastic global algorithms and BARON are presented in Section 4.5. Finally, Section 4.6 concludes this chapter.

4.2 Solving Parameter Estimation Problems in Dynamic Systems

Many chemical engineering applications involve dynamic models. Hence, application of global optimization techniques to estimate parameters in dynamic systems has received greater attention in the recent past. Parameter estimation involves fitting the model to experiment data by minimizing the objective function such as sum of squared errors (SSE). So, parameter estimation problem in a dynamic model is given by:

$$\min \ SSE = \sum_{i=1}^{n} \sum_{j=1}^{NS} (y_{i,j} - y_{i,j}^{e})^2 \tag{4.1}$$

subject to the model constraints and bounds on parameters:

$$\frac{d(y_i)}{dt} = f_i(y, \theta) \tag{4.2}$$

$$\theta^{LB} \le \theta \le \theta^{UB} \tag{4.3}$$

where n is the number of quantities and NS is the number of sampled data points. θ denotes D-dimensional parameters, $y_{i,j}$ is the calculated value for ith quantity at jth time and $y_{i,j}^{e}$ is the corresponding experimental data. $\theta^{LB}$and θ^{UB} are the lower and upper bounds on the parameters. Note that the D-parameters are the decision variables in parameter estimation problems.

The solution of the above type of optimization problems is usually very difficult due to their highly nonlinear and complex nature. Local optimization methods may get trapped at a local optimum depending on the degree of nonlinearity and initial guess. Some deterministic global optimization algorithms have been applied to parameter estimation problems (Esposito and Floudas, 2000; Papamichail and Adjiman, 2000). Although these methods can guarantee finding the global optimum, some difficulties in using them were stated in the literature (Angira and

Santosh, 2007). Several researchers have studied stochastic global optimization methods for solving parameter estimation problems in dynamic models (Kapadi and Gudi, 2004; Katare et al., 2004; Angira and Santosh, 2007; Srinivas and Rangaiah, 2007b). In this study, IDE is evaluated for parameter estimation in dynamic systems.

4.2.1 Evaluation Procedure

The application problems (i.e., differential equations and bounds on parameters) are summarized in Table 4.1. The IDE for solving these parameter estimation problems is implemented in MATLAB platform, and *fmincon* in MATLAB's optimization toolbox is used as the local optimizer. The reason for using MATLAB is that Excel does not have an efficient program for solving stiff ordinary differential equations (ODE). The ODE solver (*ode15s*) is used to solve stiff ODEs in the parameter estimation problems tested. The performance of IDE is compared with that of DETL; in order to make a fair comparison, NP used (= 30) in IDE is the same as that in DETL (Srinivas and Rangaiah, 2007b), and the rest of the parameters are kept same at the previous settlings. Each parameter estimation problem is solved 100 times independently as in Srinivas and Rangaiah (2007b). The performance of IDE is compared with that of DETL based on SR, NFE and stopping criterion. The stopping criterion used in DETL is the maximum number of iterations (SC_{max} = 7D) without improvement in the best objective function value found so far, which is also known as an improvement-based stopping criterion. The stopping criterion in IDE is NR_{max} (= 8) as in the benchmark functions studied above.

Table 4.1 Dynamic System Parameter Estimation Problems

No.	Application	Differential equations	Parameter bounds	Number of data points available
p1	First –order irreversible chain reaction	$\frac{dz_1}{dt} = -\theta_1 z_1$ $\frac{dz_2}{dt} = \theta_1 z_1 - \theta_2 z_2$	$0 \le \theta \le 10$	10
p2	First –order reversible chain reaction	$\frac{dz_1}{dt} = -\theta_1 z_1 + \theta_2 z_2$ $\frac{dz_2}{dt} = \theta_1 z_1 - (\theta_2 + \theta_3) z_2 + \theta_4 z_3$ $\frac{dz_1}{dt} = \theta_2 z_2 - \theta_4 z_3$	$0 \le \theta_1, \theta_2 \le 10$ $10 \le \theta_3, \theta_4 \le 5$	20
p3	Catalytic cracking of gas oil	$\frac{dz_1}{dt} = -(\theta_1 + \theta_2) z_1^2$ $\frac{dz_2}{dt} = \theta_1 z_1^2 - \theta_2 z_2$	$0 \le \theta \le 20$	20
p4	Methanol-to-hydrocarbon process	$\frac{dz_1}{dt} = -(2\theta_1 - \frac{\theta_1 z_2}{(\theta_2 + \theta_5) z_1 + z_2} + \theta_3 + \theta_4) z_1$ $\frac{dz_2}{dt} = \frac{\theta_1 z_1 (\theta_2 z_1 - z_2)}{(\theta_2 + \theta_5) z_1 + z_2} - \theta_3 z_1$ $\frac{dz_2}{dt} = \frac{\theta_1 z_1 (z_2 + \theta_5 z_1)}{(\theta_2 + \theta_5) z_1 + z_2} + \theta_4 z_1$	$0 \le \theta \le 20$	16
p5	Lotka-Volterra problem	$\frac{dz_1}{dt} = \theta_1 z_1 (1 - z_2)$ $\frac{dz_2}{dt} = \theta_2 z_2 (z_1 - 1)$	$0 \le \theta \le 1$	10

4.2.2 Results and Discussion

The results of solving the parameter estimation problems in dynamic models are summarized in Table 4.2. It shows that SR of IDE for all examples tested is 100%. The p5 problem is highly non-convex and difficult to solve as shown in Esposito and Floudas (2000). Results in Table 4.2 for this problem show that IDE algorithm has higher reliability compared to DETL. The reduction in NFE required for solving problems p1 to p5 by IDE is 58%, 24%, 60%, 0.2% and -11% respectively compared to DETL. For p5, IDE requires more NFE because of the difficult nature of the problem; however, it achieves 100% success rate. On average, IDE requires 26% fewer NFE compared to DETL, which shows that IDE has faster convergence and is an efficient algorithm. Thus, the proposed stopping criterion based on $\mathrm{NR}_{\mathrm{max}}$ is able to stop the algorithm effectively, which makes IDE more suitable for application problems.

Table 4.2 Results for Parameter Estimation Problems

Problem No.	DETL		IDE		Reduction in NFE (%)
	SR	NFE	SR	NFE	
p1	100	1015	100	433	57
p2	100	5025	100	3824	24
p3	100	3490	100	1408	60
p4	100	6137	100	6124	0.2
p5	96	1971	100	2192	-11
Average reduction in NFE (%) by IDE compared to DETL					**26**

4.3 Solving Parameter Estimation Problems for VLE Modeling

4.3.1 Least Squares Approach

Consider a set of observations q_{ij} of i = 1, 2, …, m response variables from j = 1, 2, …, $ndat$ experiments, where the responses can be expressed by an explicit model $f_i(r_j,\theta)$ with nl independent variables $r_j = (r_{1,j},...,r_{nl,j})$ and $npar$ parameters $\theta = (\theta_1,...,\theta_{npar})$. Measurement errors in $\boldsymbol{r}_j$ are neglected in LS approach. There are different objective functions that can be used to estimate the parameter values that provide the best fit for a model. In this study, the following objective function involving fractional errors is used since it tends to weigh errors in small and large quantities equally.

$$F_{obj} = \sum_{j=1}^{ndat} \sum_{i=1}^{m} \left(\frac{q_{ij} - f_i(r_j,\theta)}{q_{ij}} \right)^2 \tag{4.4}$$

The decision variables in the LS approach are *npar* parameters θ .

In the case of VLE data, excess Gibbs energy equation is normally used for phase equilibrium modeling. Hence, the objective function used in Eq. (4.4) can be modified as follows to fit the activity coefficient data:

$$F_{obj} = \sum_{j=1}^{ndat} \sum_{i=1}^{c} \left(\frac{\gamma_{i,j}^{\exp} - \gamma_{ij}^{calc}}{\gamma_{i,j}^{\exp}} \right)^2 \tag{4.5}$$

where $\gamma_{i,j}^{\exp}$ is the experimental value for the activity coefficient of component *i* in *j*th experiment, γ_{ij}^{calc} is the calculated value for the activity coefficient of component *i* in *j*th experiment, and *c* is the number of components in the mixture. $\gamma_i^{\exp}$ can be calculated from VLE data as follows:

$$\gamma_i^{\exp} = \frac{y_i^{\exp} P}{x_i^{\exp} P_i^0} \qquad i = 1,...,c \tag{4.6}$$

where $x_i^{\exp}$ and $y_i^{\exp}$ are, respectively, the measured mole fraction of component *i* in the liquid and vapor phases at equilibrium, P_i^0 is the vapor pressure of pure component *i* at the system temperature ***T*** and P is the pressure of the system. For Eq. (4.6), it is assumed that, at low pressure, the fugacity coefficients of pure components cancel each other and the values of Poynting corrections are very close to one.

Non-Random-Two-Liquid (NRTL), Wilson and UNIversal QUAsi-Chemical (UNIQUAC) models have been used to evaluate γ_i^{calc}, and Eq. (4.5) is optimized by changing the energy parameters (θ) of these models. Owing to the explicit nature of the equations for γ_i^{calc}, global minimization of LS objective function, Eq. (4.5) can be solved as an unconstrained optimization problem. Details of the test problems used for LS approach are summarized in Table 4.3.

Table 4.3. Details of VLE-LS Problems

No.	System	Data	Models	Variables	Global optimum	Reference
LS-1	tert butanol + 1 butanol	P = 100 mm Hg $ndat$ = 9	Wilson and Ideal gas models	D = 2; $\theta_1, \theta_2 \in$ (-8500, 320000)	F_{obj} = 0.01026; θ_1 = -567.96 θ_2 = 745.33	Alvarez et al., 2008
LS-2		P = 700 mm Hg $ndat$ = 9			F_{obj} = 0.013690 ; θ_1 = -733.95 θ_2 = 1318.23	Bonilla-Petriciolet et al., 2007
LS-3		P = 500 mm Hg $ndat$ = 9			F_{obj} = 0.006852; θ_1 = -718.01 θ_2 = 1264.74	Gau et al.,2000
LS-4	water + 1,2 ethanediol	P = 430 mm Hg $ndat$ = 18	Wilson and Ideal gas models	D = 2; $\theta_1, \theta_2 \in$ (-8500, 320000)	F_{obj} = 1.039134; θ_1 = 5072.361 θ_2 = -1921.62	Bonilla-Petriciolet et al., 2007
LS-5			UNIQUAC Ideal gas models	D = 2; $\theta_1, \theta_2 \in$ (-5000, 20000)	F_{obj} = 1.408547; θ_1 = -1131.84 θ_2 = 3617.65	Bonilla-Petriciolet et al., 2007
LS-6			NRTL and Ideal gas models	D = 3; $\theta_1, \theta_2 \in$ (-2000, 5000); $\alpha_{12} \in$ (0.01, 10.0)	F_{obj} = 1.253531; θ_1 = -678.99 θ_2 = 3046.13; α_{12} = 0.621375	Bonilla-Petriciolet et al., 2007
LS-7	benzene + hexafluoro-benzene	T = 50 °C $ndat$ = 11	Wilson and Ideal gas models	D = 2 $\theta_1, \theta_2 \in$ (-8500, 320000)	F_{obj} = 0.008935; θ_1 = -424.08 θ_2 = 983.06	Bollas et al., 2009
LS-8		P = 300 mm Hg $ndat$ = 17			F_{obj} = 0.014860 ; θ_1 = -432.49 θ_2 = 992.85	Bonilla-Petriciolet et al., 2007
LS-9		P = 760 mm Hg $ndat$ = 29			F_{obj} = 0.014616; θ_1 = -334.70 θ_2 = 704.74	Gau et al., 2000
LS-10		T = 30 °C $ndat$ = 10			F_{obj} = 0.011783; θ_1 = -467.76 θ_2 = 1313.94	Bonilla-Petriciolet et al., 2007

4.3.2 Error-in-variables Approach

Unlike LS approach, EIV approach considers errors in the state variables $z_{i,j}$ for the experiments of the system to be modeled (namely, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{T}$ and $\boldsymbol{P}$ for VLE data modeling). This results in the following objective function:

$$F_{obj} = \sum_{j=1}^{ndat} \sum_{i=1}^{nest} \frac{z_{i,j}^{t} - z_{i,j}}{\sigma_i^2} \qquad (4.7)$$

subject to

$$g(z_{i,j}^{t}, \theta) = 0 \quad i = 1,..., nest \quad j = 1,..., ndat \qquad (4.8)$$

Here, *g* is a vector of model functions, *nest* is the number of state variables, $z_{i,j}^{t}$ is the unknown "true" value of i^{th} state variable in j^{th} experiment and σ_i is the standard deviation associated with the measurement of i^{th} state variable. The decision variables of EIV formulation include the set of $z_{i,j}^{t}$ values in addition to *npar* parameters $\boldsymbol{\theta}$ in the model. This leads to significant increase in the dimensionality of the optimization problem depending on the number of experimental data. Bounds for state variables are usually three standard deviations away from the measured values.

Assuming the experiment is conducted at low pressure, VLE problems can be defined by the following equations:

$$P = \sum_{i=1}^{c} (\gamma_i x_i P_i^0) \qquad (4.9)$$

$$y_j = \frac{\gamma_i x_i P_i^0}{\sum_{j=1}^{c} (\gamma_j x_j P_j^0)} \qquad i = 1, \ldots, c \qquad (4.10)$$

The above equations allow the VLE problems to be solved as unconstrained optimization problems through their substitution for P^t and y_i^t in Eq. (4.7) to give the following objective function for EIV optimization problem.

$$F_{obj} = \sum_{j=1}^{ndat} \sum_{i=1}^{c} \left(\frac{(x_{i,j}^{t} - x_{i,j}^{\exp})^2}{\sigma_{x_i}^2} + \frac{(y_{i,j}^{t} - y_{i,j}^{\exp})^2}{\sigma_{y_i}^2} + \frac{(T_j^{t} - T_j^{\exp})^2}{\sigma_T^2} + \frac{(P_j^{t} - P_j^{\exp})^2}{\sigma_P^2} \right) \qquad (4.11)$$

The independent variables are the set, $z = (x_i^{\exp}, T^{\exp})$ for all the measurements. The decision variables for optimization are the set of $z^t = (x_i^t, T^t)$ and $\theta = (\theta_1, ..., \theta_{npar})$, and the total number of decision variables is *npar* + *c* × *ndat*. Both LS and EIV formulations involve complex, non-linear thermodynamic models, which results in non-linear and potentially non-convex objective function with local optima. Details of VLE-EIV problems studied are given in Table 4.4.

Table 4.4. Details of VLE-EIV Problems

No.	Data	Variables	Global optimum	Reference
EIV-1	P = 500 mmHg *ndat* = 16 ; σ(0.003, 0.0029, 1.7, 0.083)	D = 34 θ_1, $\theta_2 \in$ (-10000, 200000); $x_{ij}^t \in (x_{ij} - 3\sigma, x_{ij} + 3\sigma)$ $T^t \in (T - 3\sigma, T + 3\sigma)$	F_{obj} = 19.998720 θ_1 = -429.85 θ_2 =1029.32	Bonilla-Petriciolet et al., 2007
EIV-2	P = 300 ; mmHg ;*ndat* = 17 ; σ(0.003, 0.0029, 1.7, 0.083)	D = 36	F_{obj} = 42.343724 θ_1 = -437.72 θ_2 =1003.12	Esposito and Floudas, 1998
EIV-3	T = 30 °C; *ndat* = 10; σ(0.001, 0.01, 0.75, 0.1)	D = 22	F_{obj} = 11.898795 θ_1 = -472.00 θ_2 =1274.32	Gau and Stadtherr, 2002
EIV-4	T = 40 °C; *ndat* = 10; σ(0.001, 0.01, 0.75, 0.1)	D = 22	F_{obj} = 11.170496 θ_1 = -462.51 θ_2 =1197.31	Srinivas and Rangaiah, 2006
EIV-5	T = 50 °C; *ndat* = 11; σ(0.003, 0.0029, 1.7, 0.083)	D = 24	F_{obj} = 25.671042 θ_1 = -415.76 θ_2 = 944.57	Bonilla-Petriciolet et al., 2007
EIV-6	T = 60 °C**;** *ndat* = 10 σ(0.003, 0.0029, 1.7, 0.083)	D = 22	F_{obj} = 19.401593 θ_1 = -437.05 θ_2 =1065.04	Bonilla-Petriciolet et al., 2007

Note: The system used is benzene + hexafluorobenzene, and the models are Wilson and ideal gas. The bounds of variables used for all the EIV problems as given in EIV-1.

4.3.3. Test Examples and Evaluation Procedure

Altogether, there are 20 VLE problems, consisting of 10 LS and 10 EIV problems respectively (Tables 4.3, 4.4 and 4.10). All these problems are multi-modal, and their number of decision variables ranges from 2 to 60. Each VLE problem is solved 100 times independently. A trial/run is considered successful if the best objective function value obtained is within 1.0E-5 from the known global optimum. Also, GSR of different algorithms is reported for the LS and EIV problems. The performance (i.e., SR and NFE) of IDE is compared for four stopping criteria: SC-1 (i.e., SC_{max} = 6D) and SC-2 (i.e., SC_{max} = 12D) are based on the maximum number of iterations without improvement in the best objective function value, maximum number of rejected points (NR_{max} = 20), and maximum number of iterations (G_{max} = 1000) to find the success rate for a large number of iterations. The results of IDE are also compared with those by other stochastic algorithms and a deterministic algorithm.

4.3.4. Results and Discussion Using Least Squares Approach

4.3.4.1 Performance of IDE with Different Stopping Criteria and Population Size

In order to show the robustness of the IDE algorithm for VLE-LS problems, all the algorithm parameters are kept same except that the stopping criteria (SC-1 and SC-2) are the same as those in Bonilla-Petriciolet et al.(2010c). The SR, GSR, NFE, total NFE and CPU time (in seconds) for NR_{max} and G_{max} of the IDE algorithm with four stopping criteria used for the ten VLE-LS problems are summarized in Table 4.5. Recall that VBA is used in the present study because of its ready availability and used by researchers and practitioners in diverse fields. However, based on our experience, it is about one order of magnitude slower than the equivalent MATLAB code.

It is obvious from Table 4.5 that IDE with SC-1 has the lowest SR and uses the least NFE. IDE with SC-2 has better SR but requires higher NFE compared to SC-1. IDE with *NR* obtains very good SR although NFE is higher than that with SC-1 and SC-2 but much lower than G_{max}. This is due to the nature of the stopping criteria where SC-1 and SC-2 will force the algorithm to stop when the best objective function value in the population fails to improve after a certain number of iterations. NR_{max} can overcome this difficulty since it is monitoring the convergence of the whole population instead of the best objective function value. IDE with G_{max} obtains the best SR but it needs much more NFE.

Table 4.5. NFE and SR of IDE with Four Different Stopping Criteria: SC-1, SC-2,NR_{max} and G_{max} for Solving VLE-LS problems with *NP* =30.

No.	SC-1 = 6D		SC-2 = 12D		NR_{max} = 20			G_{max} = 1000		
	SR	NFE	SR	NFE	SR	NFE	CPU*	SR	NFE	CPU*
1	16	1044	49	2156	62	6238	19	66	30000	109
2	13	1001	44	2025	66	8656	26	69	30000	98
3	17	1095	44	2032	65	5618	18	68	30000	106
4	27	1197	58	2374	79	7031	16	79	30000	86
5	52	1112	58	2746	61	6165	22	62	30000	105
6	72	1843	75	3948	79	4599	20	79	30000	111
7	25	1260	43	1962	75	7271	18	75	30000	92
8	23	1032	58	2174	73	6667	18	74	30000	102
9	40	1071	70	1956	100	7898	20	100	30000	95
10	19	983	46	1980	75	6627	19	75	30000	91
GSR	30.4		54.5		73.5			74.7		
Total NFE		1.16E+04		2.34E+04		6.68E+04			3.00E+05	

* CPU time in seconds

In general, the higher the iteration level, the better the results are but the additional iterations with no significant improvement in the results are waste of computational resources. For example, for LS-4, IDE with NR_{max} can achieve 79% SR with NFE of 7031, whereas IDE with G_{max} obtained the same SR but with NFE of 30000. Thus, it is essential for the optimization algorithm to stop at the right time incurring least computational resources without compromising reliability of finding the global optimum. Results in Table 5 indicate that the IDE with NR_{max} termination criterion can stop the global search effectively. They further indicate that SR improves with increasing number of NFE initially; but, once the algorithm is converged, increasing NFE will not improve SR.

GSR and total NFE for all ten LS problems using 4 different stopping criteria in IDE is reported in Table 4.5. It is clear that IDE with NR_{max} and G_{max} can achieve better reliability compared with IDE with SC-1 and SC-2. IDE with NR_{max} obtained similar good GSR as by IDE with G_{max}, but IDE with NR_{max} required much less total NFE. Furthermore, G_{max} is difficult to choose for application problems especially for unknown problems. Very low G_{max} value will increase the probability of trapping in local optimum, and very high G_{max} value will waste computational resources. In order to show the implication of the trade-offs between reductions in NFE and SR between IDE either G_{max} or IDE with NR_{max}, CPU time for each problem is reported in Table 5. It is clear that IDE with NR_{max} uses much less computation time compared to IDE with G_{max}. Overall, the IDE with NR_{max} stopping criterion has better reliability and efficiency.

Population size is another control parameter which can affect the performance of the stochastic algorithms. The effect of population size on VLE-LS problems is studied using the IDE with *NP* of 50D and different stopping criteria. Comparing these results shown in Table 6 with those in Table 4.5, it can be seen that both SR and NFE are increased with *NP* = 50D. This is expected because, when the population size is increased, the algorithm can search more thoroughly but requires more computational effort.

Results for IDE with each of the four stopping criteria (Table 4.6) indicate that the IDE with NR_{max} can stop the algorithm effectively. This observation is consistent with the results in Table 4.5. The overall performance of IDE with each of four different stopping criteria is shown in Table. 4.6. It is clear that IDE with NR_{max} and G_{max} can achieve better reliability compared to IDE with SC-1 and SC-2. IDE with NR_{max} obtained good GSR similar to IDE with G_{max} but with only one quarter NFE. Thus, the stopping criterion (NR_{max}) is very efficient and robust than other stopping criteria, and performs consistently well for higher population size as well.

Table 4.6. NFE and SR of IDE with Four Different Stopping Criteria: SC-1, SC-2, NR_{max} and G_{max} for Solving VLE-LS problems with *NP* = 50D

	SC-1 = 6D		SC-2 = 12D		NR_{max} = 20		G_{max} = 1000	
No.	SR	NFE	SR	NFE	SR	NFE	SR	NFE
1	26	3189	51	6814	85	24921	90	100000
2	27	3091	60	6855	88	29680	95	100000
3	29	3266	51	7197	74	26729	93	100000
4	39	3361	74	7370	97	23616	97	100000
5	77	3661	76	6494	79	18957	80	100000
6	99	10260	100	21568	100	14701	100	150000
7	26	3325	68	7581	97	24896	97	100000
8	31	3297	55	7376	89	24003	94	100000
9	46	3396	87	6844	100	28007	100	100000
10	34	3112	62	6821	91	31497	93	100000
GSR	43.4		68.4		90		93.9	
Total NFE		4.00E+04		8.49E+04		2.47E+05		1.05E+06

4.3.4.2 Comparison of IDE with Other Stochastic Methods

For the ten VLE-LS problems, GSR of the IDE algorithm with SC-1, SC-2 and NR_{max} is compared with that of other stochastic algorithms, namely, SA, DE, DETL and PSO in Fig. 4; in all these algorithms, NP = 30. It is clear that the overall reliability (GSR) of IDE for VLE-LS problems is superior to SA, DE, DETL and PSO with SC-1 and SC-2. As expected, Fig. 4.1 shows that all the algorithms obtained better GSR when SC-2 is used instead of SC-1 but this is at the expense of computational effort (i.e., more NFE as can be seen in Table 5 for IDE). IDE with NR_{max} obtained the best GSR compared to other methods and stopping criteria. This again shows that IDE with NR_{max} is very reliable for solving VLE-LS problems.

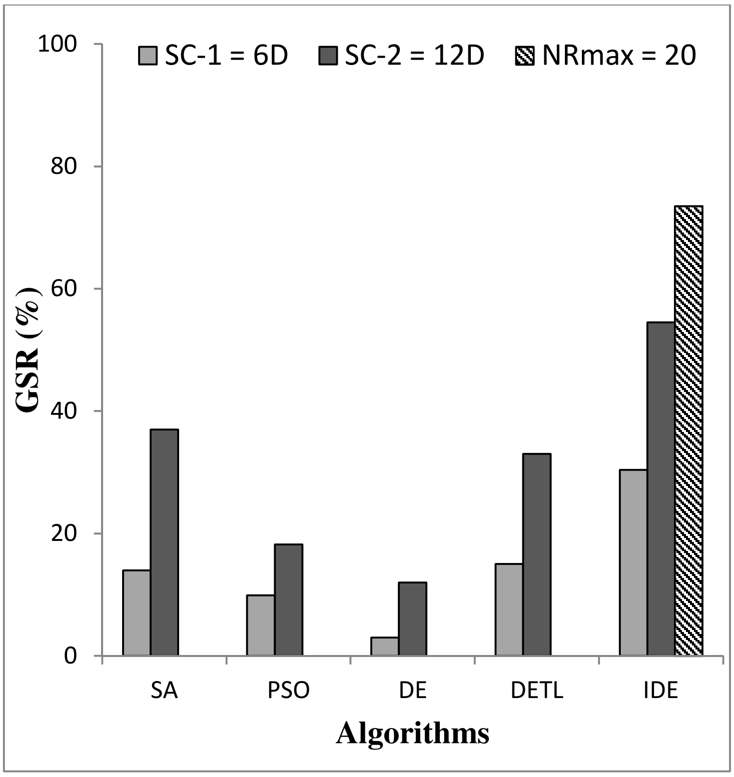


Figure 4.1. Global Success Rate (GSR) for SA, PSO, DE, DETL and IDE with Different Stopping Criteria for VLE-LS Problems Using NP = 30.

In order to illustrate the robustness of the IDE algorithm, Fig. 4.2 compares its GSR with that of SA, PSO, DE and DETL at different iteration levels without stopping criterion. For PSO, DE, DETL and IDE, population (NP) = 50D is used. For

SA, $NS \times NT = 50D$ is used. Note that NT is the number of iterations before the reduction of annealing temperature and NS is the number of cycles for updating the decision variables. At different iteration levels, local optimizer is used to find the accurate optimum. Fig. 5 shows that IDE achieves slightly worse GSR compared to SA and DETL at 50 generations. However, at higher iterations, IDE can give very good GSR comparable to or better than SA, DE, DETL and PSO. Interestingly, SA achieves the best GSR at all different iteration levels.

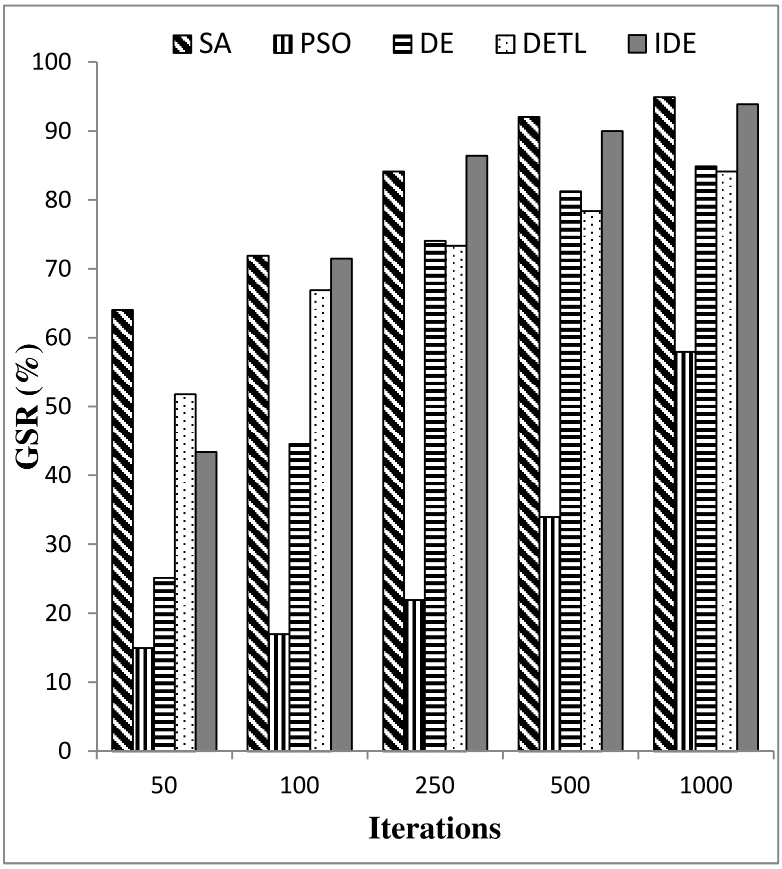


Figure 4.2. Global Success Rate (GSR) for SA, PSO, DE, DETL and IDE at Different Iteration Level for VLE-LS Problems Using NP = 50D.

4.3.4.3 Comparison of IDE with a Deterministic Method

BARON is a deterministic global optimization solver, developed by Sahinidis and Tawarmalani (2010) and available in GAMS(Rosenthal, 2010). Purely continuous, purely integer and mixed-integer nonlinear model types can be solved using this deterministic method. BARON combines constraint propagation, interval analysis and duality approaches for problem reduction along with enhanced branch and bound concepts. It can solve a wide range of optimization problems but it cannot solve the problems containing trigonometric functions (Rosenthal, 2010; Sahinidis and Tawarmalani, 2010). However, it requires the some reformulation when the denominator on the right hand side of the equations, defining dependent variables, contains decision variables. Note that definition of dependent variables is often required for application problems where the objective function involves numerous, complex terms. All the VLE-LS problems are solved using the latest version (23.6.3/9.0.6) of GAMS/BARON. Maximum CPU time is set at 1 hour.

The comparison of IDE (with NR_{max} and NP = 30) and BARON is performed in order to assess their relative merits for VLE-LS problems. The common comparison of IDE (Table 4.5) and BARON (Table 4.7) based on finding the global optimum and computational time for it, shows that BARON obtained global solution for all ten VLE-LS problems tested. Note that the computational time is on the same computer but IDE and BARON are on different platforms (namely, Excel and GAMS). BARON uses less computational time for LS-2, 3, 7, 8, 9 and 10 compared to IDE. But for LS-1, 4, 5 and 6 BARON has used more computational time. Especially for LS-6, which has 3 variables, BARON has reached the maximum CPU time to solve this problem. Based on the results in Table 4.5 and use of IDE in VBA (which is an order of magnitude slower than software such as MATLAB), it can be concluded that IDE is better than or comparable to BARON for solving VLE-LS problems.

Table 4.7. The Performance and Time Required by BARON for VLE-LS Problems

Problem	BARON (Global Optimum Found)	CPU time (seconds)
		BARON
LS-1	yes	25
LS-2	yes	12
LS-3	yes	4
LS-4	yes	46
LS-5	yes	27
LS-6	yes	3600
LS-7	yes	12
LS-8	yes	13
LS-9	yes	19
LS-10	yes	5

4.3.5. Results and Discussion Using Error-in-Variables Approach

4.3.5.1 Performance of IDE with Different Stopping Criteria

Unlike LS approach, EIV approach considers errors in the measured data in all decision variables, and hence the number of decision variables, *npar* + *c*×*ndat*, increases significantly. The accounting for the error in all variables leads to unbiased estimates of parameter values, and achieves more accurate models. Furthermore, in solving the VLE problems using EIV approach, not only parameter estimation results are obtained but also data reconciliation is performed. Owing to increased number of decision variables, VLE-EIV problems are more challenging than VLE-LS problems. For solving VLE-EIV problems, population size of *NP* = 10D is used which is the same as that by Bonilla-Petriciolet et al.(2010c), and other parameter settings are the same as above (Section 4.3). Table 4.8 presents SR and NFE results using IDE with each of four stopping criteria; SR of EIV problems is better than that of LS problems in Table 4.5 with the same stopping criterion, probably due to larger population size arising from 10D and large number of variables. As expected and as in the LS approach, SR improves with increasing iterations for VLE-EIV problems too.

It is clear from Table 8 that IDE with any of the 4 criteria tested can achieve 100% SR except for problems 3 and 4. NFE shows that SC-1 and SC-2 can terminate the algorithm before reaching the maximum number of iterations except for problems 1 and 2. NR_{max} can terminate the algorithm before reaching the maximum number of iterations except for problem 1, and it uses the least NFE compared to others. Compared to the results with LS problems in Table 4.5, it is obvious that SC_{max} stopping criteria performs differently. For LS problems, SC_{max} terminated the global search very early but for EIV problems, SC_{max} terminated the algorithm very late. This is probably due to the use of objective function improvement in this stopping criterion and characteristics of the objective function. Thus, suitable values for SC_{max} are problem-dependent. The CPU time of IDE with NR_{max} and IDE with G_{max}, reported in Table 4.8, shows that use of NR_{max} significantly reduced the computation time for 5 of the 6 EIV problems tested, compared to IDE with G_{max}, without affecting reliability. Considering both GSR and total NFE for the four stopping criteria (Table 8), it is clear that NR_{max} is an efficient and robust stopping criterion for VLE-EIV problems too.

Table 4.8. NFE and SR of IDE with Four Different Stopping Criteria (SC-1, SC-2, NR_{max} and G_{max}) for Solving VLE-EIV Problems

No.	SC-1 = 6D		SC-2 = 12D		NR_{max} = 20			G_{max} = 1000		
	SR	NFE	SR	NFE	SR	NFE	CPU	SR	NFE	CPU
1	100	340000	100	340000	100	340000	3157	100	340000	3157
2	100	360000	100	360000	100	164524	1132	100	360000	3549
3	96	147727	98	178750	98	113559	771	98	220000	1655
4	97	146300	97	220000	97	120192	795	97	220000	1622
5	100	167760	100	193157	100	118612	915	100	240000	1842
6	100	144569	100	184278	100	106668	728	100	220000	1576
GSR	98.8		99.2		99.2			99.2		
Total NFE		1.31E+06		1.48E+06		9.64E+05			1.60E+06	

4.3.5.2 Comparison of IDE with Other Stochastic Methods

Of the many stochastic algorithms, SA, PSO, DE and DETL have been evaluated for parameter estimation involved in modeling VLE data.(Bonilla-Petriciolet et al., 2010b and 2010c) Fig. 4.3 shows GSR values for solving VLE-EIV problems by these and IDE algorithms using $NT \times NS = NP = 10D$. It can be seen that IDE achieves consistently much higher GSR with SC-1, SC-2 and NR_{max} compared to SA, PSO, DE and DETL. This is a clear indication that IDE is very reliable for solving VLE-EIV problems.

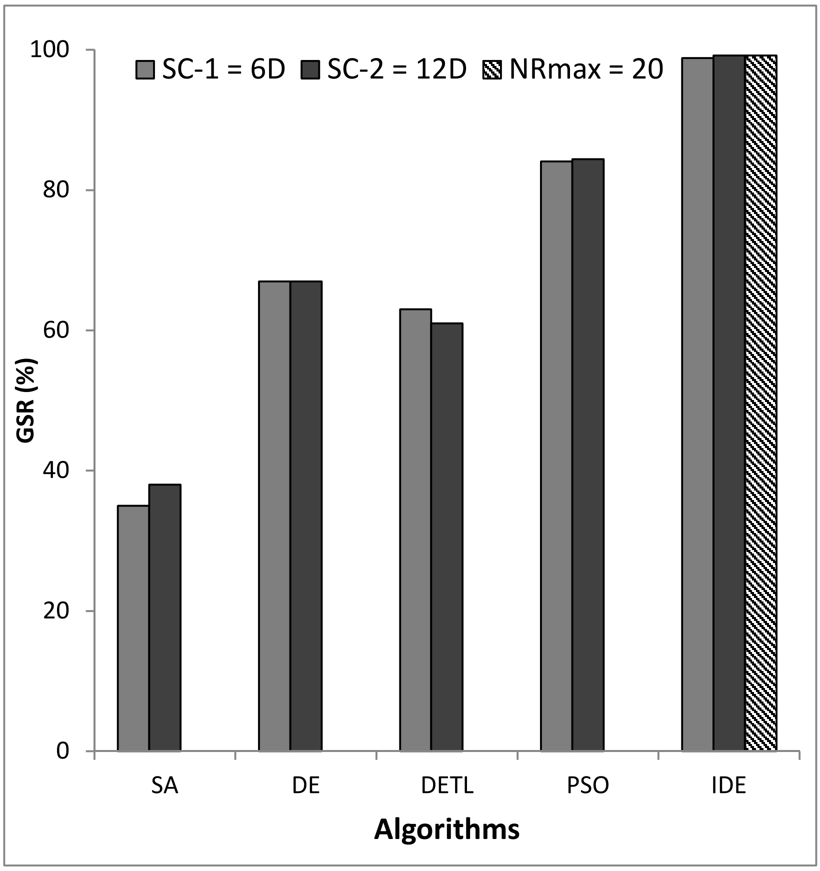


Figure 4.3. Global Success Rate (GSR) for solving VLE-EIV problems by SA, DE, DETL, PSO and IDE algorithms with different stopping criteria.

4.3.5.3 Comparison of IDE with a Deterministic Method

BARON is used to solve VLE-EIV problems also, and the performance results and CPU time required of BARON are summarized in Table 4.9. Comparing the performance of BARON with that of IDE in Table 4.8, reliability of BARON is low within the maximum CPU time setting of one hour; it obtained only the local optimum for EIV-1 and EIV-2, and terminated with the message: "No feasible solution was found and bounds too wide – model status uncertain" for the remaining 4 problems. So, bounds on θ_1 and θ_2 are reduced by a factor of 10, and BARON was tried again for solving all 6 VLE-EIV problems. The global solution can then be obtained; however, CPU time was still the maximum one hour (Table 4.9). In contrast, IDE has shown high reliability and efficiency; it obtained the global optimum with significantly less CPU time for 5 of the 6 EIV problems (Table 4.8). The above message indicated that BARON in GAMS platform is taking explicit equations such as equations 14 and 15 as constraints. So, an attempt was made to substitute these equations into the objective function, which became extremely long and prone to error. In any case, this laborious substitution was tried for EIV-6 problem. The global solution can then be obtained; however, CPU time was still the maximum one hour.

Table 4.9. Performance and Time Required by BARON for VLE-EIV Problems

Problem	Original Bounds (See Table 4)		Reduced Bounds*	
	Global Optimum Found?	CPU time (seconds)	Global Optimum Found?	CPU time (seconds)
EIV-1	Local Optimum	3600	yes	3600
EIV-2	Local Optimum	3600	yes	3600
EIV-3	No Convergence	3600	yes	3600
EIV-4	No Convergence	3600	yes	3600
EIV-5	No Convergence	3600	yes	3600
EIV-6	No Convergence	3600	yes	3600

* In this case, bounds on θ_1 and θ_2 are reduced by a factor of 10 from (-10,000 to 200,000) to (-1,000 to 20,000); bounds for other variables are same as those in Table 4.4.

4.3.5.4 Solution of LS Problems Using EIV Approach

Four LS problems from Table 4.3 (LS-1, LS-2, LS-3 and LS-9) for different systems, are also studied using EIV approach; these are referred to as EIV-7, EIV-8, EIV-9 and EIV-10 respectively. Details of these EIV problems are listed in Table 4.10. These problems are in addition to problems LS-7, LS-8 and LS-10, which have been considered via EIV approach as EIV-5, EIV-2 and EIV-3 (in Table 4.4) respectively. As stated in Section 4.3, different objective functions are used in the two approaches to obtain the optimal parameter values for the respective objective. By comparing the objective function value obtained from both LS and EIV approaches, it is clear that LS approach can achieve a better objective value. This is because EIV approach includes many more error residuals divided by respective variance in the objective function. However, EIV approach is more realistic for data modeling problems.

Table 4.10. Details of VLE-LS Problems Using EIV approach

No.	System	Data	Variables	Global optimum
EIV-7	tert butanol + 1 butanol	P = 100 mmHg *ndat* = 9 ; σ(0.003, 0.0029, 1.7, 0.083)	D = 20 θ_1, $\theta_2 \in$ (-10000, 200000); $x_{ij}^t \in (x_{ij} - 3\sigma, x_{ij} + 3\sigma)$ $T^t \in (T - 3\sigma, T + 3\sigma)$	F_{obj} = 2.677843 θ_1 = -674.78 θ_2 =1234.60
EIV-8	tert butanol + 1 butanol	P = 700 mmHg ;*ndat* = 9 ; σ(0.003, 0.0029, 1.7, 0.083)	D = 20	F_{obj} = 17.658547 θ_1 = -780.45 θ_2 =1695.77
EIV-9	tert butanol + 1 butanol	P = 500 mmHg ; *ndat* = 9; σ(0.003, 0.0029, 1.7, 0.083)	D = 20	F_{obj} = 7.585433 θ_1 = -746.22 θ_2 =1498.13
EIV-10	benzene + hexafluoro-benzene	P = 760 mmHg **;** *ndat* = 29; σ(0.001, 0.01, 0.75, 0.1)	D = 60	F_{obj} = 16.925014 θ_1 = -420.67 θ_2 =1060.34

Note: The models systems used are Wilson and ideal gas. The bounds of variables used for these the EIV problems as given in EIV-7.

The performance of IDE with *NR* criterion for the same problem by both LS and EIV approaches is compared in Table 4.11. These results indicate that IDE with *NR* criterion is somewhat more reliable for EIV approach compared to LS approach. This is interesting since LS approach involves significantly fewer parameters, and is probably due to the smaller population size used in LS problems (namely, 30 compared to 10D in EIV problems) that limits the exploration of the whole search space (note that wide bounds of decision variables indicate larger searching space). On the other hand, NFE required for EIV approach is 5 times more than that for LS approach, mainly due to the larger number of variables in the former approach.

Table 4.11. Performance of IDE with *NR* Criterion for LS and EIV Approaches

Problem	SR	NFE	Problem	SR	NFE
LS-1	85	24921	EIV-7	93	104189
LS-2	88	29680	EIV-8	96	159112
LS-3	74	26729	EIV-9	95	172600
LS-7	97	24896	EIV-5	100	118612
LS-8	89	24003	EIV-2	100	164524
LS-9	100	28007	EIV-10	100	241782
LS-10	91	31497	EIV-3	98	113559

4.4 Conclusions

The IDE algorithm developed by our group is applied to solve the parameter estimation problems in chemical engineering. The performance of IDE indicates that IDE is more efficient and reliable than DETL for parameter estimation problems in dynamic models. Subsequently, IDE is applied to parameter estimation in VLE modeling problems based on LS and EIV approaches. Compared to SA, PSO, DE and DETL, IDE was found to be the overall best for these modeling problems. In addition, among the stopping criteria tested, the stopping criterion based on NR_{max} can switch the global search to local search at the right time in order to achieve good reliability and to reduce computational resources. The comparison of results shows that IDE is comparable to or better than BARON for VLE modeling problems tested, particularly for VLE-EIV problems with more than 20 decision variables and very wide bounds. In summary, IDE algorithm with NR_{max} stopping criterion is robust, reliable, easy to use and suitable for solving application problems.

Chapter 5

Evaluation of Integrated Differential Evolution for Phase Equilibrium and Stability Problems*

5.1. Introduction

Phase equilibrium calculations (PEC) and phase stability (PS) problems are crucial for the analysis of chemical process. Novel processes handle complex mixtures, severe operating conditions, or even incorporate combined unit operations (e.g. reactive distillation, extractive distillation etc.). The reliable computation of the thermodynamic state for these systems is especially important due to the direct impact of wrong estimations on energy consumption and operating costs. When a mixture is analyzed, PEC involves not only the calculation of number of moles of each phase but also the number of stable phases where PS is used to determine the stability of the calculated composition at equilibrium. In general, number and type of phases, at which Gibbs free energy function achieves the global minimum, are unknown in PEC problems, and so several calculations may have to be performed using different phase configurations to identify the stable equilibrium state. Therefore, the unknown phases of general PEC increase the complexity of the optimization problem. Both PEC and PS problems require the global optimization of a specific function; usually, these have to be solved numerous times during a simulation. Specifically, PS analysis requires the minimization of tangent plane distance function (TPDF), while the Gibbs free

* This Chapter is based on the paper - Zhang, H., Fernandez-Vargas, J.A., Rangaiah, G.P., Bonilla-Petriciolet, A. and Segovia-Hernandez, J.G., Evaluation of Integrated Differential Evolution and Unified Bare-bones Particle Swarm Optimization for Phase Equilibrium and Stability Problems. *Fluid Phase Equilibrium*, 310, 129-141, 2011.

energy function needs to be minimized for PEC (Srinivas and Rangaiah, 2007a). A reactive phase equilibrium calculation (rPEC) or chemical equilibrium, is performed if any reaction is possible in the system under study, and the objective function must satisfy the chemical equilibrium constraints.

In general, there are several challenges in finding the global optimum of Gibbs free energy function. First, number and type of phases where the thermodynamic function achieves the global optimum are usually unknown *a priori*. Second, high non-linearity of thermodynamic models, non-convexity of Gibbs free energy function and the presence of a trivial solution in the search space make PEC and PS problems difficult to solve. Moreover, for a fixed number of phases and components, Gibbs free energy function may have local optima that are very comparable to the global optimum value, which makes it challenging to find the global optimum (Bonilla-Petriciolet et al., 2011). Thus, PEC, rPEC and PS problems require a reliable, efficient and robust global optimization algorithm. Further, application of global optimization techniques to these problems is very challenging.

Many deterministic and stochastic optimization algorithms have been proposed and tested for finding the global optimum in PEC, rPEC and PS problems, particularly in the past two decades (e.g., Reynolds et al., 1997; Wasylkiewicz and Ung, 2000; Rangaiah, 2001; Burgos-Solorzano et al., 2004; Srinivas and Rangaiah, 2007a; Rossi et al., 2009; Bonilla-Petriciolet et al., 2010a; Bonilla-Petriciolet et al., 2011). Deterministic global optimization studies have been applied to different PEC, PS and/or rPEC problems. Homotopy continuation methods have been applied to PEC and PS problems (Sun and Seader, 1995; Jalali and Seader, 1999; Jalali et al., 2008). Although homotopy-continuation algorithm guarantees global convergence to a single solution, it does not guarantee global convergence to multiple solutions. Even using complex search spaces, the success of continuation methods in finding all solutions cannot be assured. Burgos-Solorzano et al. (2004) applied interval Newton method for solving the PEC problems under high pressure. This method can solve nonlinear equations to find all solutions lying within the variable bounds. It requires an interval extension of the Jacobian matrix, and involves setting up and solving the interval Newton equation for a new interval. However, it is very hard to find all solutions and Jacobian matrix for the complex systems, and the computational time is significant for multi-component systems.

Recently, Rossi et al. (2011) applied convex analysis method to PEC and rPEC problems. This method employs the CONOPT solver in GAMS (General Algebraic Modeling System). The proposed method can solve PEC problems with high efficiency and reliability but it requires the convexity of the model. Branch and bound methods have been applied to many applications including PS and PEC problems (Harding and Floudas, 2000; Cheung et al., 2002). In general, these methods are often slow and require a significant numerical effort that grows exponentially with problem size (Nichita et al., 2002; Wakeham and Stateva, 2004). In addition, branch and bound methods require certain properties of the objective function, and problem reformulation is usually needed to guarantee the global convergence. The problem reformulation can be very difficult to perform, especially for complex thermodynamic models such as equations of state with non-traditional mixing rules. Finally, Nichita and co-workers applied the tunneling method to perform stability analysis of various systems (Nichita et al., 2002; Nichita et al., 2008) and to PEC problems (Nichita et al., 2002; Nichita et al., 2004). Their results suggest that tunneling method is a robust and efficient tool for these applications. However, it requires feasible and improved initial estimates for reliability and computational efficiency respectively (Nichita et al., 2002). For an unknown system, it is very difficult to provide a feasible and good initial estimate for the algorithm.

In summary, the deterministic methods can guarantee convergence to the global optimum but they usually require certain properties such as continuity, *a priori* information of the system; reformulation of the problem may be needed depending on the characteristic of the thermodynamic models, and the computational time grows exponentially with problem size. In contrast, stochastic methods are quite simple to implement and use. They do not require any assumptions or transformation of the original problems, can be applied with any thermodynamic model, and yet provide a high probabilistic convergence to the global optimum. They can often locate the global optimum in modest computational time compared to deterministic methods (Bonilla-Petriciolet et al., 2011).

In recent years, several stochastic global optimization techniques have been applied to solve the PS and PEC problems in non-reactive and reactive systems (Reynolds et al., 1997; Rangaiah, 2001; Nichita et al., 2002; Nichita et al., 2004; Bonilla-Petriciolet et al., 2006; Srinivas and Rangaiah, 2006; Srinivas and Rangaiah, 2007a; Srinivas and Rangaiah, 2007b; Nichita et al., 2008; Bonilla-Petriciolet and Segovia-Hernandez, 2010; Bonilla-Petriciolet et al., 2011). These algorithms include SA, GA, TS, DE, RT and PSO. In particular, Srinivas and Rangaiah (2007a) studied DE and TS for non-reactive mixtures, and proposed two versions of DETL, in order to improve the performance of the optimization algorithm. Srinivas and Rangaiah (2006) evaluated the RT on a number of medium-sized problems including vapor-liquid, liquid-liquid and vapor-liquid-liquid equilibrium problems. RT can locate the global optimum for most of the examples tested but its reliability is low for problems having a local minimum comparable to the global minimum. In a recent study, Bonilla-

Petriciolet and Segovia-Hernandez (2010) tested different versions of PSO for PS and PEC for both reactive and non-reactive systems, and their results show that classical PSO is a reliable method with good performance.

Systematic and comprehensive comparison of different global optimization methods is challenging. However, some comparison of stochastic with deterministic algorithms for phase equilibrium calculations can be found in the literature. Teh and Rangaiah (2002 and 2003) compared GA and TS with several deterministic algorithms such as Rachford-Rice-Mean value theorem-Wegstein's projection method, accelerated successive substitution method, Nelson's method, simultaneous equation-solving method, linearly constrained minimization method, GLOPEQ and enhanced interval analysis method for solving phase equilibrium calculations. Their comparison shows that some stochastic methods can be more efficient than deterministic algorithms.

Most of the stochastic methods have some parameters to be tuned for different problems in order to improve the convergence to the global optimum. Selection of proper parameter values for different problems usually cost a lot of effort, and an improper choice can result in computational inefficiency or poor reliability. In order to overcome such difficulties, this work evaluates two global optimization algorithms (IDE and IDE without tabu list and radius, IDE_N) that have fewer algorithm parameters, for PEC, rPEC and PS problems involving multiple components, multiple phases and popular thermodynamic models. The performance of IDE and IDE_N on PEC, rPEC and PS problems are compared with recent global optimization algorithms and discussed based on both reliability and computational efficiency using practical stopping criteria.

The rest of this chapter is organized as follows. Description of PEC, PS and rPEC problems is given in Section 5.2. Implementation of IDE_N and IDE is presented in Section 5.3. Section 5.4 presents the results and discusses the performance of IDE_N and IDE on PEC, PS and rPEC problems. Finally, conclusions of this work are summarized in Section 5.5.

5.2. Description of PEC, PS and rPEC Problems

A brief description of the global optimization problems including the objective function, decision variables and constraints, for PEC, PS and rPEC problems is given in the following sections.

5.2.1 Description of PEC Problems

A mixture of substances at a given temperature, *T,* pressure, *P* and total molar amount may separate into two or more phases. The composition of the different substances is the same throughout a phase but may significantly vary in different phases at equilibrium. If there is no reaction between the different substances, then it is a phase equilibrium problem. There are mainly two different approaches for PEC: equation solving approach and Gibbs free energy minimization approach. The former involves solving a set of non-linear equations arising from mass balances and equilibrium relationships. The latter involves the minimization of the Gibbs free energy function. Although the first approach seems to be faster and simple, the solution obtained may not correspond to the global minimum of Gibbs free energy function. Moreover, it needs a priori knowledge of phases existing at equilibrium (Rangaiah, 2001). Classic thermodynamics indicate that minimization of Gibbs free energy is a natural approach for calculating the equilibrium state of a mixture. Hence, this study uses Gibbs free energy minimization for PEC, which was used to determine phase compositions at equilibrium in several works (e.g., McDonald and Floudas, 1995; Reynolds al., 1997; Rangaiah, 2001; Teh and Rangaiah, 2003).

The mathematical formulation involves the minimization of a non-convex objective function (Gibbs free energy) subject to mass balance equality constraints and bounds that limit the range of variables. In a non-reactive system with c components and π phases, the objective function for PEC is

$$g = \sum_{j=1}^{\pi} \sum_{i=1}^{c} n_{ij} \ln(x_{ij}\gamma_{ij}) = \sum_{j=1}^{\pi} \sum_{i=1}^{c} n_{ij} \ln\left(\frac{x_{ij}\hat{\varphi}_{ij}}{\varphi_i}\right) \tag{5.1}$$

where n_{ij}, x_{ij}, γ_{ij} and $\hat{\varphi}_{ij}$ are respectively the moles, mole fraction, activity coefficient and fugacity coefficient of component i in phase j, and φ_i is the fugacity coefficient of pure component. Eq. (5.1) must be minimized with respect to n_{ij} taking into account the following mass balance constraints:

$$\sum_{j=1}^{\pi} n_{ij} = z_i n_F \quad i = 1,...,c \tag{5.2}$$

$$0 \le n_{ij} \le z_i n_F \quad i = 1,...,c \quad j = 1,...,\pi \tag{5.3}$$

where z_i is the mole fraction of component i in the feed and n_F is the total moles in the feed.

To perform unconstrained minimization of Gibbs energy function, we can use new variables instead of n_{ij} as decision variables. The introduction of the new variables eliminates the restrictions imposed by material balances, reduces problem dimensionality and the optimization problem is transformed into an unconstrained one. For multi-phase non-reactive systems, new variables $\beta_{ij} \in (0, 1)$ are defined and employed as decision variables by using the following expressions:

$$n_{i1} = \beta_{i1} z_i n_F \quad i = 1,..., c \tag{5.4}$$

$$n_{ij} = \beta_{ij}\left(z_i n_F - \sum_{m=1}^{j-1} n_{im} \right) \quad i = 1,...,c; \quad j = 2,...,\pi - 1 \tag{5.5}$$

$$n_{i\pi} = z_i n_F - \sum_{m=1}^{\pi-1} n_{im} \quad i = 1,...,c \tag{5.6}$$

Using this formulation, all trial compositions satisfy the mass balances allowing the easy application of optimization strategies. For Gibbs energy minimization, the number of phases existing at the equilibrium is assumed to be known *a priori*, and the number of decision variables β_{ij} is c (π - 1) for non-reactive systems.

Details of PEC problems used in this study are in Table 5.1. In most of the reported studies, PEC problems tested are assuming that the number and type of phases are known; such problems are also known as phase split calculations. In this study too, the same assumption is made, and the problems tested are simply referred to as PEC problems.

5.2.2 Description of PS Problems

Phase stability (PS) problem is used to determine the thermodynamic state that corresponds to the global minimum of Gibbs free energy. Its results can be used to find good starting points to improve the reliability of PEC. PS is often tested using the tangent plane criterion, which states that a phase is stable provided that the tangent plane generated at the corresponding composition lies below the molar Gibbs energy surface for all compositions (Sun and Seider, 1995; Harding and Floudas, 2000). As an alternative, Mitsos and Barton (2007) reinterpreted the Gibbs tangent plane stability criterion via a Lagrangian duality approach, as the solution of the dual problem of a primal problem that minimizes Gibbs free energy subject to material balances for solving the PS problems.

One common implementation of the tangent plane criterion (Sun and Seider, 1995; Harding and Floudas, 2000) is to minimize the tangent plane distance function (TPDF), defined as the vertical distance between the molar Gibbs energy surface and the tangent plane at the given phase composition. Specifically, TPDF is given by

$$TPDF = \sum_{i=1}^{c} y_i (\mu_i|_y - \mu_i|_z) \tag{5.7}$$

where $\mu_i|_y$ and $\mu_i|_z$ are the chemical potentials of component i calculated at compositions y and z, respectively. Eq. (5.7) is the objective function, and the constraint and bounds are

$$\sum_{i=1}^{c} y_i = 1 \text{ and } 0 \leq y_i \leq 1 \tag{5.8}$$

For stability analysis of a phase/mixture of composition z, *TPDF* must be globally minimized with respect to composition of a trial phase y. If the global minimum value of TPDF is zero, then the specified phase and others sharing the same tangent plane would coexist at equilibrium. The decision variables in phase stability problems are y_i for i = 1, 2,…, c.

The constrained global optimization of *TPDF* can be transformed into an unconstrained problem by using decision variables β_i instead of y_i as follows:

$$n_{iy} = \beta_i z_i n_F \quad i = 1,...,c \tag{5.9}$$

and

$$y_i = n_{iy} \Big/ \sum_{j=1}^{c} n_{iy} \quad i = 1,...,c \tag{5.10}$$

where n_F is the total moles in the feed mixture used for stability analysis, and n_{iy} are the conventional mole numbers of component i in trial phase y. The number of decision variables is still *c* for the unconstrained minimization of TPDF. Thus, the unconstrained global optimization problem for phase stability analysis is:

$$\begin{aligned} &\min TPDF(\beta) \\ &0 \le \beta_i \le 1 \ i=1,...,c \end{aligned} \tag{5.11}$$

The calculation of *TPDF* is straightforward with almost any thermodynamic model because:

$$\frac{\mu_i - \mu_i^0}{R_g T} = \ln\left(\frac{x_i \hat{\varphi}_i}{\varphi_i} \right) = \ln(x_i \gamma_i) \tag{5.12}$$

where R_g is the universal gas constant, μ_i is the chemical potential of component *i* at the mixture, and μ_i^0 is the chemical potential of pure component *i*. More details of PS problem formulation can be found in Rangaiah (2001). Characteristics of PS problems used in this study are summarized in Table 5.1

Table 5.1. Details of PEC and PS Problems Studied

PEC&PS				Global optimum for	
No.	System	Feed conditions	Thermodynamic models	Equilibrium	Stability
1	n-Butyl acetate + water	n_F = (0.5, 0.5) at 298K and 101.325kPa	NRTL model and parameters reported by Rangaiah (2001).	-0.020198	-0.032466
2	Toluene + water + aniline	n_F = (0.29989, 0.20006, 0.50005) at 298K and 101.325kPa	NRTL model and Model parameters reported by McDonald and Floudas (1995).	-0.352957	-0.294540
3	N_2+C_1 + C_2	n_F = (0.3, 0.1, 0.6) at 270K and 7600kPa	SRK EoS with classical mixing rules. Model parameters reported by Bonilla-Petriciolet et al. (2006).	-0.547791	-0.015767
4	C_1 + H_2S	n_F = (0.9813, 0.0187) at 190K and 4053kPa	SRK EoS with classical mixing rules. Model parameters reported by Rangaiah (2001).	-0.019892	-0.003932
5	C_2 + C_3 + C_4 + C_5 + C_6	n_F = (0.401, 0.293, 0.199, 0.0707, 0.0363) at 390K and 5583kPa	SRK EoS with classical mixing rules. Model parameters reported by Bonilla-Petriciolet et al. (2006).	-1.183653	-0.000002
6	C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + $C_{7\text{-}16}$ + C_{17+}	n_F = (0.7212, 0.09205, 0.04455, 0.03123, 0.01273, 0.01361, 0.07215, 0.01248) at 353K and 38500kPa	SRK EoS with classical mixing rules. Model parameters reported by Harding and Floudas (2000).	-0.838783	-0.002688
7	C_1 + C_2 + C_3 + iC_4 + C_4 + iC_5 + C_5 + C_6 + iC_{15}	n_F = (0.614, 0.10259, 0.04985, 0.008989, 0.02116, 0.00722, 0.01187, 0.01435, 0.16998) at 314K and 2010.288kPa	SRK EoS with classical mixing rules. Model parameters reported by Rangaiah (2001).	-0.769772	-1.486205
8	C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7 + C_8 + C_9 + C_{10}	n_F = (0.6436, 0.0752, 0.0474, 0.0412, 0.0297, 0.0138, 0.0303, 0.0371, 0.0415, 0.0402) at 435.35K and 19150kPa	SRK EoS with classical mixing rules. Model parameters reported by Bonilla-Petriciolet et al. (2006).	-1.121176	-0.0000205

Table 5.2. Details of rPEC (Chemical Equilibrium) Problems Studied

rPEC No.	System	Feed conditions	Thermodynamic models	Global optimum
1	A1+A2 ↔ A3+A4 (1) Ethanol (2) Acetic acid (3) Ethyl acetate (4) Water	n_F = (0.5, 0.5, 0.0, 0.0) at 355K and 101.325kPa	NRTL model and ideal gas. K_{eq}=18.670951	-1.298000
2	A1+A2 ↔ A3, and A4 as an inert component (1) Isobutene (2) Methanol (3) Methyl ter-butyl ether (4) n-Butane	n_F = (0.3, 0.3, 0.0, 0.4) at 373.15K and 101.325kPa	Wilson model and ideal gas. $\Delta G^0_{rxs}/R = -4205.05 +10.0982T-0.2667T\ln T$ $\ln K_{eq} = -\Delta G^0_{rxs}/R$ where T is in K	-1.434267
3	A1+A2 +2A3↔ 2A4 (1) 2-Methyl-1-butene (2) 2-Methyl-2-butene (3) Methanol (4) Tert-amyl methyl ether	n_F = (0.354, 0.183, 0.463, 0.0) at 355K and 151.95kPa	Wilson model and ideal gas. $K_{eq}=1.057*10^{-04} e^{4273.5/T}$ where is in K	-1.226367
4	A1+A2 ↔ A3+A4 (1) Acetic acid (2) n-Butanol (3) Water (4) n-Butyl acetate	n_F = (0.3, 0.4, 0.3, 0.0) at 298.15K and 101.325kPa	UNIQUAC model and ideal gas. $\ln K_{eq}=450/T +0.8$	-0.301730
5	A1+A2 ↔ A3	n_F = (0.6, 0.4, 0.0	Margules solution model. $g^E/R_gT = 3.6x_1x_2+2.4x_1x_3+2.3x_2x_3$ K_{eq}=0.9825	-1.798377
6	A1+A2+2A3 ↔ 2A4 with A5 as inert component (1) 2-Methyl-1-butene (2) 2-Methyl-2-butene (3) Methanol (4) Tert-amyl methyl ether (5) n-Pen ane	n_F = (0.1, 0.15, 0.7, 0.0, 0.05) at 335K and 151.9875kPa	Wilson model and ideal gas. $K_{eq}=1.057*10^{-04} e^{4273.5/T}$ where T is in K	-0.144508
7	A1+A2 ↔ A3	n_F = (0.52, 0.48, 0.0) at 323.15K and 101.325kPa	Margules solution model. K_{eq} = 3.5	-1.043199
8	A1+A2 ↔ A3+A4	n_F = (0.048, 0.5, 0.452, 0.0) at 360K and 101.325kPa	NRTL model K_{eq}=4.0	-1.347857

5.2.3 Description of rPEC or (Chemical Equilibrium) Problems

In rPEC problems, also known as chemical equilibrium problems, reactions increase the complexity and dimensionality of phase equilibrium problems, and so phase split calculations in reactive systems are more challenging due to non-linear interactions among phases and reactions. The phase distribution and composition at equilibrium of a reactive mixture are determined by the global minimization of Gibbs free energy subject to element/mass balances and chemical equilibrium constraints (Seider and Widagdo, 1996; Burgos-Solorzano et al., 2004). Therefore, to determine the phase equilibrium compositions in reactive systems, it is necessary to find the global minimum of the free energy with respect to mole numbers of components in each of the phases subject to constraints and bounds. The expressions for Gibbs free energy and its mathematical properties depend on the structure of the thermodynamic equation(s) chosen to model each of the phases that may exist at equilibrium (Bonilla-Petriciolet et al., 2011).

Recently, Bonilla-Petriciolet et al. (2011) concluded that the constrained Gibbs free energy minimization approach has the advantage of requiring smaller computing time compared to the unconstrained approach, is straightforward and suitable for chemical equilibrium calculations. In summary, for a system with c components and π phases subject to r independent chemical reactions, the constrained objective function for rPEC is

$$F_{obj} = g - \sum_{j=1}^{\pi} ln\, K_{eq} N^{-1} n_{ref,j} \tag{5.13}$$

where g is given by Eq. (5.1), lnK_{eq} is a row vector of logarithms of chemical equilibrium constants for r independent reactions, $\boldsymbol{N}$ is an invertible, square matrix formed from the stoichiometric coefficients of a set of reference components chosen from r reactions, and $\boldsymbol{n}_{ref}$ is a column vector of moles of each of the reference components. This objective function is defined using reaction equilibrium constants, and it must be globally minimized subject to the following mass balance restrictions (Rangaiah, 2001):

$$\sum_{j=1}^{\pi}\left(n_{ij}-\boldsymbol{v}_i\boldsymbol{N}^{-1}\boldsymbol{n}_{ref,j}\right)=n_{iF}-\boldsymbol{v}_i\boldsymbol{N}^{-1}\boldsymbol{n}_{ref,F} \quad i = 1, \ldots, c-r \tag{5.14}$$

where $n_{i,F}$ is the initial moles of component i in the feed. These mass balance equations can be rearranged to reduce the number of decision variables of the optimization problem and to eliminate equality constraints, which are usually challenging for stochastic optimization methods. Thus, Eq. (5.14) is rearranged to reduce the number of decision variables using the following expression:

$$n_{i\pi}=n_{iF}-\boldsymbol{v}_i\boldsymbol{N}^{-1}(\boldsymbol{n}_{ref,F}-\boldsymbol{n}_{ref,\pi})-\sum_{j=1}^{\pi-1}(n_{ij}-\boldsymbol{v}_i\boldsymbol{N}^{-1}\boldsymbol{n}_{ref,j}) \quad i=1,\ldots,c-r \tag{5.15}$$

Using Eq. (5.15), the decision variables for rPEC are $c\ (\pi-1)+r$ mole numbers (n_{ij}). Then, the global optimization problem can be solved by minimizing Eq. (5.13) with respect to $c\ (\pi-1)+r$ decision variables n_{ij} and the remaining $c-r$ mole numbers ($n_{i\pi}$) are determined from Eq. (5.15), subject to the inequality constraints $n_{i\pi}> 0$.

For rPEC, feasible points satisfy all the mass balance constraints, Eq. (5.14), while infeasible points violate at least one of them (i.e., $n_{i\pi}< 0$ where $i = 1, \ldots, c-r$). The penalty function method is used to solve the constrained Gibbs free energy minimization in reactive systems because it is easy to implement and is considered efficient for handling constraints in the stochastic methods (Rangaiah, 2001). For handling these constraints, absolute value of constraint violation is multiplied with a high penalty weight and then added to the objective function. In case of more than one constraint violation, all constraint violations are first multiplied with the penalty weight, and all of them are added to the objective function. Specifically, the penalty function is given by

$$F_r = \begin{cases} F_{obj} & \text{if } \forall n_{ij} > 0 \quad i = 1,...,c \quad j = 1,...,\pi, \\ F_{obj} + p & \text{otherwise,} \end{cases} \tag{5.16}$$

where p is the penalty term whose value is positive. So, the penalty term used for handling infeasible solutions in rPEC is given

$$p = 10 \cdot \sum_{i=1}^{n_{unf}} \left| n_{i\pi} \right| \tag{5.17}$$

where $n_{i\pi}$ is obtained from Eq. (5.15) and n_{unf} is the number of infeasible mole numbers (i.e., $n_{i\pi} < 0$ where i = 1, ..., $c - r$). In this study, the resulting constrained Gibbs free energy minimization for a reactive system is solved using IDE_N and IDE algorithms. The details of rPEC problems are shown in Table 5.2.

5.3 Implementation of the Methods

In this study, all the optimization algorithms and thermodynamic models are coded in MATLAB. The parameters used for the algorithms are fixed for all problems tested in order to compare the robustness of the algorithm. Further, NP = 10D for both IDE and IDE_N, and TL = 50 and TR = 0.001D are used in IDE. Altogether, there are 24 problems consisting of 8 PEC, 8 PS and 8 rPEC problems, whose details can be found in Tables 5.1 and 5.2. All these problems are multimodal with number of decision variables ranging from 2 to 36. Each problem is solved 100 times independently with a different random number seed for robust performance analysis. The performances of stochastic algorithms are compared based on success rate (SR) and average number of function evaluations (for both global and local searches) in the 100 runs (NFE), for two stopping criteria: SC-1 based on the maximum number of iterations and SC-2 based on the maximum number of iterations without improvement in the *gbest* objective function value (SC_{max}).

Note that NFE is a good indicator of computational efficiency since function evaluation involves extensive computations in application problems. Further, it is independent of the computer and software platform used, and so it is useful for comparison by researchers. SR is the number of times the algorithm located the global optimum to the specified accuracy, out of 100 runs. A run/trial is considered successful if the *gbest* objective function value obtained after the local optimization is within 1.0E-5 from the known global optimum. Also, global success rate (GSR) of different algorithms is reported for all the problems.

At the end of each run by each stochastic algorithm, a local optimizer is used to continue the search to find the global optimum precisely and efficiently. This is also done at the end of different iteration levels for analysis; however, global search in the subsequent iterations is not affected by this. Since all algorithms are implemented in MATLAB, sequential quadratic program (SQP) is chosen as the local optimizer. The best solution at the end of the stochastic algorithm is used as the initial guess for SQP, which is likely to locate the global optimum if the initial guess is in the global optimum region.

5.4. Results and Discussion

5.4.1 Performance of Algorithms on PEC problems

First, GSR values for all PEC problems by IDE and IDE_N with NP of 10D using SC-1 are illustrated in Fig. 5.1, and compared with those of UBBPSO from Zhang et al. (2011c). The results are collected at different iteration levels, starting from 50 to 1500 iteration level, after local optimization at each of these iteration levels. As expected, GSR improves with increasing number of iterations (Fig. 5.1), particularly at lower iteration levels. After 250 iterations, GSR does not improve significantly; this suggests that subsequent iterations without improvement in the results are waste of computational resources. For example, GSR of UBBPSO is 83.5% at 50 iterations; it increases to 88% at 250 iterations and 89.6% at 1500 iterations. GSR of IDE is 83%, 99.9% and 100% at 50, 250 and 1500 iterations respectively. Results in Fig. 5.1 show that IDE has higher reliability and faster convergence rate compared to IDE_N and UBBPSO, for PEC problems. Further, IDE and IDE_N can achieve 100% GSR at 1500 iteration level. Thus, it is essential for the optimization algorithm to stop at the right time incurring least computational resources without compromising reliability of finding the global optimum.

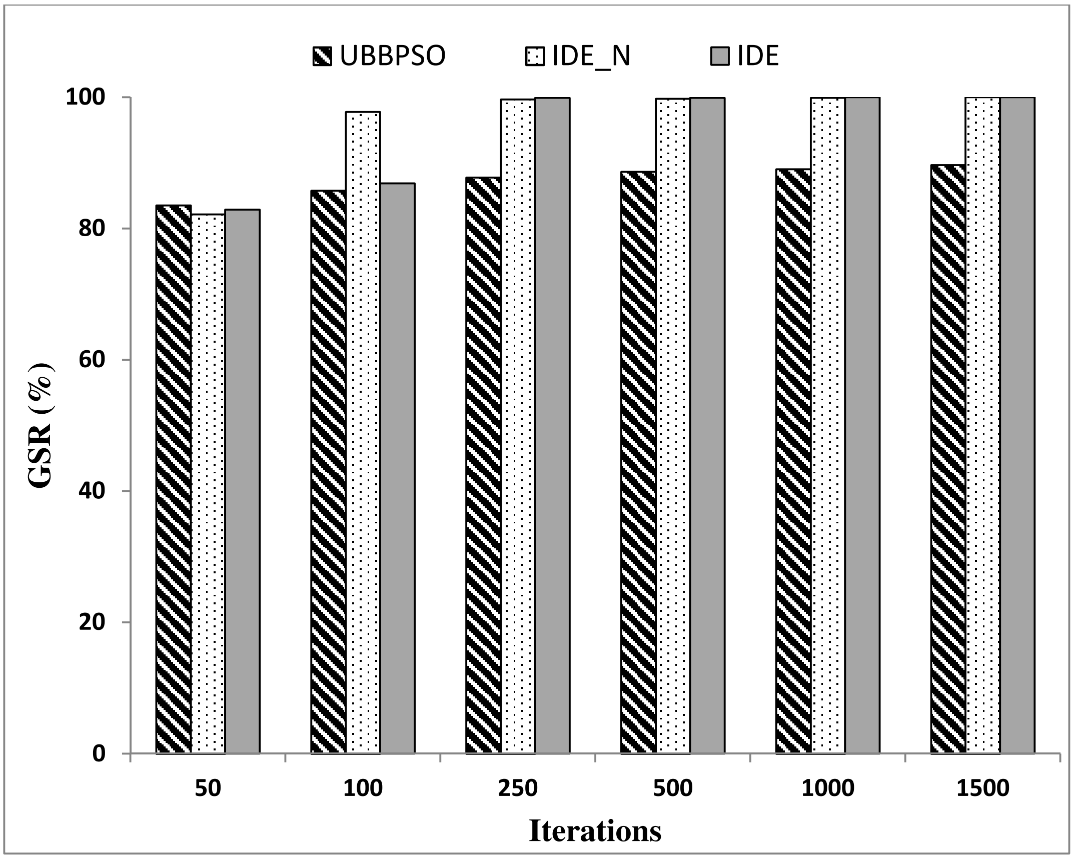


Figure 5.1. Global Success Rate (GSR) Versus Iterations for PEC Problems Using UBBPSO, IDE_N and IDE with SC-1

Table 5.3. Success Rate (SR) and Number of Function Evaluations (NFE) of UBBPSO, IDE_N and IDE for PEC Problems Using SCmax with NP of 10D

PEC		UBBPSO		IDE_N		IDE	
No.	SC_{max}	SR	NFE	SR	NFE	SR	NFE
PEC-1	10	71	1303	72	955	75	886
	25	89	4151	86	1980	93	1826
	50	100	6757	92	4405	98	3407
PEC-2	10	99	1694	100	1326	100	1307
	25	100	7898	100	4803	100	3386
	50	100	10814	100	7988	100	6021
PEC-3	10	100	1816	100	1253	100	954
	25	100	12019	100	5304	100	2916
	50	100	18224	100	7396	100	4488
PEC-4	10	67	1414	85	507	83	491
	25	88	3491	95	1922	98	1371
	50	94	10121	98	4680	100	2669
PEC-5	10	8	9851	5	4101	31	1879
	25	9	16619	54	12406	100	6991
	50	10	25062	97	14599	100	13116
PEC-6	10	99	60812	99	9441	98	6318
	25	100	73893	100	26921	100	17995
	50	100	84375	100	31615	100	29251
PEC-7	10	100	22678	99	15044	100	13699
	25	100	32835	99	30052	100	35353
	50	100	37831	100	47858	100	54626
PEC-8	10	98	116054	96	17456	80	7284
	25	99	125860	100	34345	94	19463
	50	99	138266	100	38509	100	35456

Table 5.3 summarizes SR and NFE obtained by IDE, IDE_N and UBBPSO with SC_{max} = 10, 25 and 50 along with the maximum allowable iterations of 1500 (to avoid indefinite looping), all using NP = 10D. As expected, reliability of the algorithms increases with SC_{max}, which requires more NFE, because probability to locate the global optimum region increases as the algorithms are allowed to run for more iterations. For PEC problems 1-3 and 6-8, the three algorithms obtained similar

high reliability; and, for PEC problems 4 and 5, IDE and IDE_N obtained better reliability than UBBPSO. Even through the same stopping criterion is used for all the algorithms; NFE required by IDE is much less than that of IDE_N and UBBPSO. As shown in Table 5.3, the total NFE required by IDE for all tests on PEC problems is 271153 compared to 324866 and 823838 for IDE_N and UBBPSO respectively.

Fig. 5.2 summarizes GSR and NFE of IDE, IDE_N and UBBPSO algorithms with four stopping criteria. We obtain the same conclusion of higher reliability and increased NFE with higher SC_{max}. It can be observed in Fig. 5.2a that the use of SC-2 gives lower GSR compared to SC-1. However, with the use of SC_{max} = 50, the reliability of the algorithm is only slightly lower than that with SC-1 (1500 iterations). Comparison of the three algorithms shows that IDE uses least NFE to terminate the global search by SC-2. In general, SC-2 requires significantly fewer NFE compared to SC-1. Especially, with SC_{max} = 50, SR obtained by the algorithms is comparable to that obtained with SC-1 but uses much fewer NFE (Fig. 5.2). Compared to IDE_N, IDE has achieved better reliability with fewer NFE, probably due to the tabu list and tabu check operations in IDE, which prevents revisiting the searched areas thus enhancing the global search ability. In summary, it can be concluded that IDE is better and that SC_{max} = 50 is a good stopping criterion for PEC problems in order to achieve both high reliability and efficiency.

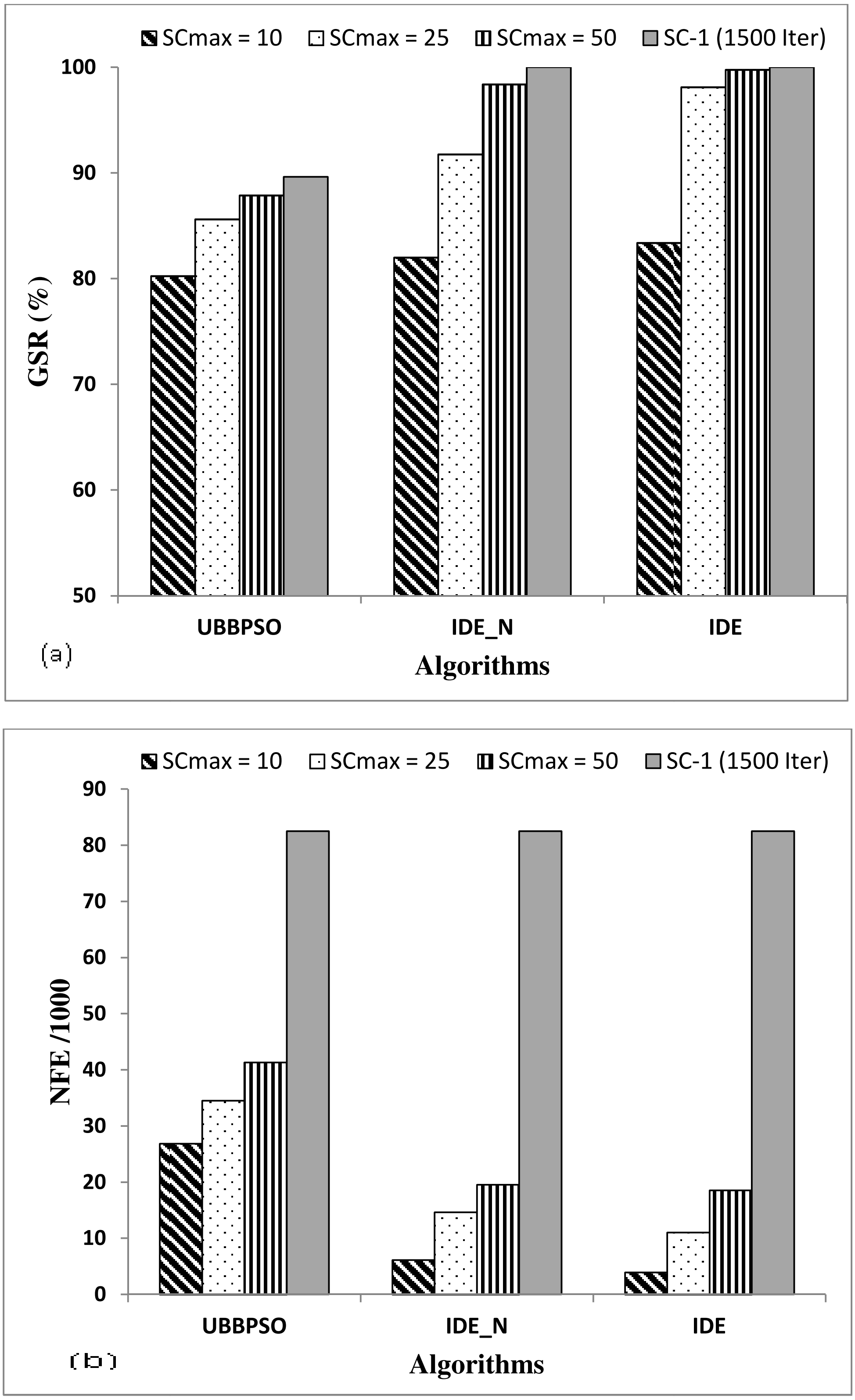


Figure 5.2. Global Success Rate, GSR (plot a) and Average NFE (plot b) of UBBPSO, IDE_N and IDE for PEC Problems Using SC-2 (SC_{max} = 10, SC_{max} = 25 and SC_{max} = 50) and SC-1 (1500 Iterations)

5.4.2 Performance of Algorithms on PS problems

On PS problems, similar tests using the three stochastic algorithms are performed. As expected, GSR of UBBPSO, IDE_N and IDE for all PS problems using SC-1 improves with increasing number of iterations (Fig. 5.3). The highest GSR is 90.1% obtained by UBBPSO. Results in Fig. 5.3 show that the selected PS problems are more difficult to optimize compared to PEC problems. At 50 iterations, IDE_N obtained best GSR, but from 100 to 1500 iterations, GSR of IDE_N did not improve but UBBPSO and IDE obtained better GSR. In fact, IDE_N has faster convergence but it is easily trapped at a local optimum. GSR of UBBPSO is 73% at 50 iterations; it increases to 84% at 250 iterations and 90% at 1500 iterations. On the other hand, GSR of IDE is 73%, 87% and 88% at 50, 250 and 1500 iterations, respectively. This performance indicates that reliability of the algorithms did not improve significantly after 250 iterations. Thus, it is necessary to use a suitable stopping criterion for the optimization algorithm to stop at the right time incurring least computational resources without compromising reliability of finding the global optimum.

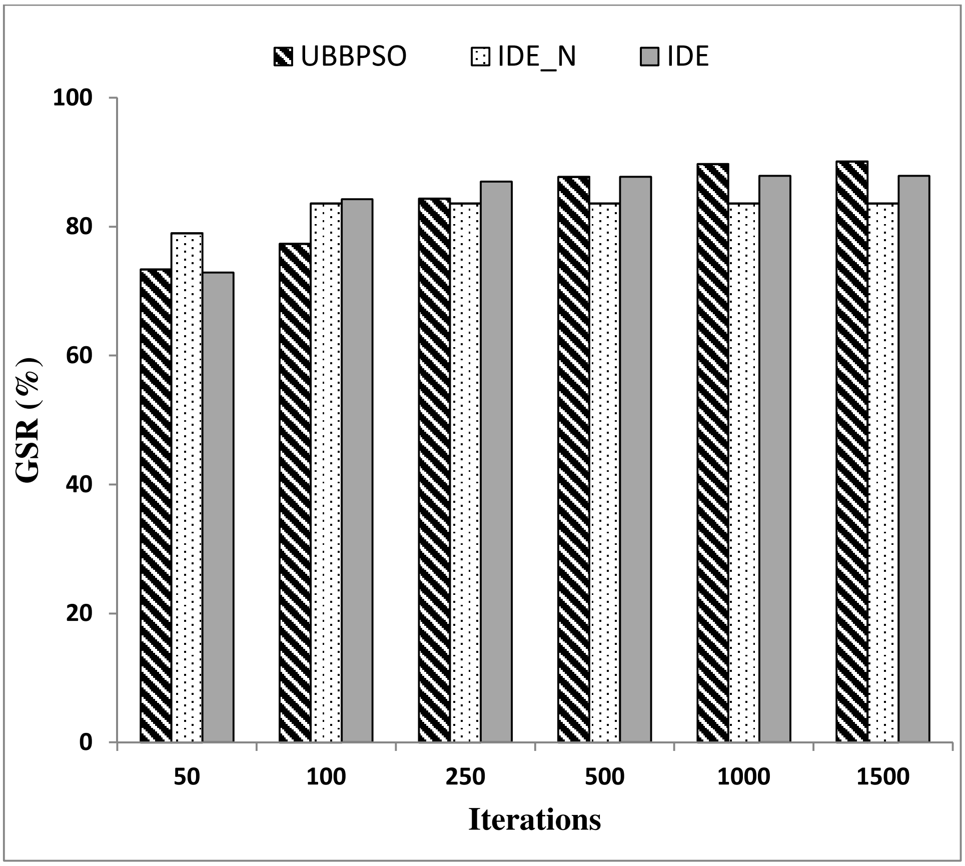


Figure 5.3. Global Success Rate (GSR) Versus Iterations for PS Problems Using UBBPSO, IDE_N and IDE with SC-1

Results on the effect of stopping criterion, SC-2 with SC_{max} = 10, 25 and 50 on IDE, IDE_N and UBBPSO for all PS problems are presented in Table 5.4. They show that reliability of the algorithm and NFE increase with increasing SC_{max}. For PS problems 1 and 5, IDE_N obtained the best reliability followed by IDE algorithm. For PS problems 2, 3, 4 and 7, the three algorithms obtained 100% SR. For PS problems 6 and 8, UBBPSO obtained the best reliability followed by IDE. However, UBBPSO and IDE_N require significantly more NFE than IDE. Although the same stopping criterion is used for the three algorithms, NFE required by IDE is much less than that of IDE_N and UBBPSO. For example, for PS-8 with SC_{max} = 50, UBBPSO needs 147533 NFE but IDE requires only 70996 NFE. As shown in Table 5.4, the total NFE required for all PS problems by IDE is 260007 compared to 332752 and 1115786 required by IDE_N and UBBPSO respectively. Thus, UBBPSO requires more than 4 times the total NFE required by IDE. Hence, IDE has faster convergence compared to IDE_N and UBBPSO, probably due to tabu list and checking.

Table 5.4. Success Rate (SR) and Number of Function Evaluations of UBBPSO, IDE_N and IDE for PS Problems Using SCmax with NP of 10D

PS		UBBPSO		IDE_N		IDE	
No.	SCmax	SR	NFE	SR	NFE	SR	NFE
PS-1	10	85	1274	93	594	89	582
	25	97	3686	100	2518	99	2631
	50	99	6051	100	3553	100	3789
PS-2	10	100	2536	100	2227	100	1448
	25	100	7880	100	5756	100	6536
	50	100	10515	100	6571	100	9053
PS-3	10	100	3342	100	1741	100	1250
	25	100	8906	100	4721	100	4747
	50	100	14856	100	6144	100	7619
PS-4	10	100	827	100	822	100	781
	25	100	2310	100	2339	100	2437
	50	100	4096	100	3369	100	3321
PS-5	10	56	21272	69	4033	46	2051
	25	78	31777	99	10573	82	10055
	50	92	44606	100	13285	99	19487
PS-6	10	73	74920	47	9296	45	3633
	25	90	93490	62	25816	72	15148
	50	94	105846	62	30045	76	38145
PS-7	10	100	64859	100	18020	100	5417
	25	100	90780	100	45681	100	10253
	50	100	105697	100	50074	100	14161
PS-8	10	25	127894	7	17033	9	5470
	25	26	140833	7	31647	10	20997
	50	26	147533	7	36894	21	70996

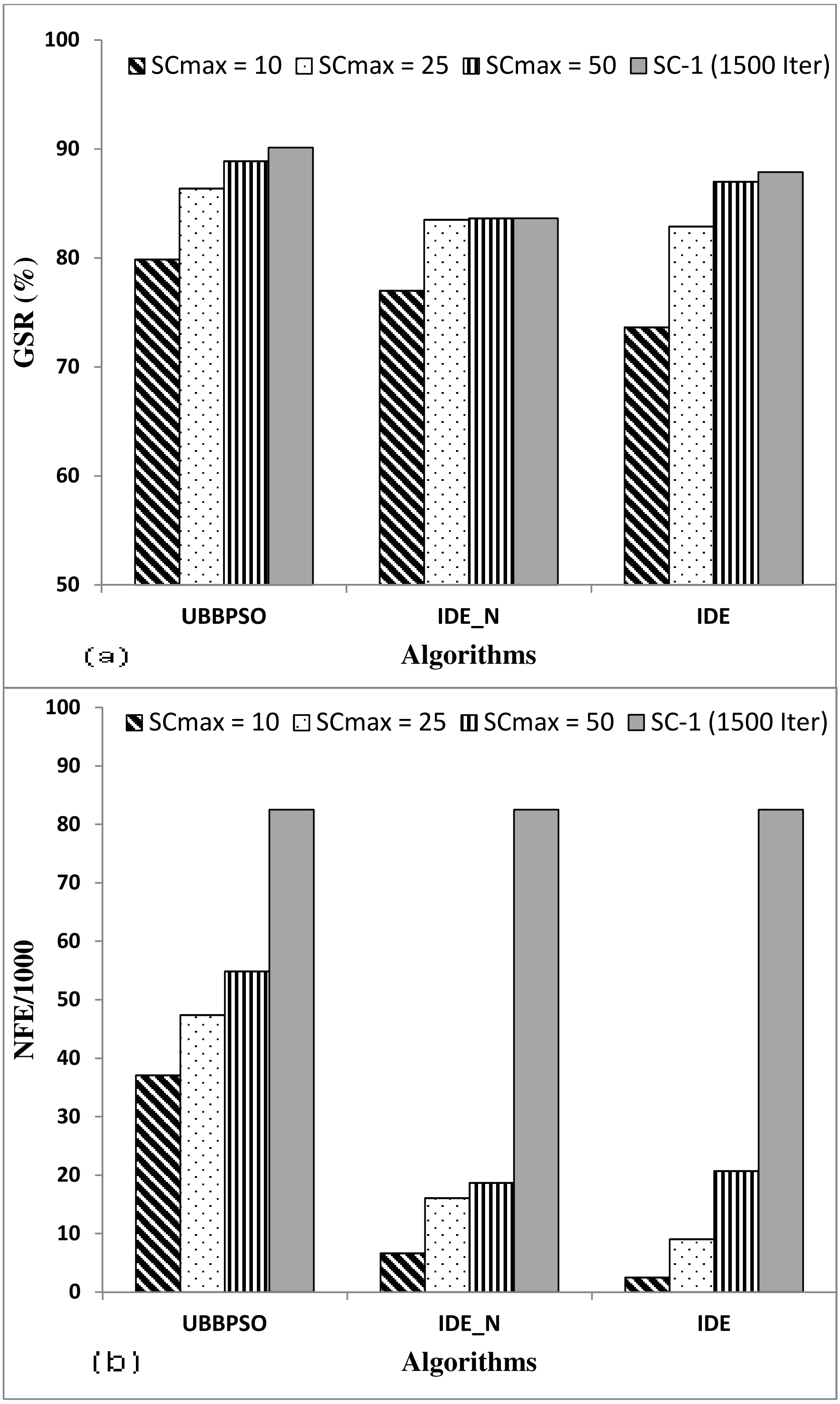


Figure 5.4. Global Success Rate, GSR (plot a) and Average NFE (plot b) of UBBPSO, IDE_N and IDE for PS Problems Using SC-2 (SC_{max} = 10, SC_{max} = 25 and SC_{max} = 50) and SC-1 (1500 Iterations)

Fig. 5.4 summarizes GSR and NFE of IDE, IDE_N and UBBPSO with four stopping criteria for PS problems. We obtain the same conclusion of higher reliability and increased NFE with higher SC_{max}. It can be seen from Fig. 5.4a that the use of SC-2 gives similar GSR compared to SC-1; especially, GSR of IDE_N with different stopping criteria is similar. Among the three algorithms, IDE uses least NFE to terminate the global search in case of SC_{max} = 10 and SC_{max} = 25 whereas IDE_N uses the least NFE to terminate the global search progress for SC_{max} = 50 (Fig. 5.4b). In general, use of SC-2 results in significantly reduced NFE compared to that using SC-1. As for PEC problems, it can be concluded that IDE is better and that SC_{max} = 50 is a good stopping criterion for PS problems in order to achieve both high reliability and efficiency.

Low SR is obtained for PS-5, 6 and 8 compared to other PS problems (Table 5.4). This is probably because of using the same NP = 10D for comparing the three algorithms studied in this paper. In general, NP is a user-specified parameter; it does not need to be fine-tuned and just a few typical values can be tried according to the pre-estimated complexity of the given problem. So PS-5, 6 and 8 are solved by IDE with higher population size of NP = 50D. The results show that IDE can obtain 100% SR using 24930 NFE for PS-5, 100% SR using 61820 NFE for PS-6, and 76% SR using 384416 NFE for PS-8. It is clear that reliability of the algorithm increases with population size but this requires more computational effort. This is reasonable because larger population size enables more thorough exploration of search space. In general, stochastic optimization methods provide only a probabilistic guarantee of locating the global optimum, and their proofs for numerical convergence usually state that the global optimum will be identified in infinite time with probability 1 (Rudolph, 1994; Niewierowicz et al., 2003; Weise, 2008). So, better performance of stochastic methods is expected if more iterations and/or larger population size are used.

To analyze further, performance of IDE is compared in Table 5.5 with that of other stochastic optimization algorithms, namely, SA, very fast SA (VFSA), modified version of direct search SA (MDSA) and stochastic differential equations algorithm (SDE) for PS-3, 5, 6 and 8 reported in Bonilla-Petriciolet et al. (2006). From this table, it is clear that IDE uses about 10 times fewer NFE compared to the other four stochastic methods for solving PS-3, 5 and 6 with the same reliability. SR of IDE is lower for PS-8 using significantly less NFE, and this may due to the different stopping criteria and population size used in the algorithms. Note that some methods may give better reliability using the stopping criteria based on either known global optimum or the number of generations (Bonilla-Petriciolet et al., 2006). However, the use of known global optimum is not applicable for new problems whose global optima are unknown, and use of number of generations may require large computational time.

Table 5.5 Comparison of SR and NFE of IDE with Other Stochastic Algorithms for Selected PS Problems

PS						IDE	IDE
No.		MDSA	SA	VFSA	SDE	(NP = 10D)	(NP = 50D)
PS-3	NFE	82611	92422	42944	449574	1250	-
	SR	100	100	100	100	100	-
PS-5	NFE	263980	266926	129534	565142	19487	24930
	SR	100	100	96	100	99	100
PS-6	NFE	752571	700865	362097	476029	38145	61820
	SR	100	100	100	100	76	100
PS-8	NFE	1167211	1104901	581396	958515	70996	384416
	SR	100	100	98	98	21	76

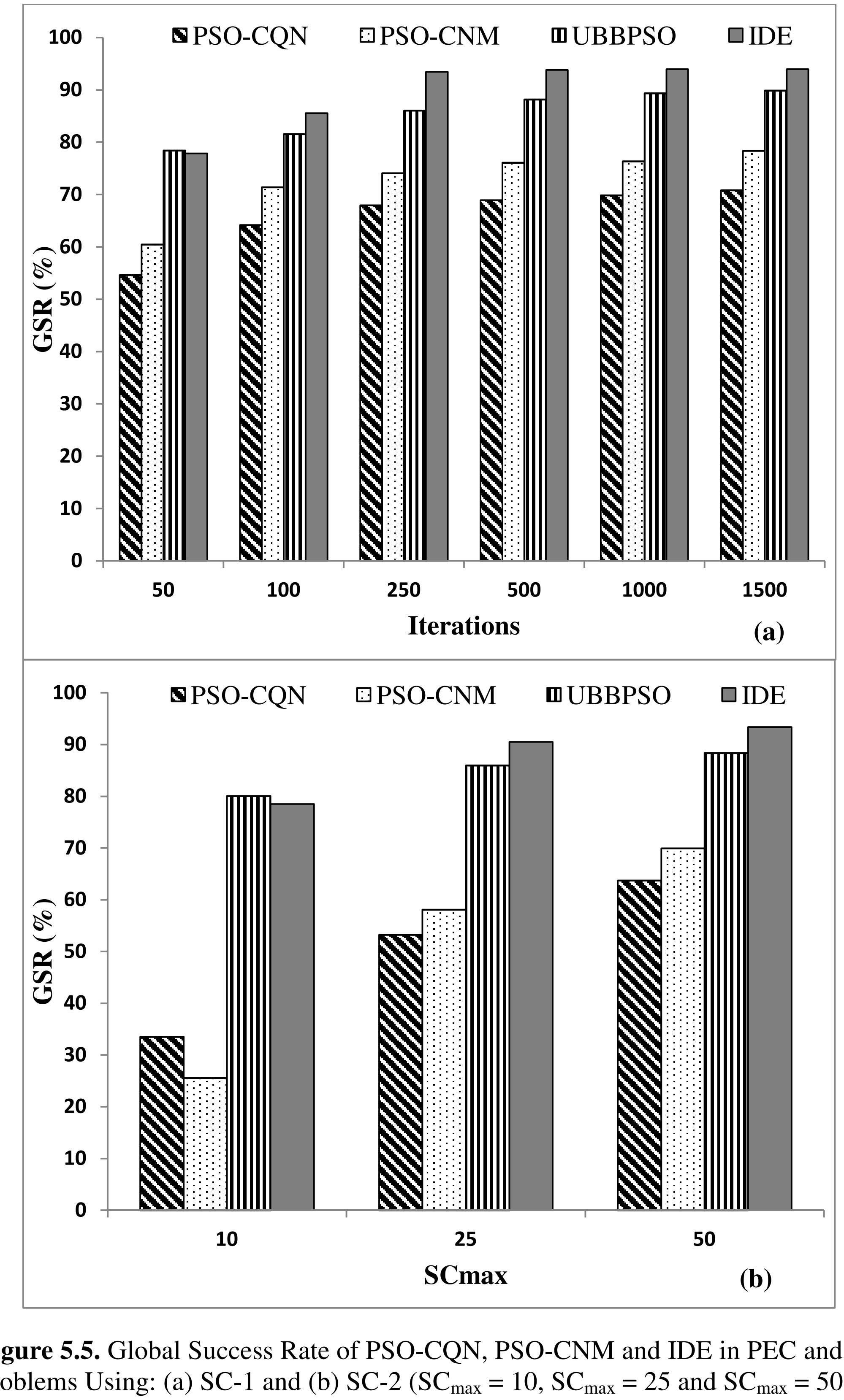


Figure 5.5. Global Success Rate of PSO-CQN, PSO-CNM and IDE in PEC and PS Problems Using: (a) SC-1 and (b) SC-2 (SC_{max} = 10, SC_{max} = 25 and SC_{max} = 50) as Stopping Criteria.

Bonilla-Petriciolet et al. (2010) compared various PSO algorithms for solving the PEC and PS problems. Their results suggest that the classical PSO outperforms other variants of PSO. So, the performance of IDE and UBBPSO is compared with two classical PSO algorithms reported in Bonilla-Petriciolet et al. (2010) for both PEC and PS problems, with different types of stopping criteria in Fig. 5.5. The two PSO algorithms are classical PSO with quasi-Newton method (PSO-CQN) and classical PSO with Nelder-Mead simplex method (PSO-CNM). Fig. 5.5a shows that IDE achieved the best reliability at 100 iterations or more, compared to UBBPSO, PSO-CQN and PSO-CNM. UBBPSO achieved the second best reliability at different iteration levels among the four algorithms tested with SC-1. The reliability comparison of the four algorithms with SC-2 stopping criterion is shown in Fig 5.5b. IDE gave the highest GSR even with SC-2 as the stopping criterion with SC_{max} = 25 and 50, among the four algorithms. With stopping criterion, SC_{max} = 10, UBBPSO obtained slightly better GSR than IDE but much better GSR than PSO-CQN and PSO-CNM. Overall, IDE is superior to UBBPSO, PSO-CQN and PSO-CNM algorithms for PEC and PS problems.

5.4.3 Performance of Algorithms on rPEC problems

GSR of UBBPSO (Zhang et al., 2011c), IDE_N and IDE algorithms for all rPEC problems using SC-1 is illustrated in Fig. 5.6. It generally improves with increasing number of iterations for these problems as well. The highest GSR is 91% obtained by IDE. At 50 iterations, IDE_N obtained best GSR, but from 250 to 1500 iterations, its GSR did not improve; on the other hand, UBBPSO and IDE obtained better GSR at higher iterations. GSR of UBBPSO is 80% at 50 iterations, and it increases to 87% at 250 iterations and 90% at 1500 iterations. GSR of IDE is 77%, 85%

and 91% at 50, 250 and 1500 iterations, respectively. In short, UBBPSO, IDE_N and IDE algorithms obtained good GSR of 80% or more (Fig. 5.6).

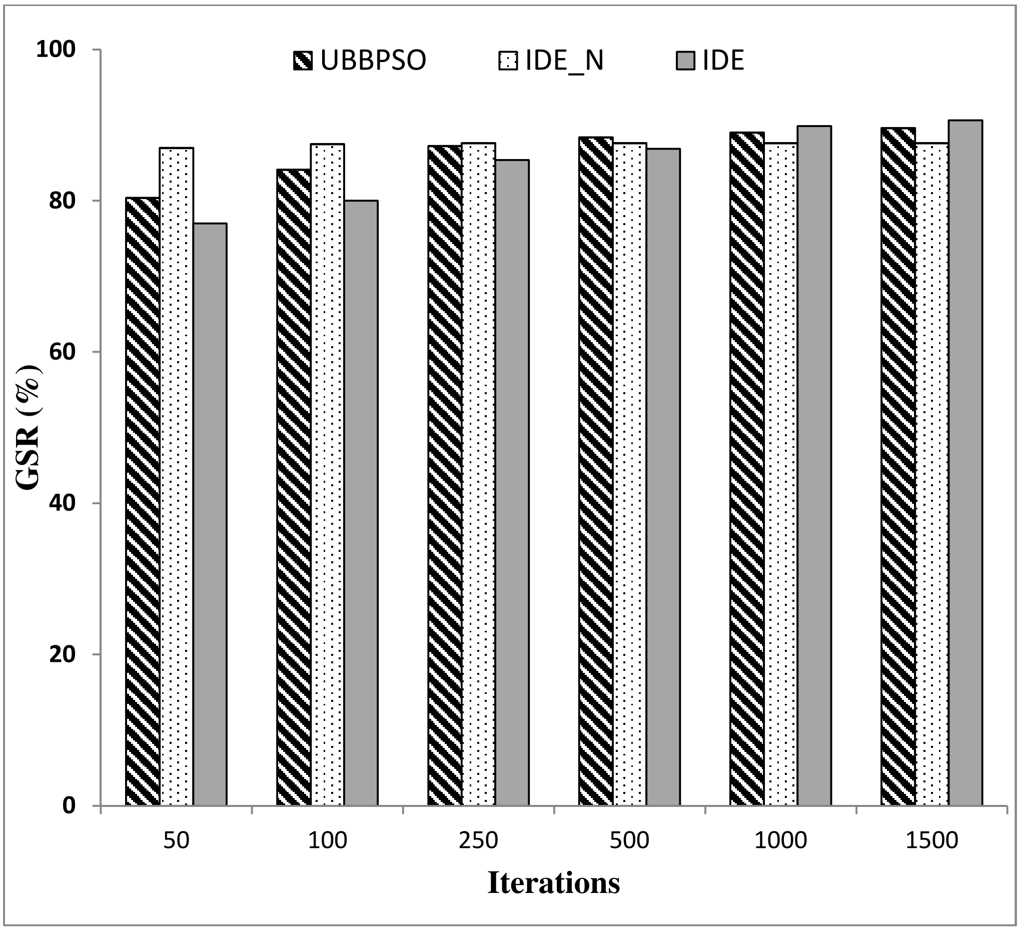


Figure 5.6. Global Success Rate (GSR) Versus Iterations for rPEC Problems Using UBBPSO, IDE_N and IDE with SC-1

Results obtained on the effect of stopping criteria on the three algorithms using SC-2 with SC_{max} = 6D, 12D and 24D, for rPEC problems are summarized in Table 5.6. Note that these SC_{max} values are those used by Bonilla-Petriciolet et al. (2011a) so that the present results can be compared with those in this reference. Table 5.6 shows that reliability of the algorithm increases with SC_{max} but requires more NFE. For rPEC-1, 2, 3, 6 and 8, UBBPSO, IDE_N and IDE algorithms obtained 100% SR with different SC_{max} tried. For rPEC problem 4, IDE obtained the best reliability followed by UBBPSO and IDE_N, but its SR is only 17%. So rPEC-4 is solved by IDE using larger population of NP = 50D. The results show that IDE can obtain 88%

SR at 118108 NFE. Among the three algorithms, NFE required by IDE is much less than that by IDE_N and UBBPSO. As shown in Table 5.5, total NFE required by IDE for all rPEC problems is 325487 compared to 389885 by IDE_N and 878907 by UBBPSO. On the other hand, mean SR for the three algorithms is almost the same. Thus, the results show that IDE has faster convergence compared to IDE_N and UBBPSO for the same reliability.

Table 5.6. Success Rate (SR) and Number of Function Evaluations of UBBPSO, IDE_N and IDE for rPEC Problems Using SCmax with NP of 10D

rPEC		UBBPSO		IDE_N		IDE	
No.	SCmax	SR	NFE	SR	NFE	SR	NFE
rPEC-1	6D	100	21585	100	14484	100	5585
	12D	100	47720	100	16102	100	14720
	24D	100	68953	100	19387	100	16641
rPEC-2	6D	100	33794	100	13012	100	12698
	12D	100	61252	100	14511	100	16559
	24D	100	70037	100	17511	100	18885
rPEC-3	6D	100	14896	100	14928	100	8143
	12D	100	29376	100	16461	100	14157
	24D	100	47986	100	19460	100	27034
rPEC-4	6D	10	23107	1	11676	9	10312
	12D	13	34655	1	14394	17	13765
	24D	14	44068	1	19014	17	20763
rPEC-5	6D	93	14792	98	8794	90	2644
	12D	98	32370	99	12160	98	5664
	24D	98	43177	100	15793	100	11264
rPEC-6	6D	100	39571	100	18361	100	19504
	12D	100	43742	100	22175	100	21135
	24D	100	50421	100	28628	100	29150
rPEC-7	6D	90	16129	99	9229	90	2546
	12D	97	28041	100	11945	97	4680
	24D	98	33326	100	14001	98	10330
rPEC-8	6D	100	22954	100	15487	100	5780
	12D	100	25941	100	18652	100	10411
	24D	100	31014	100	23720	100	23117

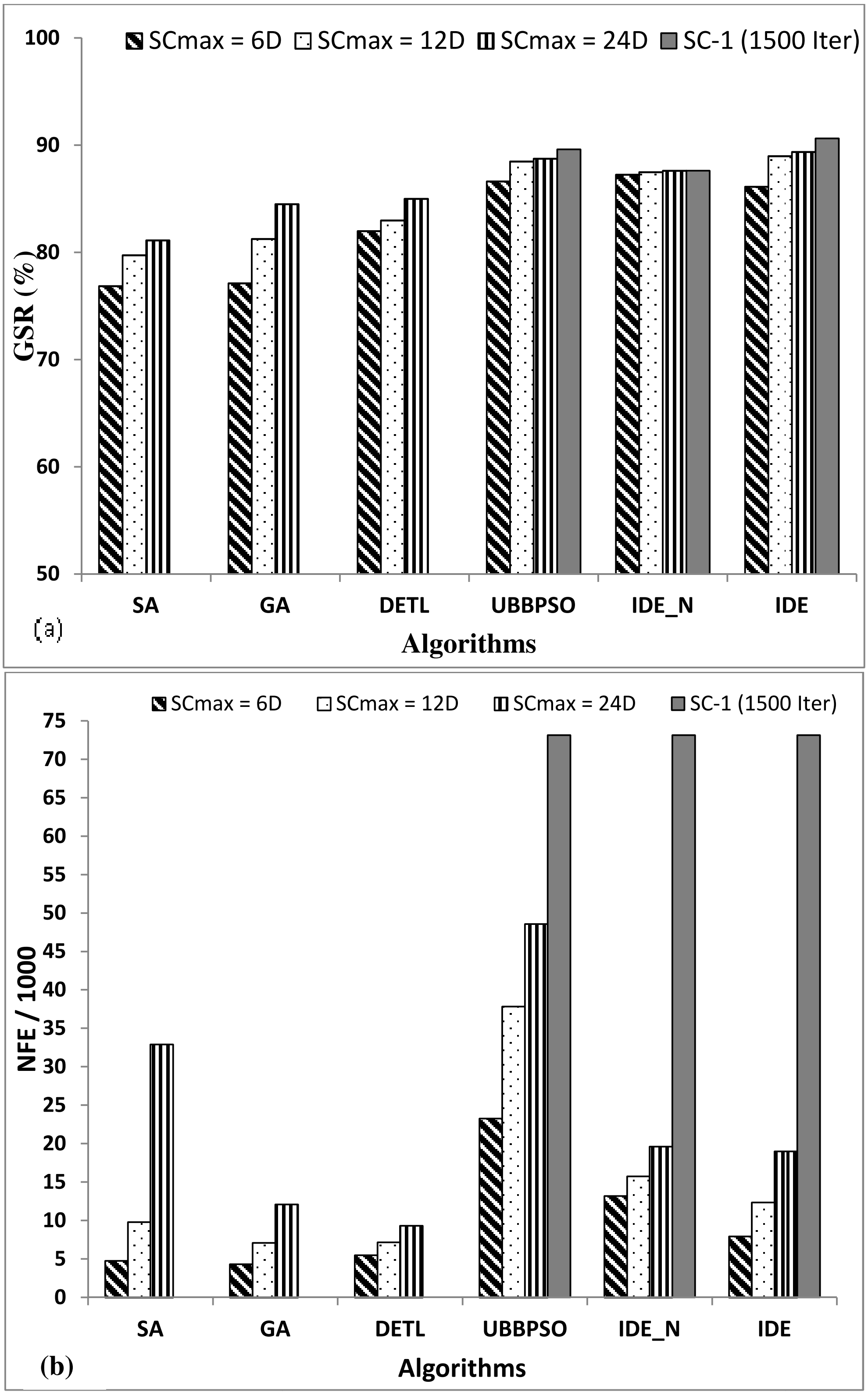


Figure 5.7. Global Success Rate, GSR (plot a) and Average NFE (plot b) of UBBPSO, IDE_N, IDE, SA, GA and DETL for rPEC Problems Using SC-2 (SC_{max} = 6D, SC_{max} = 12D and SC_{max} = 24D) and SC-1 (1500 Iterations)

Fig. 5.7 shows GSR and NFE of IDE, IDE_N and UBBPSO with different stopping criteria for rPEC problems. Again, we conclude that the higher the SC_{max}, the better the reliability of the algorithm is, and that the use of SC-2 gives similar GSR compared to SC-1 (Fig. 5.7a). However, IDE_N gave almost the same GSR with different stopping criteria. Among IDE, IDE_N and UBBPSO, IDE uses least NFE to terminate the global search via SC-2. Between IDE and IDE_N, results show that IDE provides better reliability with fewer NFE (Fig. 5.7), probably due to enhancement of global search ability by tabu list and tabu check operations in IDE. For optimal efficiency and reliability of IDE, SC_{max} = 12D is a good stopping criterion for rPEC problems.

Recently, SA, DETL and GA have been tested for rPEC problems in Bonilla-Petriciolet et al. (2011a). All these stochastic algorithms were run 100 times independently in FORTRAN environment. At the end of every run, a deterministic local optimizer (namely, DBCONF of IMSL library) was activated. The performance of IDE is compared with that GA, SA and DETL for rPEC problems, in Table 5.7. The data of IDE in Table 5.7 is slightly different from those in Table 5.5 because NFE given in Bonilla-Petriciolet et al. (2011a) is based on successful runs only (and not all runs). For fair comparison, similar data of IDE are given in Table 5.7. From this table, it is clear that IDE achieved the best reliability compared to SA, GA and DETL at SC_{max} = 6D, 12D and 24D. For example, for rPEC problems 5 and 7, IDE obtained much better SR than other algorithms compared. However, IDE requires more NFE compared to GA and DETL. This indicates that GA and DETL have faster convergence rate for rPEC problems but they can be trapped at the local optimum in several runs. The computational time (in seconds) for solving rPEC problems using IDE algorithm with SCmax = 6D, 12D and 24D is reported in the last column of

Table 5.7. Obviously, CPU times increases with NFE. Further, there is nearly linear relationship between NFE and CPU time, which supports the use of NFE for comparison.

Table 5.7. Success Rate (SR) and Number of Function Evaluations of SA, GA, DETL and IDE for rPEC Problems Using SCmax with NP of 10D

		SA		GA		DETL		IDE		
rPEC No.	SCmax	SR	NFE	SR	NFE	SR	NFE	SR	NFE	Time (s)
rPEC-1	6D	93	5544	89	4650	100	7791	100	5585	3
	12D	99	10818	98	7866	100	9787	100	14720	7
	24D	100	44083	99	13228	100	11548	100	16641	9
rPEC-2	6D	98	4703	99	4465	100	4237	100	12698	6
	12D	100	10159	100	7363	100	5708	100	16559	10
	24D	100	35837	100	12464	100	6665	100	18885	12
rPEC-3	6D	95	4740	98	4670	100	6366	100	8143	5
	12D	100	9299	100	7484	100	8017	100	14157	8
	24D	100	32630	100	12680	100	10893	100	27034	15
rPEC-4	6D	11	4460	15	4234	1	10621	9	6799	3
	12D	10	8396	11	7065	1	15420	17	11647	6
	24D	11	29589	17	12136	2	22043	17	11652	6
rPEC-5	6D	71	2401	63	2403	88	2235	90	2706	2
	12D	76	4725	73	4224	86	2705	98	5685	4
	24D	79	9508	70	8292	92	3205	100	11264	7
rPEC-6	6D	99	8038	100	6420	100	4859	100	19504	11
	12D	100	18759	100	10708	100	5994	100	21135	14
	24D	100	66336	100	17623	100	8532	100	29150	19
rPEC-7	6D	56	2690	58	2913	65	3955	90	2573	1
	12D	55	5389	70	4584	77	4985	97	4682	2
	24D	61	13599	90	8257	85	5713	98	10338	5
rPEC-8	6D	92	5733	95	4906	100	4057	100	5780	2
	12D	98	11096	98	7777	100	5036	100	10411	4
	24D	98	31887	100	12420	100	6124	100	23117	10

For rPEC 4, at SCmax = 6D, GA and SA are better than IDE in terms of both SR and NFE. When SCmax = 12D, IDE is more reliable than GA and SA but with more NFE. When SCmax = 24D, the IDE is better than GA and SA in both SR and NFE. This is probably due to the self-adaptive strategy of IDE which needs more generations initially for adaptive tuning of algorithm parameters. For high reliability of GSR = 85%, GA requires SC_{max} of 24D and NFE of 12138 whereas IDE requires SC_{max} of 12D and NFE of 12375 to obtain 89% GSR. This shows that IDE uses 2% more NFE to improve GSR by 4%. Further, IDE has fewer parameters to be tuned, which makes the algorithm more robust. In summary, the present results indicate that IDE offers a good balance between diversification and intensification stages for reliable and efficient phase equilibrium calculations in both reactive and non-reactive systems. Compared to other stochastic methods, its reliability and efficiency are generally better for solving these thermodynamic problems.

5.5 Conclusions

The stochastic global optimization algorithms, namely, IDE_N and IDE studied in this work have fewer parameters to be tuned. The performance of these algorithms has been tested and compared for solving PEC, rPEC and PS problems. IDE was found to be the overall best performer across different problems tried. Results for IDE and IDE_N confirm that use of tabu radius and tabu list improves reliability and decreases computational effort although it involves two parameters. Comparison of IDE with PSO variants for both PEC and PS problems shows that IDE provides higher reliability and efficiency. Comparison of IDE with SA, GA and DETL suggests that the former provides higher reliability for rPEC problems. The stopping criterion, SC-1 gives slightly better reliability than SC-2 at the expense of computational resources, and the use of SC_{max} can significantly reduce the computational effort for solving PEC, rPEC and PS problems without much effect on the reliability of the stochastic algorithms studied.

Chapter 6

An Efficient Constraint Handling Method *

6.1 Introduction

Nowadays, optimization is a necessity in almost every field such as business, science and engineering. In every area, effective optimization techniques are required for improving the performance of applications and processes. To achieve this goal, we need to have a mathematical model for the application and an objective function that depends on decision variables. A typical optimization problem has an objective function, equality/inequality constraints and upper/lower bounds on decision variables. Most of the practical optimization problems are nonlinear and non-convex in either the objective and/or constraints, and so optimization of such problems requires a global optimization method. Hence, the study of global optimization for constrained optimization problems is an active research area (Angira and Babu, 2006; Babu and Angira, 2006; Liu et al., 2010; Luo et al., 2007; Srinivas and Rangaiah, 2007; Zahara and Kao, 2009).

This study considers the following optimization problem with equality and inequality constraints, and upper and lower bounds.

$$\begin{aligned} &\text{Minimize } f(x) \\ &\text{Subject to } h_i(x) = 0, i = 1,2,...,m1 \\ &\qquad g_j(x) \le 0, j = 1,2,...,m2 \\ &\qquad x_k^l \le x_k \le x_k^u, k = 1,2,...,n \end{aligned} \tag{6.1}$$

* This chapter is based on the papers - Zhang, H. and Rangaiah, G.P., An Efficient constraint Handling Method with Integrated Differential evolution for Numerical and Engineering optimization. *Comp. & Chem. Eng.*, 2011, in press.

Here, x is the n-dimensional vector of decision variables, $f(x)$ is the objective function, $h_i(x) = 0$ and $g_j(x) \leq 0$ are respectively $m1$ equality constraints and $m2$ inequality, and x_k^l and x_k^u are respectively the lower and upper bounds of x_k.

Global optimization methods can be classified into two broad categories: deterministic and stochastic methods (Pardalos et al., 2000). The former methods include outer approximation methods and branch & bound methods, and have been proposed for solving constrained optimization problems (Floudas et al., 1989; Kocis & Grossmann, 1998; Ryoo & Sahinidis, 1995). The stochastic algorithms include SA, GA, TS, DE, PSO and ACO, and they have been applied to constrained optimization problems too (Das et al., 1990; Jayaraman et al., 2000; Lampinen, 2002; Lin & Miller, 2004; Srinivas & Rangaiah, 2007; Yu et al., 2000). The deterministic methods can provide a guaranteed global optimal solution but they require certain properties of objective function and constraints such as continuity and convexity. The stochastic methods require little or no assumption on the characteristics of the optimization problem, provide probabilistic convergence to the global optimum, are usually simple in principle and easy to implement. However, they do not guarantee global optimality in finite iterations although they can often locate the global optimum in modest computational time compared to deterministic methods (Lin & Miller, 2004). One of the major challenges of the stochastic methods is handling constraints, particularly equality constraints.

Recently, many researchers have studied the solution of constrained optimization problems using stochastic algorithms (Kheawhom, 2010; Liu et al., 2010; Luo et al., 2007; Srinivas & Rangaiah, 2007; Yuan & Qian, 2010; Zahara & Kao, 2009). Srinivas and Rangaiah (2007) employed DETL, which incorporated the useful concept of TS to avoid revisits during the global search in order to improve computational efficiency and reliability, and the penalty function method to handle inequality constraints for solving nonlinear programming (NLP) and mixed-integer NLP (MINLP) problems. Luo et al. (2007) proposed an improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints. The proposed method requires elimination of equality constraints by partitioning variables and identifying reduced variables for optimization through analysis and tearing equality constraints. Zahara and Kao (2009) proposed a hybrid Nelder-Mead simplex search and PSO for constrained engineering design problems. It embeds a constraint handling method which includes the gradient repair method and ranking based on constraint fitness.

Yuan and Qian (2010) proposed genetic algorithm with local solver to solve twice-differentiable NLP problems; their results show that the proposed method can solve such NLP problems with inequality constraints. Liu et al. (2010) proposed hybrid PSO with DE (PSO-DE) to solve constrained optimization problems. It uses DE to update the previous best positions of the particles to force PSO jump out of stagnation. The constraints are handled by minimizing the objective function and the degree of constraint violation simultaneously with two populations of identical size. Kheawhom (2010) proposed a constraint handling scheme for DE to solve chemical engineering optimization problems. It uses a repair algorithm based on the gradient information derived from the equality constraints, to correct infeasible solutions. The dominance-based selection (feasibility approach) is also used to handle constraints. Most of the problems studied in the works cited in this and previous paragraph are inequality constrained problems, and equality constraints, if present, are eliminated by some way.

Thus, global optimization of problems with equality constraints is still challenging without elimination of equality constraints. Hence, the present study focuses on finding the global optimum of equality constrained optimization problems without any re-formulation to eliminate equality constraints.

Among many stochastic global optimization algorithms, DE is a population-based global optimization method proposed by Storn and Price (Price et al., 2005; Storn & Price, 1997). Its principle is instinctive to understand, number of parameters involved is fewer compared to other algorithms, has a relatively faster convergence and high reliability to find the global optimum (Babu & Munawar, 2007; Srinivas & Rangaiah, 2006). The original DE is designed for unconstrained optimization problems. In the recent past, many DE variants have been proposed to improve its performance. Many of them are on two aspects: adapting DE parameters (Brest et al., 2006; Qin et al., 2009), and hybridization to combine DE with another optimization method to enhance the performance (Angira & Babu, 2006; Babu & Munawar, 2007; Liu et al., 2010; Srinivas & Rangaiah, 2007; Zhang & Rangaiah, 2011). Our recent algorithm (Zhang & Rangaiah, 2011), referred to as integrated DE (IDE), combines both these aspects by integrating the tabu list to avoid revisits (Srinivas & Rangaiah, 2007), and self-adaptation of parameters and mutation strategy (Qin et al., 2009). In addition, a novel stopping criterion is included in IDE to terminate the global search after finding the minimum approximately and then a local optimizer is used to find the minimum accurately and efficiently (Zhang et al., 2011).

When the problems contain constraints, the feasible region is reduced, leading to many difficulties in solving them. In this study, we propose an efficient constraint handling method, particularly for equality constraints, for stochastic global optimization methods. It incorporates adaptive constraint relaxation method with the feasibility approach of Deb (2000) to handle the relaxed constraints; in this, both the equality and inequality constraints are relaxed in order to increase temporarily the feasible region in the initial exploration. Then, the feasible region is gradually shrunk according to the fraction of feasible individuals in the population.

The focus and contribution of this study are on a new constraint handling method and its use with IDE algorithm. The proposed constrained IDE, C-IDE is tested on benchmark problems with equality and/or inequality constraints and its performance is compared with the recent techniques. Subsequently, it is applied to chemical engineering application problems with equality/inequality constraints. The remainder of this chapter is organized as follows. Strategies for handling equality and inequality constraints are outlined in Section 6.2. Section 6.3 describes the proposed constraint relaxation method to handle the constraints. The IDE algorithm incorporating this constraint handling method is described in Section 6.4. The numerical experiments and results on the benchmark problems are presented and discussed in Section 6.5. Section 6.6 discusses performance of the C-IDE on chemical engineering application, and compares it with other methods. Finally, the conclusions of this work are summarized in Section 6.7.

6.2 Methods for Handling Constraints

Many chemical engineering applications such as process design, synthesis, control and scheduling involve formulating and solving optimization problems with equality constraints besides inequality constraints and bounds on variables. Of the constraints, equality constraints are more challenging to handle due to the tiny feasible region. In principle, they can be used to reduce the number of variables in the optimization problem by elimination procedure. However, identifying the reduced variables is hard, some equality constraints are irreducible and cannot be used to decrease the problem dimension, and the transformed problem may be more difficult to solve.

The methods for handling constraints in the stochastic global optimization methods have received much attention in the last decade. They can be divided into five categories: (1) Penalty functions, (2) Separation of objectives and constraints, (3) Special representations and operators, (4) Repair algorithms, and (5) Hybrid methods (Coello Coello, 2002). The first two categories are widely applied for solving constrained problems, and the last three categories are infrequently used (Lampinen, 2002). The main disadvantage of penalty function method is the difficulty to choose a suitable penalty parameter because it is problem-dependent. The main advantage of the second method is that it does not require any parameter to handle constraints, and the results show that it is very effective to handle inequality constraints. It is also referred to as the feasibility approach (Deb, 2000; Lampinen, 2002); its disadvantage is the difficulty to maintain population diversity, which may cause premature convergence (Coello Coello, 2002).

In the recent past, many researchers have been using or modifying the feasibility approach to solve the constrained problems by stochastic global optimization algorithms. He and Wang (2007) proposed a hybrid PSO (HPSO) to solve constrained global optimization problems; it combines PSO with simulated annealing in order to enhance the search ability and prevent premature convergence. The feasibility approach is used for handling the constraints. The results show that the proposed algorithm has better searching quality and robustness for constrained engineering design problems. Takahama and Sakai (2006) proposed the ε-constrained DE with gradient-based mutation and feasible elites (εDE). It combines ε-constrained method and DE with gradient-based mutation that finds a feasible point using the gradient of constraints at an infeasible point. In the ε-constraint handling method, the relaxation of constraints is controlled by using the ε parameter. The proper control of ε is essential in order to obtain high-quality solutions for problems with equality constraints (Takahama & Sakai, 2006). The ε value is updated until the generation counter G reaches the control generation T_c (eq. 3). After the generation counter exceeds T_c, ε is set to zero to obtain solutions with no constraint violation.

$$\varepsilon(0) = \delta(x_k) \tag{6.2}$$

$$\varepsilon(t) = \begin{cases} \varepsilon(0)(1 - \frac{G}{T_c})^{cp} & 0 < G < T_c \\ 0 & G \geq T_\mathrm{c} \end{cases} \tag{6.3}$$

where $\delta(x)$ is the constraints violation and x_k is the top k^{th} individual and k = 0.2NP, where NP is the population size. The recommended parameter ranges are $T_c \in [0.1G_{\max}, 0.8G_{\max}]$ and $cp \in [2, 10]$ (Takahama & Sakai, 2006).

The selection of individuals in the evolution by using the ε-constraint technique is similar to the feasibility approach. The results in Takahama and Sakai (2006) show that εDE algorithm can handle problems with small feasible region and equality constraints but it has more parameters compared to others (Eqs. 6.2 and 6.3). Ali and Kajee-Bagdadi (2009) proposed local exploration-based DE (LEDE) method with feasibility approach and parameter-free penalty method for constrained global optimization. They tested the proposed algorithm on many benchmark problems, and the results show that it outperforms other algorithms for constrained optimization problems.

6.3 Proposed Constraint Handling Method

In this study, we propose an effective constraint handling method to handle both equality and inequality constraints, for use with IDE and other stochastic algorithms. It incorporates adaptive relaxation of constraints and the feasibility approach in the selection step of IDE. In the initialization step, values of the objective function and constraints of the whole population are calculated. The total absolute violation ($\mathrm{TAV_k}$) of individual k is defined as

$$\mathrm{TAV}_k = \sum_{i=1}^{m1} |h_i(\mathrm{x})| + \sum_{j=1}^{m2} \max(0, \mathrm{g}_j(\mathrm{x})) \quad k = 1, 2, \dots, NP \tag{6.4}$$

which includes violation, if any, of all constraints. The median of $\mathrm{TAV_k}$ of all individuals in the population is assigned to the initial constraint relaxation value (μ).

If the total absolute violation of all constraints (and not violation of a constraint) of an individual is less than μ, then it is temporarily treated as a feasible solution; else, it is taken as an infeasible solution. Thus, relaxation is for all equality and inequality constraints together. During the generations, μ is gradually reduced according to the fraction of feasible individuals ($\mathrm{F_F}$) with respect to the relaxed constraints, in the latest population at generation, G. After considering and testing several possibilities, we propose the following equation for reducing μ for the next generation, G+1.

$$\mu(\mathrm{G}+1) = \mu(\mathrm{G})(1 - \frac{\mathrm{F_F}}{\mathrm{NP}}) \tag{6.5}$$

The relaxation of constraints in the initial generations helps greater exploration of the search space for locating the global optimum region. The extent of relaxation is related to the feasibility of the population. When fewer individuals in the current population are feasible, rate of reducing μ will be slower. Hence, more generations are required to find the feasible and better solutions. For example, when all individuals in the current population are infeasible, μ will not be reduced until some feasible individuals are found. Without checking the feasibility of individuals in the population and using this information to reduce μ may have faster convergence but it is likely to lead to premature convergence and failure to find the global optimum. In the proposed adaptive relaxation, μ is dynamically reduced based on the fraction of (temporarily) feasible solutions in the population but it will not be increased (Eq. 6.5). This is because increasing μ will result in individuals with higher constraint violations, which will slow the convergence of the algorithm.

It is clear, when μ is near 0, the rate of decrease of μ can be low. However, this makes the method more robust since it is still based on the performance of the population. In general, some problems can have very large constraint violation, and so the rate of decrease of μ will be fast. Other problems can have very small constraint violation, and so the rate of decrease of μ will be slow. Therefore, we cannot apply the same rate of decrease of μ to different problems. It is better, as in the proposed method, to decrease μ adaptively according to constraint violations and thus for different problems. The proposed adaptive relaxation by Eq. 6.5 will effectively control the extent of relaxing and enhance the global search. Compared to the εDE, it has no parameter, and relaxation is adapted based on search performance.

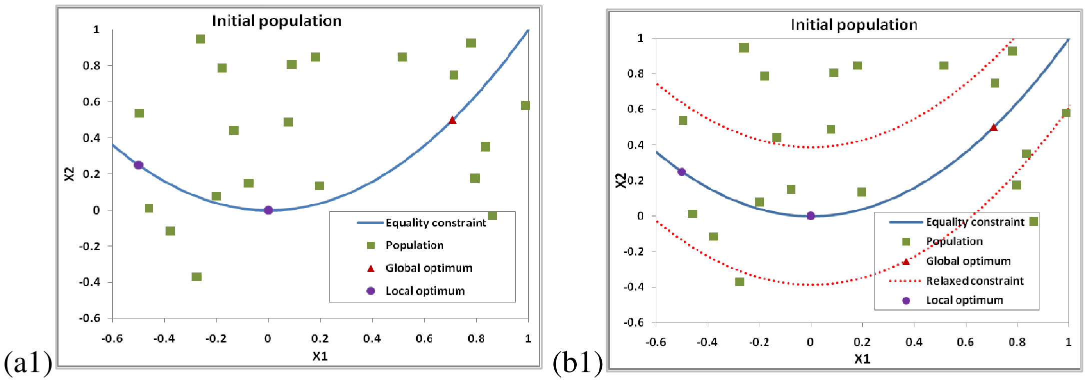
The selection between target and trial individuals in the C-IDE algorithm is based on the feasibility approach of Deb (2000) along with temporarily treating an individual as feasible if its total absolute constraint violation is less than μ. According to the feasibility approach, (a) a feasible solution is preferred over an infeasible solution, (b) among two feasible solutions, the one with better objective function value is preferred, and (c) among two infeasible solutions, the one with smaller TAV is chosen. The emphasis in this approach is on feasibility, which may affect exploration. The adaptive relaxation and feasibility approach will improve global search by gradually forcing the population towards the feasible region. This is demonstrated below for some optimization problems with equality and/or inequality constraints.

6.3.1 Illustration of the Proposed Constraint Handling Method

The effectiveness of the proposed constraint handling method and its comparison with feasibility approach alone (i.e., without constraint relaxation) are shown in Fig. 6.1 for G11 problem in Table 6.1, which has 2 variables and one equality constraint. Plots a1, a2, a3 and a4 in this figure depict the distribution of population using feasibility approach alone at initialization, 20th, 50th and 300th generation respectively. Plots b1, b2, b3 and b4 in Fig. 1 show the distribution of population using the proposed constraint handling method at initialization, 20th, 50th and 300th generation respectively; in these plots, the relaxed constraint is shown by the red, dotted curves. The green triangles in Fig. 6.1 represent the population at the stated generation, and the red square is the global optimum of the problem.

In the initialization stage, a solution/individual in plot a1 in Fig. 6.1 is feasible only if it is on the equality constraint indicated by the blue line whereas individuals in the population will be treated as feasible if they are within the region between the two red, dotted curves in plot b1 in Fig. 6.1. Thus, it is clear that the feasible region is more with the constraint relaxing method. The selection criterion based on the feasibility approach of Deb (2000) forces the population towards the equality constraint with generations as shown in plots a2 and b2 of Fig. 6.1. In plot a2, the entire population moves to the equality constraint line (i.e., becomes feasible) but the individuals are far from the global optimum. However, in plot b2 of Fig. 6.1, with the same selection criterion based on the feasibility approach, the population moves towards the global optimum due to constraint relaxation.

Note the gradual reduction in the constraint relaxation with generations in plots b1 to b4 (Fig. 6.1), which guides the population towards the equality constraint. At the 50th generation, the solution does not improve much without constraint relaxation (plot a3 in Fig. 6.1) but the population generated by the proposed method moves towards the global optimum (plot b3 in Fig. 6.1). When the search progresses to the 300th generation, the population is trapped at the local optimum in case of feasibility approach alone as shown in plot a4 of Fig. 6.1. With the constraint relaxation, the population is converged to the global optimum (plot b4, Fig. 6.1). Therefore, Fig. 6.1 for problem G11 clearly indicates that the proposed constraint relaxation can enhance the global search ability for problems with equality constraints, without introducing any additional parameters.



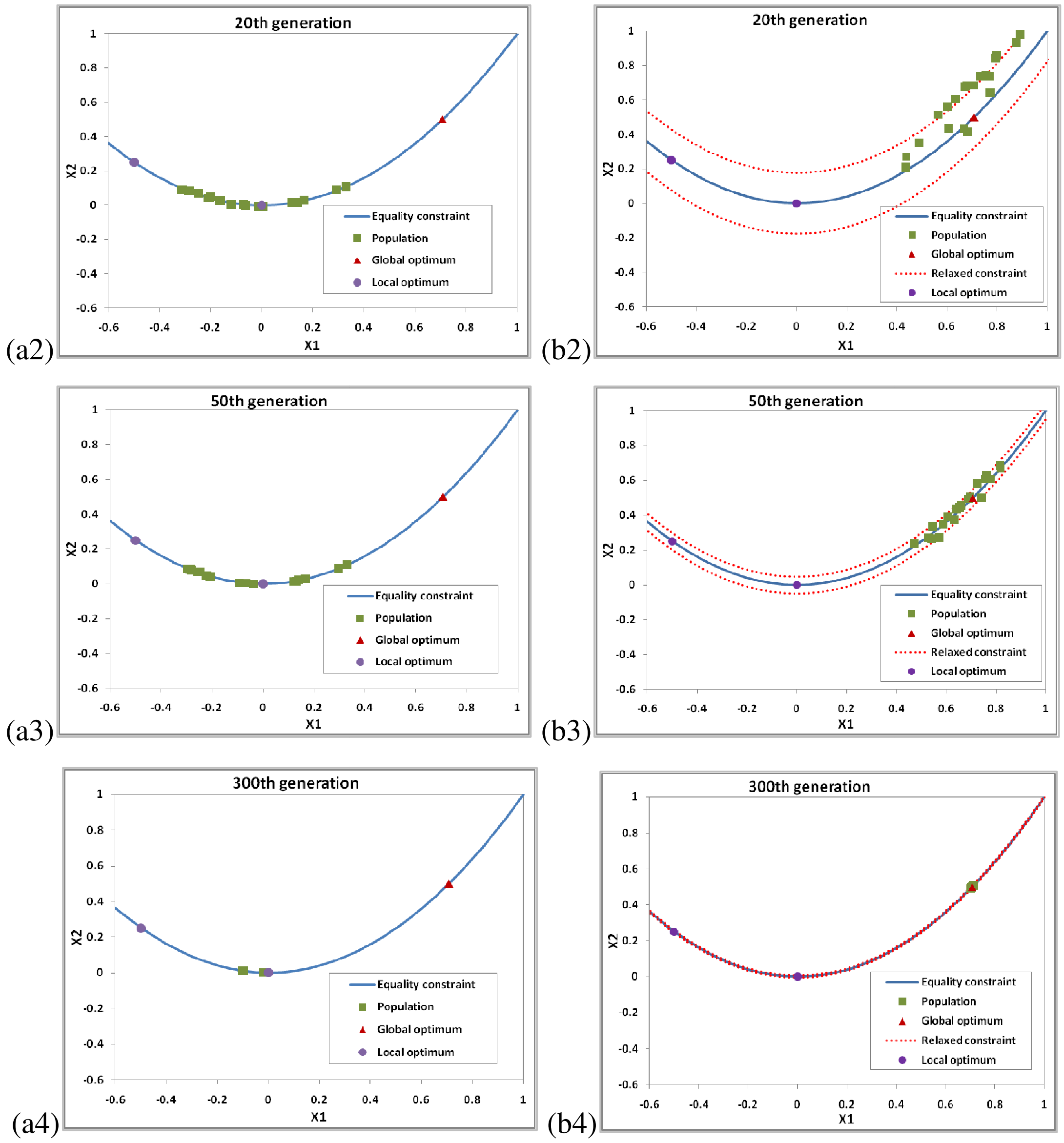


Figure. 6.1. (a1-a4) Distribution of the Population of IDE with Feasibility Approach Alone (i.e., without Constraint Relaxation) at Different Generations, and (b1-b4) Distribution of the Population of IDE with the Adaptive Relaxed Constraint Handling Method at Different Generations: Problem G011

For inequality constraints, the proposed constraint handling strategy is applied for solving problem G06 in Table 1, which contains inequality constraints only. For this inequality problem, it is clear from the contour plots in Fig. 6.2 that the proposed relaxation method with feasibility approach can guide the population to approach the feasible region of the search space from different directions. Compared with the results for the equality problem shown in Fig. 6.1, convergence of the population towards the feasible region is faster for the inequality problem. We can see that at 20th generation, the population has moved to the feasible region as shown in Figs. 6.2(b) and (c), and it has converged to the global optimum at 50th generation as shown in Fig. 6.2(d). The comparison of the proposed relaxation method with feasibility approach alone also shows that the proposed method obtains better result. Thus, the proposed relaxation method can handle inequality constraints efficiently.

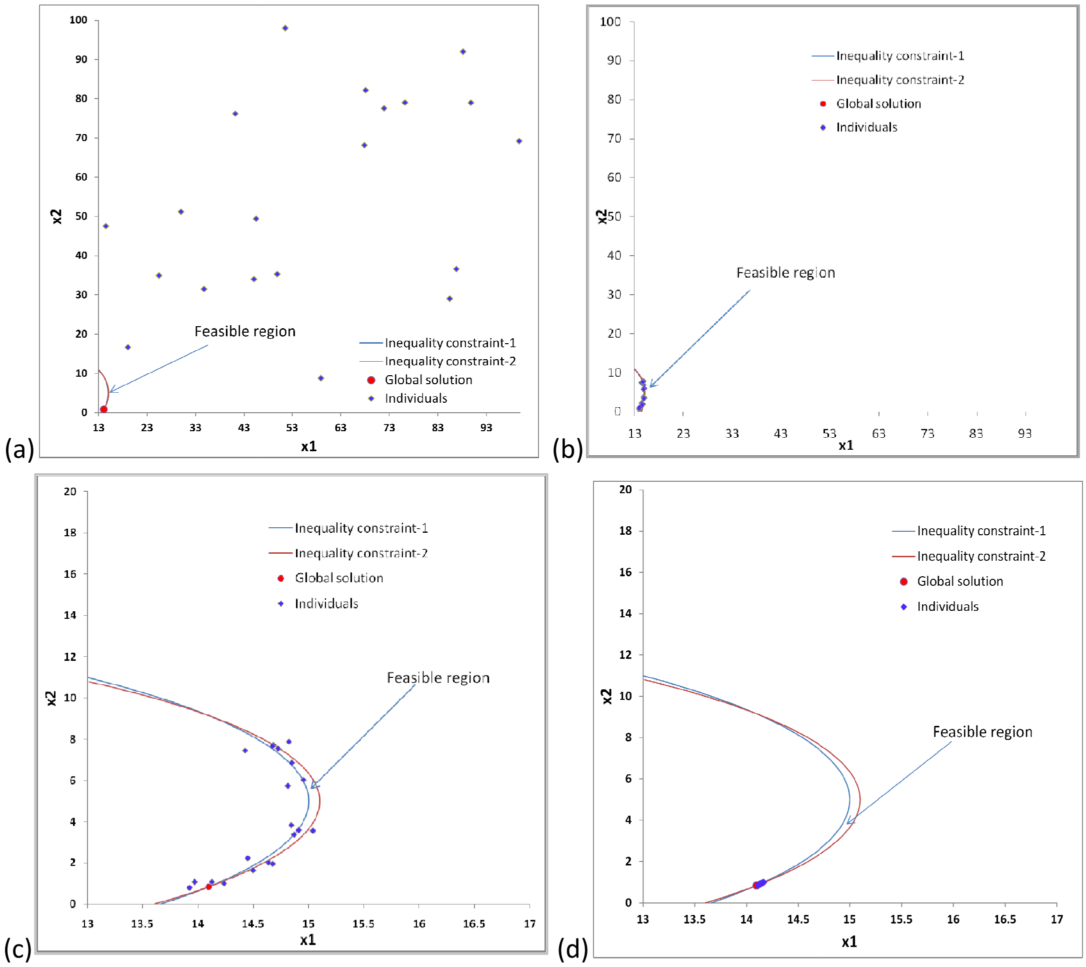


Figure. 6.2. Contour Plots of Problem G06. (a) Search Space and Population at Generation 1. (b) Search Space and Population at Generation 20. (c) Enlarged Plot of (b). (d) Enlarged Search Space and Population at Generation 50.

In order to illustrate the effectiveness of the proposed constraint handling method for both equality and inequality constraints, problem G05 in Table 6.1 is next used. This problem has 4 variables, 2 inequality constraints and 3 equality constraints. The convergence profiles of IDE with the proposed constraint handling method and with the feasibility approach alone (without constraint relaxation), are shown in Fig. 6.3. In the initial generations, feasibility approach has faster convergence compared to the proposed method. However, the best solution is fluctuating because the selection operation chooses the solution with less constraint violation or feasible solution even though it has worse objective value. After 1200 NFE, the best solution obtained with the feasibility approach does not change any more. This is because the feasible solution is found and the population is trapped at a local optimum. On the other hand, the proposed method can continuously improve the solution until the global optimum is reached. The gradual reduction of relaxation (μ) with NFE can be seen in Fig. 6.3, and this is very important in order to avoid trapping at a local optimum. It is interesting that the reducing trend of μ is very similar to that of the objective function value. This is consistent with Eq. (6.5) as more feasible individuals in population results in faster reduction of μ and faster convergence. The convergence profiles of Fig. 6.3 clearly indicate that the proposed constraint handling method can handle both equality and inequality constraints better than the feasibility approach alone.

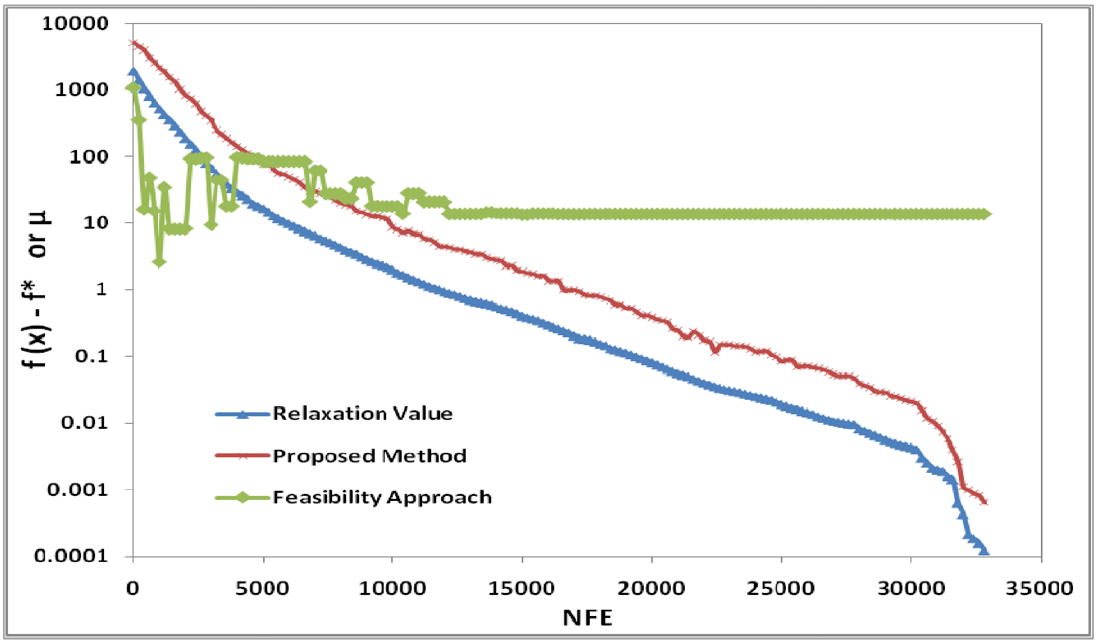


Figure. 6.3. Convergence Profiles of IDE with the Proposed Method and with the Feasibility Approach Alone, and also the Profile of the Relaxation Value (μ) with NFE for Problem G05.

6.3.2 Analysis of Constraint Relaxation Rules

The relaxation (μ) of constraints plays an important role in the proposed constraint handling method. Initially, we have tested Eqs. (6.2) and (6.3), but there are too many parameters to be tuned. Furthermore, the reduction of μ is not based on the progress and performance of the global search, which can vary with the problem characteristics. Therefore, we developed parameter-free rules for reducing μ based on the performance of algorithm. These are Eqs. (6.5), (6.6) and (6.7).

$$\mu(\mathrm{G}+1)=\mu(\mathrm{G})\left(1-\frac{\mathrm{F_F}}{\mathrm{NP}}\right)^2 \qquad (6.6)$$

$$\mu(\mathrm{G}+1)=\begin{cases}\mu(\mathrm{G})(1-\mathrm{F_F}) & 0\le \mathrm{F_F}\le 0.9\\ 0.1\mu(\mathrm{G}) & 0.9<\mathrm{F_F}\end{cases} \qquad (6.7)$$

The performance of the different rules for reducing the relaxation value (μ) and the objective function value for problem G03 (Table 6.1) are shown in Fig. 6.4. It is clear that Eq. (6.7) offers the fastest reduction of μ but solution convergence is very slow. The relaxation based on Eq. (6.6) has faster reduction of μ in the early generations and slower reduction in later stages; however, the solution stagnates due to the initial faster reduction of μ (Fig. 6.4). Eq. (6.5) shows the slowest in reduction of μ but the solution can continuously improve with the generations and finally converge to global optimum. Therefore, Eq. (6.5) for reducing the relaxation value (μ) is very promising as it can gradually guide the search towards the global optimum.

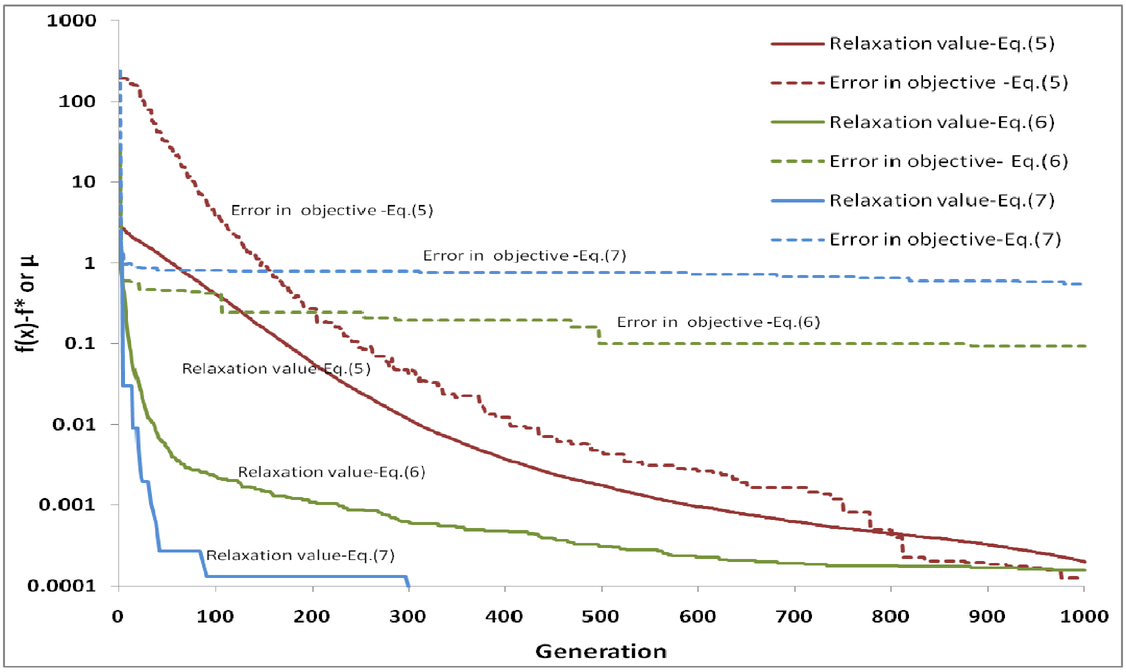


Figure. 6.4. Profiles of Constraint Relaxation Value (μ) and Objective Function Value with Generations, for Solving Problem G03 by C-IDE with Different Relaxation Rules

In order to confirm the robustness of the proposed constraint handling method, Eq. (6.5) is tested for two more problems (G01 and G13) with inequality and equality constraints respectively, and the convergence profiles with generations are presented in

Fig. 6.5. The constraint relaxation value (μ) is reduced to a very small value (Fig. 6.5), which indicates that Eq. (6.5) can force the solutions into the feasible region. Comparing Figs. 6.5(a) and (b), it is clear that the reduction of μ for the inequality constraint problem is faster than that for the equality constraint problem. This is reasonable because the inequality constrained problem usually has bigger feasible region than the equality constrained problem and the reduction of μ is related to the feasibility of the population. Thus, the reduction of μ is not the same in different problems since it is based on the feasibility of the population and consequently on the problem characteristics. Hence, the proposed constraint relaxation rule (Eq. 6.5) is robust and suitable for different problems.

The convergence profile of error (i.e., discrepancy from the global optimum) in the objective function of the best solution at different generations in Fig. 6.5 shows that the proposed constraint handling method can find the global optimum effectively. Further, Fig. 6.5 shows that the solution convergence for inequality constrained problem is faster than that for the equality constraint problem. In summary, the convergence plots (Figs. 6.1-6.5) for different problems with equality, inequality, and both inequality and equality constraints, show that the proposed constraint handling method along with IDE can effectively and robustly handle the constrained problems.

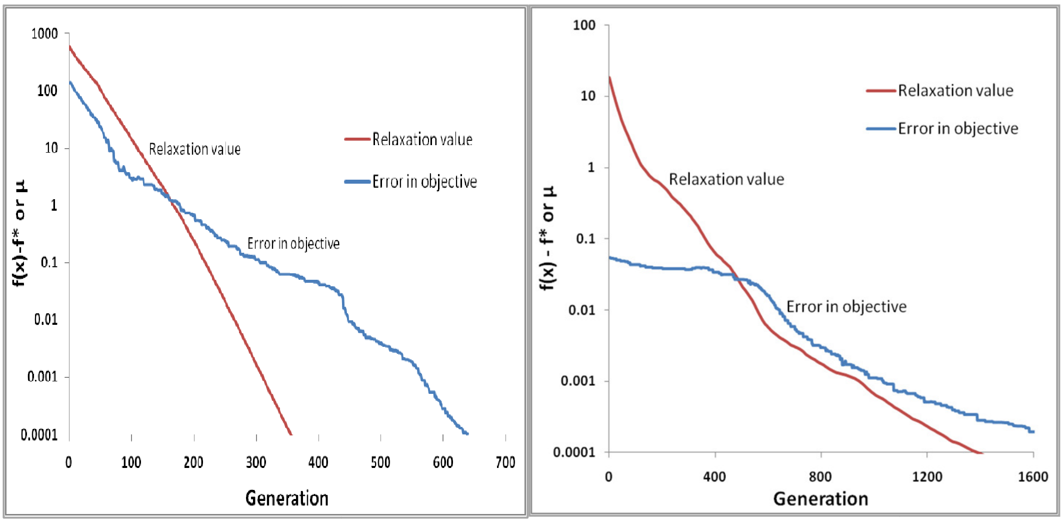


Figure. 6.5. Profiles of Constraint Relaxation Value (μ) and the Objective Function Value for Problem G01 (left plot) and G13 (right plot)

6.4 Description of IDE with the Proposed Constraint Handling Method

The classic DE consists of four steps: initialization, mutation, crossover and selection (Storn & Price, 1997). Mutation and crossover steps are used to generate new individuals. After evaluation of the objective function value, the selection step chooses either the trial or target individual based on the survival of the fittest criterion. Therefore, the individual with better fitness/objective value will proceed to the next generation. The mutation and crossover steps together with the selection step constitute one generation or iteration of the DE algorithm. The generation procedure is repeated until the specified stopping criterion is satisfied.

In IDE, several strategies are incorporated into the classic DE to enhance the algorithmic robustness, reliability and efficiency (Zhang & Rangaiah, 2011). They are: (1) crossover parameter, mutation strategy and parameter are self-adapted according to the learning experience from the previous generations; (2) tabu list and tabu check are used to avoid revisiting the same area, increase the diversity of the population and avoid unnecessary function evaluation in order to enhance global exploration and prevent premature convergence; (3) a novel and effective stopping criterion is used based on the number of rejected points during the generation of a trial point; and (4) a local optimizer is employed. In the present work, the proposed adaptive constraint relaxation technique is included in the IDE for constrained problems.

The flowchart of the C-IDE algorithm is shown in Fig. 6.6. The algorithm begins with the setting of parameter values: population size (NP), learning period (LP), tabu list size (TLS), tabu radius (TR), maximum number of generations (Gmax) and maximum number of rejections (NRmax). The initial population of NP individuals is generated using uniformly distributed random numbers within the search space. The objective function and constraints of each individual are evaluated. The median value (μ) of the total absolute violation of constraints (TAV) of all individuals in the initial population is calculated. The feasibility of each individual in the population is checked as follows. If the TAV (and not violation of each constraint) of an individual is less than μ, then it is temporarily treated as a feasible solution; else, it is taken as an infeasible solution. The best individual in the initial population based on the feasibility approach (Deb, 2000) is selected. The evaluated individuals are then sent to the tabu list, which is used to prevent the algorithm from searching the area close to these individuals.

The concept of tabu list, from the tabu search, keeps track of the previously evaluated points in the search space to avoid the revisit of the already searched regions. During the generation process, when a trial individual is generated, its location in the search space is compared to the previously evaluated points in the tabu list in terms of

Euclidean distance. If the Euclidean distance is smaller than the specified tabu radius (TR), which indicates that the trial individual is close to one of the points in the tabu list, the newly generated trial individual is rejected since it may not bring new information about the search space but increase the number of function evaluations (NFE). The rejected trial individual is replaced by generating a new trial individual until the Euclidean distance between the new trial individual and all the points inside tabu list is greater than TR. This procedure avoids revisiting the already visited regions, makes the individuals in the population more diverse, enhances the exploration of the search space and avoids unnecessary evaluation of the objective function. The objective function is evaluated at the trial individual only if it is away from all the points in the tabu list. After each evaluation, the tabu list is updated dynamically to keep the latest points in the list by replacing the earliest entered point(s). Thus, the new point is added to the list by rejecting the oldest point in the list so that the recently added points are retained. The tabu check will require extra computational effort but this is negligible in application problems where the evaluation of objective function and constraints is computationally intensive.

In each generation, mutation, crossover and selection steps of DE are performed on the population (Fig. 6.6). The probability of kth mutation strategy, P_k and crossover rate for kth mutation strategy, Cr_k are self-adapted based on the performance from the previous generations; the number of mutation strategies used in this study is 4. For the first LP generations, probability of each mutation strategy is 0.25 and mean crossover rate for each strategy, Crm_k is 0.5. For subsequent generations, P_k is updated based on the performance of the strategy in the previous LP generations, and the median of stored Cr_k values is used for Crm_k. During each generation, a strategy for each target individual is selected with probability P_k using stochastic universal sampling method (Baker, 1987). For generating ith trial individual, mutation factor F is calculated based on normal distribution with mean of 0.5 and standard deviation of 0.3, and $Cr_{i,k}$ is based on normal distribution with mean of Crm_k and standard deviation of 0.1. The ith trial individual is then produced according to the assigned mutation strategy, F and $Cr_{i,k}$ through mutation and crossover steps of DE (Price et al., 2005).

A boundary violation check is performed to make sure the decision variables of the trial individual generated satisfy their respective bounds (Fig. 6.6). If any bound is violated, a new trial individual is generated randomly within the bounds. The trial individual is then compared with the individuals/points in the tabu list. If it is near to any point in the tabu list, the trial individual is rejected and another point is generated through the mutation and crossover operations. If the number of rejections, NR is greater than NR_{max} for the same trial individual, then it indicates the algorithm has converged to either the approximate global optimum or trapped at a local optimum, and so the best solution found so far is unlikely to improve significantly in subsequent generations. Therefore, the evolution process is terminated, and the local optimizer is started from the best point found so far.

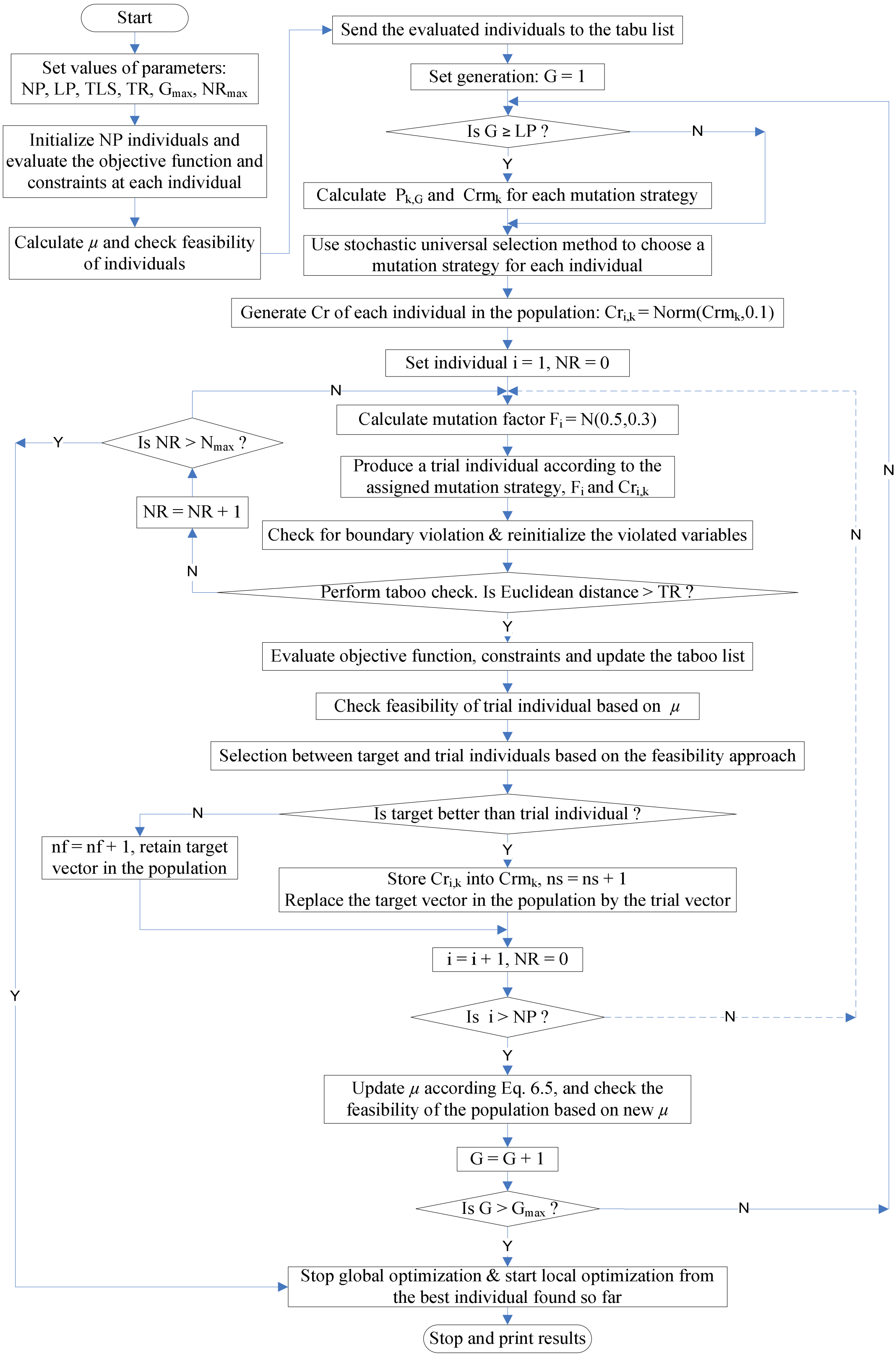

Start
Set values of parameters: NP, LP, TLS, TR, G_{max}, NR_{max}
Initialize NP individuals and evaluate the objective function and constraints at each individual
Calculate μ and check feasibility of individuals
Send the evaluated individuals to the tabu list
Set generation: G = 1
Is G ≥ LP ?
N
Y
Calculate $P_{k,G}$ and Crm_k for each mutation strategy
Use stochastic universal selection method to choose a mutation strategy for each individual
Generate Cr of each individual in the population: $Cr_{i,k} = Norm(Crm_k, 0.1)$
Set individual i = 1, NR = 0
N
Is NR > N_{max} ?
Y
Calculate mutation factor $F_i = N(0.5,0.3)$
Produce a trial individual according to the assigned mutation strategy, F_i and $Cr_{i,k}$
NR = NR + 1
Check for boundary violation & reinitialize the violated variables
N
N
N
Perform taboo check. Is Euclidean distance > TR ?
Y
Evaluate objective function, constraints and update the taboo list
Check feasibility of trial individual based on μ
Selection between target and trial individuals based on the feasibility approach
N
Is target better than trial individual ?
nf = nf + 1, retain target vector in the population
Y
Store $Cr_{i,k}$ into Crm_k, ns = ns + 1
Replace the target vector in the population by the trial vector
i = i + 1, NR = 0
Y
Is i > NP ?
N
Y
Update μ according Eq. 6.5, and check the feasibility of the population based on new μ
G = G + 1
Is G > G_{max} ?
N
Y
Stop global optimization & start local optimization from the best individual found so far
Stop and print results

Figure. 6.6. Flowchart of C-IDE for Constrained Problems

If NR is not greater than NR_{max}, then the algorithm will continue the generation process (Fig. 6.6). After evaluating the objective function and constraints of the trial individual produced, the tabu list is updated and feasibility check is performed according to the constraints relaxed by μ (i.e., feasible if TAV < μ). Then, the selection step between the target and trial individuals is performed according to the feasibility approach based on the objective function value, (temporary) feasibility of the individual and TAV. If the trial individual is selected, it replaces the target individual in the population immediately, and $Cr_{i,k}$ is stored and the success of the corresponding mutation strategy is updated as $ns_{k,G} = ns_{k,G} + 1$. Otherwise, the target individual remains in the population and the failure of the corresponding mutation strategy is recorded as $nf_{k,G} = nf_{k,G} + 1$. Then, NR is reset to 0 for generating the trial individual for the next target individual until all NP target individuals are covered.

After each generation, μ is updated according to Eq. (6.5), and the feasibility of the population based on the new μ is checked. The updating of P_{kG}, calculation of Cr_{mk}, mutation, crossover and selection operations are repeated for the next generation until the maximum number of generations, G_{max}. Then, the best point obtained over all generations is refined using the local optimizer. Solver tool in Excel is used as the local optimizer in order to find the precise solution. It is a powerful gradient-based optimizer, readily available in MS Excel. It can solve different types of the linear and nonlinear optimization problems with both equality and inequality constraints. Generalized reduced gradient (GRG) method, used for solving nonlinear problems in Solver tool, is an efficient local optimizer and uses finite difference approximation for numerical derivatives of the objective function. It works by first evaluating the objective/constraint functions and their partial derivatives at the initial values of the decision vector, and then iteratively searches for a better solution using a search direction suggested by derivatives. To determine a search direction, Solver uses the quasi-Newton or conjugate gradient method. The user is not required to provide the partial derivatives with respect to decision variables; instead, forward or central difference approximation, as per user preference, is used in Solver (Stokes & Plummer, 2004).

6.5 Numerical Experiments, Results and Discussion

The proposed constraint handling method with IDE, C-IDE, is implemented in MS Excel environment using Visual Basic Application (VBA) language because of its ready availability and use by researchers and practitioners in diverse fields. A user-friendly interface is developed for coding the objective function, constraints and calling the optimization algorithm by anyone familiar with a spreadsheet. Performance evaluation of the proposed constraint handling method with IDE using this program in MS Excel, and its comparison with the recent methods and results in the literature are presented in this section. All computations were performed on Dell Optiplex 750 with Intel Core 2 (Duo CPU 2.66 GHz, 3.25 GB RAM), which can complete 594 MFlops (million floating-point operations) for the LINPACK benchmark program (at http://www.netlib.org/) for a matrix of order 500.

6.5.1 Benchmark Problems

The proposed C-IDE is evaluated on 22 benchmark problems with equality and/or inequality constraints, given in CEC 2006 (Liang et al., 2006), for testing and comparing with other constrained global optimization algorithms. Mathematical characteristics of these benchmark problems are summarized in Table 6.1. Note that each of these problems involves one or more constraints.

Table 6.1. Mathematical Characteristics of the Benchmark Problems (Liang et al., 2006)

Problem	Variables	ρ (%)	LI	NI	LE	NE	α	Global Optimum, f*
G01	13	0.0111	9	0	0	0	6	-15.0000
G02	20	99.9971	0	2	0	0	1	-0.8036
G03	10	0.0000	0	0	0	1	1	-1.0005
G04	5	52.1230	0	6	0	0	2	-30665.5387
G05	4	0.0000	2	0	0	3	3	5126.4967
G06	2	0.0066	0	2	0	0	2	-6961.8139
G07	10	0.0003	3	5	0	0	6	24.3062
G08	2	0.8560	0	2	0	0	0	-0.0958
G09	7	0.5121	0	4	0	0	2	680.6301
G10	8	0.0010	3	3	0	0	6	7049.2480
G11	2	0.0000	0	0	0	1	1	0.7499
G12	3	4.7713	0	1	0	0	0	-1.0000
G13	5	0.0000	0	0	0	3	3	0.0539
G14	10	0.0000	0	0	3	0	3	-47.7649
G15	3	0.0000	0	0	1	1	2	961.7150
G16	5	0.0204	4	34	0	0	4	-1.9052
G17	6	0.0000	0	0	0	4	4	8853.5397
G18	9	0.0000	0	13	0	0	6	-0.8660
G19	15	33.4761	0	5	0	0	0	32.6556
G20	7	0.0000	0	1	0	5	6	193.7245
G21	9	0.0000	0	2	3	1	6	-400.0551
G22	2	79.6556	0	2	0	0	2	-5.5080

Notes: ρ is the estimated ratio between the feasible region and the search space. LI and NI are respectively the number of linear and nonlinear inequality constraints, LE and NE are respectively the number of linear and nonlinear equality constraints, and α is the number of active constraints at the global optimum.

6.5.2 Parameter Settings and Testing Details

The following parameters were used throughout this study: population size, NP = 20, tabu list size = 20 and tabu radius = 0.001D, which are the same as in Srinivas and Rangaiah (2007), And learning period, LP = 20. Stopping criterion is the satisfaction of either the maximum number of rejections, NR_{max} = 20 or the maximum number of function evaluations, NFE_{max} = 50,000; the latter is included to avoid indefinite iterations. The same parameter values were used for all benchmark problems and application problems in order to test the robustness of the algorithm for practical applications.

Since C-IDE is a stochastic optimization algorithm, random numbers may affect convergence speed and reliability of finding the global optimum. Hence, 25 independent runs, each time starting from a different random number seed, were performed on each benchmark problem. A successful run means that the algorithm can achieve the objective function value very close to the known global optimum value, f*. Here, a run is considered to be successful if the objective function value found is ≤ [f* + 0.0001] for all benchmark functions. Success rate (SR) is the percentage of number of successful runs out of 25 trials. The SR, mean and standard deviation of the error, $| f^{best} - f^*|$, and average number of objective function/constraints evaluations (NFE) in the 25 trials are reported in Table 2; here, f^{best} is the best value found in each run. The solution quality, NFE and SR are compared and discussed in the following section.

6.5.3 Results and Discussion

6.5.3.1 Comparison with εDE, SaDE and LEDE

Table 6.2 summarizes the results obtained by C-IDE and C-IDE without local search, both with the proposed constraint handling method; recall that C-IDE (Fig. 6.6) includes local optimization after the completion of global search. It also compares their performance with εDE (Takahama & Sakai, 2006), LEDE (Ali & Kajee-Bagdadi, 2009) and SaDE (Huang et al., 2006) for all benchmark problems in Table 6.1. The εDE and SaDE have shown good performance on the 2006 CEC benchmark functions (Liang & Suganthan, 2006). LEDE (Ali & Kajee-Bagdadi, 2009) is a recent constrained optimization method, which has been tested on many benchmark functions with both equality and inequality constraints. In order to check the performance of the proposed constraint handling method, we first consider C-IDE without local search and using NFEmax = 50,000 as the stopping criterion. From Table 6.2, it is obvious that C-IDE without local search can achieve the best mean and standard deviation of the objective function value for many of the benchmark problems tested. The smallest error mean, standard deviation and NFE of the average of the 22 benchmark functions indicates that the solution obtained by the proposed algorithm is very close to the global solution. From this, we can conclude that the proposed constraint handling method is more effective than SaDE, εDE and LEDE to solve the constrained optimization problems. Note that SaDE and εDE use gradient based local search during the generations.

Compared to SaDE, εDE, LEDE and C-IDE without local search, it is clear that C-IDE with the NRmax stopping criterion and local search obtained the best mean and standard deviation using the least NFE. For problems G01 and G02, the reduction in NFE is about 75%. For problem G06, the reduction in NFE is more than 95%. This may due to fewer decision variables, and so C-IDE is able to converge faster. Similar reduction can also be seen for problems G08, G11 and G22. The total NFE reduction for all problems tested is about 61%.

Table 6.2. Comparison of DE-based Algorithms for Benchmark Problems

Problem & Optimal value		εDE (Takahama & Sakai, 2006)	SaDE (Huang et al., 2006)	LEDE (Ali & Kajee-Bagdadi, 2009)	C-IDE w/o Local Search	C-IDE
G01 -15.0000	Mean	8.44E-04	2.94E-10	7.95E-04	4.98E-04	3.15E-08
	std	2.52E-04	4.95E-10	3.40E-04	2.27E-03	1.67E-07
	NFE	50000	50000	50000	50000	16529
G02 -0.8036	Mean	6.94E-02	2.18E-02	2.68E-01	3.31E-02	2.71E-02
	std	2.77E-02	8.27E-03	7.20E-02	3.83E-02	2.83E-02
	NFE	50000	50000	50000	50000	19049
G03 -1.0005	Mean	5.44E-02	6.24E-02	2.38E-01	5.04E-04	7.50E-07
	std	2.29E-02	1.54E-01	3.90E-02	3.30E-05	1.32E-06
	NFE	50000	50000	50000	50000	25493
G04 -30665.5387	Mean	4.92E-11	2.98E-07	1.44E-03	2.30E-05	6.64E-07
	std	3.40E-11	2.51E-08	5.20E-05	2.42E-05	3.49E-06
	NFE	50000	50000	50000	50000	11728
G05 5126.4967	Mean	3.22E-02	6.40E-02	1.10E-01	1.47E-03	9.60E-06
	std	3.42E-02	5.43E-04	4.10E-03	7.22E-04	3.21E-09
	NFE	50000	50000	50000	50000	21363
G06 -6961.8139	Mean	1.18E-11	4.55E-11	6.70E-07	0.0E+00	2.57E-12
	std	0.00E+00	0.00E+00	2.70E-07	0.0E+00	2.43E-11
	NFE	50000	50000	50000	50000	1523
G07 24.3062	Mean	3.88E-03	4.93E-03	2.22E-03	1.71E-03	5.23E-06
	std	1.60E-03	5.93E-03	2.10E-04	2.66E-03	3.17E-06
	NFE	50000	50000	50000	50000	18429
G08 -0.0958	Mean	4.16E-17	8.20E-11	2.35E-06	1.93E-10	3.47E-16
	std	1.23E-32	6.05E-18	7.00E-07	3.28E-11	2.32E-16
	NFE	50000	50000	50000	50000	2668
G09 680.6301	Mean	2.41E-07	4.45E-06	1.21E-07	7.53E-08	7.53E-16
	std	1.45E-11	8.79E-06	3.50E-08	2.71E-07	6.26E-15
	NFE	50000	50000	50000	50000	35003
G10 7049.2480	Mean	1.67E+00	3.98E-01	6.20E+00	1.48E+00	1.08E-05
	std	1.18E+00	7.43E-01	3.00E+00	1.13E+00	1.17E-06
	NFE	50000	50000	50000	50000	37620
G11 0.7499	Mean	5.66E-06	9.40E-06	0.0E+00	0.0E+00	2.38E-11
	std	4.37E-06	2.75E-05	0.0E+00	0.0E+00	2.13E-10
	NFE	50000	50000	50000	50000	6609
G12 -1.0000	Mean	0.00E+00	0.00E+00	3.57E-05	0.00E+00	0.00E+00
	std	0.00E+00	0.00E+00	6.60E-06	0.00E+00	0.00E+00
	NFE	50000	50000	50000	50000	13905
G13 0.0539	Mean	7.41E-05	1.08E-01	1.63E-01	1.35E-05	1.54E-11
	std	6.99E-05	1.76E-01	1.4E-01	4.90E-05	2.47E-10
	NFE	50000	50000	50000	50000	19180

Table 6.2 Continuous

G14 -47.7649	Mean	3.92E+00	1.61E-04	NA	2.06E+00	2.25E-09
	std	1.65E+00	1.10E-04	NA	1.64E+00	6.43E-09
	NFE	50000	50000	NA	50000	34825
G15 961.7150	Mean	1.33E+00	6.34E-05	NA	1.28E-07	3.28E-10
	std	2.21E+00	6.01E-05	NA	1.47E-09	1.75E-10
	NFE	50000	50000	NA	50000	11706
G16 -1.9052	Mean	5.03E-15	6.53E-11	NA	6.96E-10	3.25E-11
	std	8.16E-16	6.76E-14	NA	1.89E-09	9.45E-10
	NFE	50000	50000	NA	50000	9765
G17 8853.5397	Mean	4.37E+01	7.27E+01	NA	3.75E+01	1.55E+01
	std	3.70E+01	6.85E+00	NA	3.37E+01	6.17E+00
	NFE	50000	50000	NA	50000	43369
G18 -0.8660	Mean	3.98E-04	1.53E-02	NA	1.23E-03	2.73E-07
	std	2.22E-04	5.30E-02	NA	2.55E-04	2.10E-07
	NFE	50000	50000	NA	50000	31478
G19 32.6556	Mean	1.18E+01	6.07E-03	NA	2.40E-01	6.69E-07
	std	2.96E+00	2.24E-02	NA	5.17E-02	1.80E-06
	NFE	50000	50000	NA	50000	19111
G20 193.7245	Mean	3.04E+01	6.03E-02	NA	1.86E+01	1.05E-02
	std	4.25E+01	3.81E-03	NA	4.03E+01	1.08E-02
	NFE	50000	50000	NA	50000	23631
G21 -400.0551	Mean	3.87E+02	5.02E-02	NA	1.10E+03	2.56E-13
	std	1.45E+02	1.36E-02	NA	1.99E+02	3.25E-13
	NFE	50000	50000	NA	50000	41671
G22 -5.5080	Mean	5.77E-14	4.64E-12	NA	5.62E-10	6.57E-09
	std	2.52E-29	0.00E+00	NA	5.51E-09	2.08E-08
	NFE	50000	50000	NA	50000	1739
Average	Mean	2.29E+01	3.50E+00	7.19E-01	1.57E+00	7.06E-01
	std	1.11E+01	3.82E-01	3.11E-01	2.97E+00	2.82E-01
	NFE	5.00E+04	5.00E+04	5.00E+04	5.00E+04	2.03E+04

Notes: C-IDE and C-IDE without local search in the last two columns include the proposed constraint handling method; NA - Data are not available; std – Standard deviation.

C-IDE obtained good reliability with 100% SR for all the benchmark functions except G02, G17and G20 for which SR is 60%, 92% and 88% respectively. The lower SR of G02 and G20 is because there are some local optima that are very near to the global optimum. For G02, the global solution is -0.8036, but there are many local solutions such as -0.79466, -0.79261 and -0.78527. For G20, the global solution is 193.7245 at (193.7245, 0, 17.3192, 100.0479, 6.6845, 5.9917, 6.2145), but there is a local optimum of 193.7581 at (193.7581, 0, 17.3192, 100.0479, 6.6845, 5.9917, 6.2145). This caused the gradient-based local optimization technique (GRG) to converge prematurely in a few trials. Even through SR for G17 problem by C-IDE is only 92%, the mean and standard deviation are better than other algorithms. Furthermore, the feasibility of this problem is 100%. From detailed results, we noticed that the few failures are mainly due to the wrong direction taken by the local search. The solution obtained after the global search is very near to the global solution, but the local search converges to a local solution that is far from the global solution. Overall, the proposed constraint handling method, C-IDE, which integrates the global and local searches together with switching from global to local search based on maximum number of rejections, is very reliable and efficient for solving general constrained problems.

6.5.3.2 Comparison with Other Methods

The superior performance of the C-IDE over DE-based algorithms (SaDE, εDE and LEDE) is demonstrated in Table 6.2. In this sub-section, the performance of the proposed constraint handling method with IDE is compared with that of recent non-DE based constrained global optimization algorithms. Leguizamon and Coello Coello (2009) proposed a boundary search with ant colony metaphor algorithm (ANT-β) for constrained optimization. ANT-β focuses the search on the boundary region between feasible and infeasible search space. The penalty function method is used as a complementary mechanism for handling the constraints. The proposed method has been tested on a number of benchmark functions and compared with other methods (Leguizamon & Coello Coello, 2009). Liu et al. (2010) used a hybrid PSO-DE algorithm with feasibility approach to optimize the objective function and constraints simultaneously with two populations of same size. Mani and Patvardhan (2009) proposed a hybrid constraint handling method with evolutionary algorithm (EA) to solve the constrained optimization problems. Two constraint handling methods (penalty function and feasibility approach) with two populations are cooperated in EA. This approach overcomes the drawbacks of both penalty function and feasibility approach, and utilizes their strengths to handle constraints. Results on the benchmark problems demonstrate its efficacy.

The performance results of the C-IDE are compared in Table 6.3, with those of ANT-β, PSO-DE and EA taken from (Leguizamon & Coello Coello, 2009; Liu et al., 2010; Mani & Patvardhan, 2009) respectively. These algorithms are implemented differently with different accuracies, termination criteria and/or parameter values. Hence, it would be unfair and difficult to draw any conclusions from the direct comparisons of these results. However, we wish to highlight and comment on certain aspects, e.g. the quality of solution obtained and associated effort required. Table 6.3 shows that C-IDE and ANT-β have been used to solve 22 and 17 problems whereas PSO-DE and EA were tested on only 11 and 10 problems respectively. The solution quality by them is indicated by the best, mean and worst of the objective function values found (in 100 runs of PSO-DE, 30 runs of EA and ANT-β) for each problem. For G01, G03, G04, G11 and G22, C-IDE and ANT-β obtained equally good results. For the remaining 12 problems, C-IDE obtained much better results compared to ANT-β. Among the 11 problems tested by the PSO-DE, solutions obtained for 10 problems are as good as those of C-IDE. For G02, C-IDE obtained better solutions compared to PSO-DE. Compared with EA, C-IDE found better solutions for 6 out of 10 problems. ANT-β, PSO-DE and EA were terminated after 1,500,000, 140,100 and 200,000 NFE (Leguizamon & Coello Coello, 2009; Liu et al., 2010; Mani & Patvardhan, 2009). On the other hand, maximum NFE used by the C-IDE is 43,369 for G17 (see Table 6.2). Thus, C-IDE with the stopping criterion on NRmax stops the search efficiently and reliably for constrained problems, even without knowing the optimum solution in advance.

Table 6.3. Comparison of Objective Function Value Found by C-IDE, ANT-*β*, PSO-DE and EA on Benchmark Problems in (Liang et al., 2006)

Probs	C-IDE			ANT-β			PSO-DE			EA		
	Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
G01	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000	-15.000
G02	-0.804	-0.795	-0.754	-0.804	-0.803	-0.793	-0.804	-0.757	-0.637	NA	NA	NA
G03	-1.001	-1.001	-1.001	-1.000	-1.000	-1.000	-1.001	-1.001	-1.001	NA	NA	NA
G04	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54	-30665.54
G05	5126.497	5126.497	5126.497	5126.500	5138.370	5132.140	NA	NA	NA	5126.498	5127.235	5135.928
G06	-6961.814	-6961.814	-6961.814	-6961.810	-6961.740	-6961.710	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814
G07	24.306	24.306	24.306	24.370	24.640	24.920	24.306	24.306	24.306	24.319	24.410	24.541
G08	-0.096	-0.096	-0.096	NA	NA	NA	-0.096	-0.096	-0.096	-0.096	-0.096	-0.096
G09	680.630	680.630	680.630	680.630	680.670	680.720	680.630	680.630	680.630	680.630	680.646	680.667
G10	7049.248	7049.248	7049.248	7052.300	7199.010	7493.150	7049.248	7049.248	7049.248	7049.424	7075.022	7111.849
G11	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
G12	-1.000	-1.000	-1.000	NA	NA	NA	-1.000	-1.000	-1.000	NA	NA	NA
G13	0.054	0.054	0.054	0.054	0.055	0.055	NA	NA	NA	0.054	0.057	0.076
G14	-47.764	-47.764	-47.764	-47.760	-47.652	-46.724	NA	NA	NA	NA	NA	NA
G15	961.715	961.715	961.715	961.715	961.716	961.717	NA	NA	NA	NA	NA	NA
G16	-1.905	-1.905	-1.905	NA	NA	NA	NA	NA	NA	NA	NA	NA
G17	8853.540	8859.711	8927.598	8855.819	8937.446	8952.621	NA	NA	NA	NA	NA	NA
G18	-0.866	-0.866	-0.866	NA	NA	NA	NA	NA	NA	NA	NA	NA
G19	32.656	32.656	32.656	NA	NA	NA	NA	NA	NA	NA	NA	NA
G20	193.725	196.834	202.862	193.783	194.345	202.068	NA	NA	NA	NA	NA	NA
G21	-400.055	-400.055	-400.055	-399.985	-249.008	-28.448	NA	NA	NA	NA	NA	NA
G22	-5.508	-5.508	-5.508	-5.508	-5.508	-5.508	NA	NA	NA	NA	NA	NA

Note: NA - Data are not available.

6.6 Application to Chemical Engineering Optimization Problems

In this section, C-IDE is tested on 10 chemical engineering optimization problems taken from the literature (Babu & Angira, 2006; Kheawhom, 2010; Lee et al., 2010; Luo et al., 2007; Pintarič & Kravanja, 2006; Srinivas & Rangaiah, 2007). These examples are carefully chosen so that they have equality constraints in addition to inequality constraints, multiple minima and/or narrow feasible region. Mathematical characteristics of the selected application problems are summarized in Table 6.4.

Problems 1, 5 and 7 are for the design of heat exchanger networks, all of which have active constraints at the global minimum. The objective function is to minimize the cost subject to the approach temperature, heat and energy balance equality constraints. Problem 2 is a reactor network design problem that involves the design of a sequence of two reactors. It is difficult to solve because the local minimum value (-0.3881) is very close to the global minimum value (-0.3888) of this problem. Problem 3 is the design of an insulated tank problem, and Problem 4 is a pooling problem. Problem 6 is on the design of a three-stage process system with recycle. Problem 8 is the separation network synthesis problem, and problem 9 is the alkylation process optimization. Problem 10 is the optimization of the William-Otto process (Williams & Otto, 1960), which is representative of chemical processes and has been used in many studies.

Table 6.4. Mathematical Characteristics of the Application Problems

Problem	No. of Variables	No. of Equality Constraints	No. of Inequality Constraints	Global Optimum (*f**)
P1	8	3	3	7049.248
P2	6	4	1	-0.3888
P3	4	2	0	5194.866
P4	10	5	2	-400.0
P5	3	2	0	189.312
P6	6	3	3	-13.402
P7	12	11	0	36162.989
P8	22	16	0	1.864
P9	10	7	0	1162.027
P10	10	6	0	Refer Table 7

6.6.1 Result and Discussion

Each of the 10 problems listed in Table 6.4 is solved 100 times, each time with a different random number seed. The success rate (SR), which indicates the reliability of the algorithm, and NFE are reported. Firstly, C-IDE is tested on P1 to P6 in Table 6.4, which were solved by Srinivas and Rangaiah (2007) by eliminating the equality constraints. The results by C-IDE with and without eliminating the equality constraints of the original problems are compared with DETL from Srinivas and Rangaiah (2007) in Table 6.5. Results obtained to solve the reformulated problems (by eliminating equality constraints) are indicated by an asterisk (i.e., DETL* and C-IDE*) in this table. SR results show that C-IDE* has better reliability compared to DETL*. It can solve all the six problems with 100% SR. This indicates that C-IDE could handle the problems without equality constraints efficiently. C-IDE* has also comparable convergence speed with DETL*, as indicated by NFE in Table 6.5.

In order to demonstrate the effectiveness of the proposed constraint handling method, especially for equality constraints, C-IDE is used to solve the original problems P1 to P6 without eliminating equality constraints. Results in Table 6.5 show that C-IDE is able to solve these problems with 100% SR but it requires more NFE compared to that for solving the re-formulated problems. Total NFE required for C-IDE is almost double the NFE required by C-IDE* for the problems in Table 6.5. This confirms that problems with equality constraints are more difficult to solve by stochastic methods compared to those without equality constraints. This is because of the tiny feasible area of the equality constraints. So, where possible, it is better to eliminate equality constraints in the optimization problems for efficient solution by stochastic methods.

Table 6.5. Performance of C-IDE and DETL for Problems P1 to P6

Problem	DETL*		C-IDE*		C-IDE	
Number	SR	NFE	SR	NFE	SR	NFE
P1	95	11999	100	15632	100	15846
P2	98	1468	100	3975	100	11604
P3	96	1495	100	858	100	7930
P4	100	4421	100	7475	100	19772
P5	100	418	100	328	100	3648
P6	100	2912	100	3773	100	5441

* The equality constraints in the problems are eliminated, and the transformed problems without equality constraints are solved by the stochastic method.

We now compare the performance of C-IDE, with the recent DE-based constrained global optimization algorithm proposed by Kheawhom (2010), who employed Newton's method to solve the equality constraint equations in the problems, in Table 6.6. From this, it is obvious that both C-IDE and the algorithm of Kheawhom (2010) have very high and comparable reliability (i.e., SR). C-IDE is significantly faster than the algorithm of Kheawhom (2010), as indicated by NFE in Table 6.6. For P7, C-

IDE needs 11524 NFE to obtain 95% SR; although SR is slightly lower, it requires significantly fewer NFE compared to the algorithm of Kheawhom (2010).

Table 6.6. Comparison of Results by C-IDE with Those of Kheawhom (2010)

Problem	(Kheawhom, 2010)		C-IDE	
Number	SR	NFE	SR	NFE
P1	100	380000	100	15632
P2	100	10500	100	11604
P7	100	864000	95	11524
P8	100	954500	100	16338

Problem 9 is an alkylation process important in petroleum refining. Rangaiah (1985) studied two different formulations of this problem: (a) original one with 7 equality and no inequality constraints, and (b) reduced one with 14 inequality and no equality constraints. Babu and Angira (2006) studied the reduced problem using DE and modified DE (MDE), both with penalty method to handle the constraints. On the other hand, C-IDE is used to solve both the reduced and original problems. The results show that, for the reduced problem with no equality constraints, DE, MDE and C-IDE can obtain 100% SR but C-IDE requires only 11,925 NFE compared to 100,126 and 92,287 NFE required by DE and MDE respectively. Further, for the more difficult original problem with equality constraints, C-IDE obtained 100% SR using 33,320 NFE. These results indicate that the proposed constraint handling technique, C-IDE is very effective for solving the constrained optimization problems.

The complete model of WO process consists of 82 variables and 78 equality constraints (Lee et al., 2010). The 82 variables are the reaction rate constants, mass flow rates, reactor temperature, reactor volume, component mass fraction and purge fraction. The equality constraints arise from mass balances on the reactor, cooler, decanter and column. Most of the 78 equality constraints can be easily eliminated, leaving 6 nonlinear equality equations (Lee et al., 2010). For the present study, we chose this model of Lee et al. (2010) with 10 decision variables and 6 equality constraints. Four objectives used for optimizing the WO process by Pintaric and Kravanja (2006) and Lee et al. (2010) are considered here; these are the pay-back period (PBP), profit before taxes (PBT), total annual cost (TAC) and net present worth (NPW).

TAC is the sum of operating cost and annualized capital cost, calculated by dividing the fixed capital investment (FCI) over the lifetime of the project (assumed to be 10 years). FCI is the capital necessary for the installed process equipment with all components needed for complete process operation. PBT is the difference between the annual revenue and TAC, without accounting for taxes. PBP is the time required to recover FCI from the annual cash flows (CF). The CF is the sum of profit after taxes and depreciation, where the tax rate is taken to be 30% per year. The NPW is the present value of all investments and cash flows during the project lifetime, where time value of money is at the expected rate of return (0.12 year-1). Note that the program used by Pintaric and Kravanja (2006) is GAMS/CONOPT (a local optimizer), and Lee et al. (2010) used NSGA-II-aJG (a version of genetic algorithm) with an equation solver to handle the equality equations. Lee et al. (2010) stated that the local optimizer can give the correct optimum only when the initial guesses are close to the optimum solution.

In the present study, the proposed C-IDE is able to handle the equality constraints without using any equation solver, to optimize the WO process for different objectives. Optimal values of the decision variables and objectives are compared with the reported results (Lee et al., 2010; Pintarič & Kravanja, 2006) in Table 6.7. This table clearly shows that the proposed constraint handling method is able to optimize the WO process for all the four different objective functions considered, with SR of 99-100%. The average NFE for the four objective functions are 11936, 11918, 20000 and 19770 (Table 6.7). The computational times for these functions are 50, 51, 84 and 83 seconds respectively. It can be seen that the CPU time has almost linear relationship with NFE. This indicates that NFE is a good indicator for comparison of convergence speed since function evaluation involves extensive computations in application problems, and NFE is independent of the computer and software platform used. Note that VBA is used in the present study because of its ready availability and use by researchers and practitioners in diverse fields. Based on our experience, VBA is about one order of magnitude slower than the equivalent MATLAB code. For optimizing PBP, PBT and NPW, the C-IDE obtained same results compared to those of Lee et al. (2010) and Pintaric and Kravanja (2006). For minimizing TAC, C-IDE obtained slightly better result at 9.043 compared to 9.082 and 9.081 obtained by Lee et al (2010) and Pintaric and Kravanja (2006) respectively. These results show the efficiency and robustness of the C-IDE method for solving constrained problems.

In summary, C-IDE method can efficiently solve the chemical engineering optimization problems with equality and/or inequality constraints. It is more reliable and efficient compared to the recent algorithms (Babu & Angira, 2006; Kheawhom, 2010; Lee et al., 2010; Pintarič & Kravanja, 2006; Srinivas & Rangaiah, 2007). Furthermore, it does not require the global optimum in advance since the stopping criterion can effectively switch the algorithm from global search to local search.

:**Table 6.7.** Results for the Optimization of William-Otto Process for Four Different Objectives

Quantity (units)	**Min. Pay-Back Period (PBP)**			**Max. Profit Before Tax (PBT)**			**Min. Total Annual Cost (TAC)**			**Max. Net Present Worth (NPW)**		
	Pintaric and Kravanja (2006)	Lee et al. (2010)	C-IDE	Pintaric and Kravanja (2006)	Lee et al. (2010)	C-IDE	Pintaric and Kravanja (2006)	Lee et al. (2010)	C-IDE	Pintaric and Kravanja (2006)	Lee et al. (2010)	C-IDE
V(m^3)	0.873	0.871	0.873	6.82	6.80	6.81	7.90	7.89	7.42	3.75	3.71	3.75
T(K)	374	375	375	342	343	343	342	342	345	351	351	351
η	0.100	0.100	0.100	0.113	0.113	0.113	0.102	0.102	0.107	0.109	0.11	0.109
$q_{m,1}{}^{A}$(kg/h)	6123	6121	6123	4957	4956	4957	4808	4809	4937	5239	5247	5239
$q_{m,2}{}^{B}$(kg/h)	13956	13965	13953	11113	11118	11112	10880	10882	10936	11792	11809	11789
FCI (Million US$/yr)	0.925	0.924	0.926	7.22	7.21	7.23	8.37	8.37	7.87	3.97	3.94	3.97
CF (Million US$/yr)	0.876	0.875	0.877	2.42	2.42	2.42	2.52	2.52	2.43	2.00	1.99	2.00
PBP (years)	1.056	1.056	1.056	2.983	2.979	2.988	3.321	3.321	3.239	1.985	1.980	1.985
PBT (Million US$/yr)	1.120	1.118	1.121	2.425	2.425	2.425	2.406	2.404	2.353	2.286	2.286	2.286
TAC (Million US$/yr)	10.680	10.681	10.679	9.117	9.117	9.117	9.081	9.082	9.043	9.315	9.315	9.314
NPW (Million US$/yr)	4.02	4.02	4.03	6.44	6.46	6.45	5.86	5.87	5.88	7.30	7.31	7.31
SR (%)	NA	NA	99	NA	NA	100	NA	NA	100	NA	NA	100
CPU (Seconds)	NA	NA	50	NA	NA	51	NA	NA	84	NA	NA	83

6.7 Conclusions

We have presented an adaptive relaxation of constraints for use with IDE, where the parameters of DE are adapted based on the previous performance, tabu list is used to prevent re-visiting the same place, and the stopping criterion is used to terminate global search and start local search, for solving constrained optimization problems. The proposed adaptive relaxation of constraints is gradual based on the number of feasible solutions found, and can enhance the global search, especially for problems with equality constraints. Furthermore, it does not involve any parameters. The proposed C-IDE is tested on benchmark problems with equality and/or inequality constraints, and its performance is shown to be significantly better than four state of the art algorithms (SaDE, εDE, LEDE and ANT-β) in terms of reliability and efficiency. Subsequently, it is applied to 10 chemical engineering optimization problems with equality and inequality constraints. The results show that the C-IDE can find the global optimum reliably and efficiently compared to the recent algorithms. Overall, the proposed adaptive relaxation constraint handling algorithm is robust, efficient and suitable for constrained optimization problems even with equality constraints. It has potential for use with other global stochastic optimization methods.

Chapter 7

Global Optimization of Pooling Problems*

7.1 Introduction

Pooling problems that arise in optimal blending of materials via pools/tanks to produce products are important in petroleum refineries (Baker and Lasdon, 1985) and wastewater treatment (Bagajewicz, 2000). They are also encountered in other engineering fields such as supply chain operations and communications (Misener and Floudas, 2009).[3] In pooling problems, streams from various processing units are mixed and stored in intermediate tanks, called "pools". The stored streams in pools are subsequently allowed to mix to meet varying requirements. These pools enhance the operational flexibility of the process, but complicate the decision making for blending of pools into final products.

Finding the least costly mixing recipe from intermediate streams to pools and then from pools to final products is referred to as the pooling problem. It can be formulated as an optimization problem whose objective is to minimize cost (or maximize profit) by optimal allocation of intermediate streams to pools and the blending of pools into final products. The main difficulties in solving the pooling problems are: they involve nonlinearities and non-convexities leading to the existence of several optima, the need to find the global optimal solutions, and the presence of many equality/inequality constraints. Consequently, a robust, efficient and yet simple global optimization technique is required for reliably solving pooling problems.

Many researchers have studied the global optimization of pooling problems because a slightly better solution can lead to large savings for the companies. The

* This chapter is based on the paper - Zhang, H. Lee, K.L. and Rangaiah, G.P., New formulation and approach for global optimization of pooling problems, submitted to *AIChE J.*

literature on the pooling problems is on two major areas: problem formulation and solution methodologies (Almutairi and Elhedhli, 2009). From the modeling perspective, several formulations have been proposed such as *p*-formulation (Haverly, 1978), *q*-formulation (Ben-Tal et al., 1994), *pq*-formulation (Tawarmalani and Sahinidis, 2002) and the generalized formulation (Audet et al., 2004; Meyer and Floudas, 2006). This paper focuses on the optimization method only, and the *p*-formulation is used as it is widely studied and easy to understand.

Haverly (1978) is the first one to study the solution of pooling problems using recursive linear programming method. The main drawback of this method is that it does not always converge to the global optimum and it is sensitive to the starting point. Further, with increasing number of pools and products, recursive methods become more unstable and face computational difficulties. Successive linear programming (SLP) algorithms have also been applied to solve the pooling problems (Main, 1993). In SLP, the nonlinear problem is solved via a sequence of the linear problems. Lasdon (1979) applied an SLP algorithm for solving pooling problems. Later, Baker and Lasdon (1985) applied SLP to improve blending schemes at Exxon.

Generally, a complex problem can be decomposed into a simpler problem by fixing values of certain variables. Based on this approach, several extensions of the decomposition method have been developed and applied to solving the pooling problems. For example, Floudas and Aggarwal (1990) introduced a decomposition approach for pooling problems. A disadvantage of decomposition strategy is that it does not guarantee the convergence of the solution to the global optimum (Foulds et al., 1992; Sahinidis and Grossmann, 1991). Visweswaran and Floudas (1990) proposed a deterministic global optimization (GOP) algorithm and tested it on Haverly's pooling problems. This algorithm is based on rigorous solution of the problem through a series of primal and relaxed dual problems until the upper and lower bounds of these problems converge to the global optimum. Subsequently, Visweswaran and Floudas (1993) proposed an improved version of their deterministic GOP algorithm, which can solve more complex pooling problems involving more pools, products and quality components.

Many researchers have used branch and bound method as the basic algorithm for solving the pooling problems. This method is based upon partitioning, sampling, and subsequent lower and upper bounding procedures, which are iteratively applied to the collection of active subsets within the feasible set. Branch and bound methods are non-heuristic in the sense that they maintain provable upper/lower bounds on the globally optimal objective value. An important factor influencing their convergence rate is the quality of the convex relaxations and the tightness of the bounding procedure. Hence, many studies have focused on the bounding procedures that would provide tight bounds within the branch and bound framework. Accordingly, many modifications of the branch and bound method have been proposed and applied to solving the pooling problems.

Foulds et al. (1992) introduced the approach of bilinear envelopes with the branch and bound algorithm, and tested it on a number of pooling problems. Ben-Tal et al. (1994) studied the principle of reducing duality gap between a non-convex program and its Lagrangian dual. This idea was implemented in a branch and bound algorithm to find the approximate global solution and the lower bound of the global optimum value. Visweswaran and Floudas (1996) implemented the branch and bound framework into GOP (Visweswaran and Floudas, 1990) method. The combination of these strategies is to prune the tree and provide tighter under-estimators for the relaxed dual problems. Consequently, the algorithm complexity is reduced and its efficiency is increased, and it was successfully applied to some pooling problems. Adhya et al.(1999) introduced a Lagrangian relaxation approach for developing tighter lower bounds for pooling problems using *p*-formulation. Later, Tawarmalani and Sahinidis (2002) showed that it is no tighter than the linear programming relaxation obtained using bilinear envelopes for pooling problems with *pq*-formulation.

Almutairi and Elhedhli (2009) suggested a new Lagrangean relaxation for solving pooling problems. This relaxation targets all nonlinear constraints, and results in a Lagrangean sub-problem with a nonlinear objective function and linear constraints. The proposed relaxation is integrated with the branch and bound method, and shown to be often tighter than the previously developed relaxations for pooling problems. Pham et al. (2009) proposed a convex-hull discretization approach with the branch and bound method for solving pooling problems efficiently. This approach is based on three concepts: linearization by discretization of nonlinear variables, preprocessing to form a convex hull which limits the size of the search space, and application of integer cuts to ensure compatibility between the original and the discretized problems. Gounaris et al. (2009) proposed a piecewise-linearization scheme to partition the original domain of variables, and applied bilinear relaxation principles for each of the resulting sub-domains. The effectiveness of resulting schemes has been shown on many pooling problems. Comprehensive reviews of global optimization methods and their applications to pooling problems can be found in Floudas et al. (2005), Floudas and Gounaris (2009) and Misener and Floudas (2009).

Recently, several deterministic global optimization algorithms are proposed and applied to pooling problems (Misener and Floudas, 2010; Misener et al., 2010;

Misener et al., 2011). Misener and Floudas (2010) introduced a new formulation for the generalized pooling problem based on the generic superstructure of the water treatment topology, and solved small, medium, large and two very large problems using novel piecewise under-estimators within the branch and bound method. Misener et al.(2011) developed a new formulation for the piecewise relaxation of bilinear functions with a logarithmic number of binary variables. The resulting computational tool, APOGEE can globally optimize the standard, generalized and extended pooling problems. Misener et al. (2010) proposed extended pooling problems with the Environmental Protection Agency (EPA) complex emissions constraints, and solved them using a branch and bound algorithm with a MILP solver (CPLEX) and a local solver (MINOS).

All the above attempts to solve the pooling problems are using deterministic algorithms, which can provide a guaranteed global optimal solution while exploiting the mathematical structure of the given problem. Many studies have investigated different kinds of global optimization algorithms for various applications. The comparison of several deterministic and stochastic global optimization algorithms has also been discussed in the literature (Srinivas and Rangaiah, 2007b; Moles et al., 2003; Teh and Rangaiah, 2003; Mladenovic et al., 2008; Mashinchi et al., 2011; Bonilla-Petriciolet and Segovia-Hernandez, 2010; Nocedal and Wright, 2006; Weise, 2008; Exler et al., 2008). In general, stochastic methods are more robust, require little or no assumption on the characteristics of the optimization problem, and yet provide a high probabilistic convergence to the global optimum. Further, they are usually simple in principle, easy to implement and use. Although stochastic algorithms do not guarantee global optimality in finite time, they can often locate the global optimum in modest computational time compared to deterministic methods (Moles et al., 2003; Exler et al., 2008; Lin and Miller, 2004; Liberti and Kucherenko, 2005; Bonilla-Petriciolet et al., 2010b).

To the best of authors' knowledge, no stochastic global algorithm has been applied to pooling problems, particularly in the last two decades. This could be due to the presence of many equality constraints in pooling problems. On the other hand, there have been many developments on stochastic global optimization in the last two decades (Rangaiah, 2010). In this work, a stochastic global optimization method, namely, integrated differential evolution (IDE) is applied to optimize the pooling problems. The remainder of this chapter is organized as follows. The formulation of the pooling problems is presented in Section 7.2. The application of IDE to pooling problems and performance comparison of IDE with other algorithms are discussed in Section 7.3. Finally, Section 7.4 concludes this chapter.

7.2 Description and Formulation of the Pooling Problems

The mathematical formulation of pooling problems using "*p*-formulation", which has been studied by many researchers, is described in this section. Figure 7.1 shows the general pooling problem network with one bypass, and the nomenclature used for the pooling problem formulation is given in Table 7.1. When a bypass is present in the network, it goes directly to all the products as shown in Figure 1. Its role is similar to the pools except that all the streams going to the products will have same qualities as the bypass stream. To generalize the formulation with bypass streams, the most convenient way is to consider each bypass stream as a pool. Thus, when one bypass exists as shown in Fig. 7.1, the number of pools becomes P+1.

Table 7.1. Notation for Pooling Problems

Type	Symbol	Description
Indices	i	Input streams, $i=1,\ldots, N_j$
	j	Pools, $j=1,\ldots, P$
	k	End products, $k=1,\ldots,R$
	w	Qualities, $w=1,\ldots,L$
Parameters	$A^L{}_i$	Minimum availability of the ith input raw material
	$A^U{}_i$	Maximum availability of the ith input raw material
	B_j	Capacity of the jth pool
	C_{ij}	Unit cost of the ith input stream into jth pool
	$D^L{}_k$	Minimum demand of the kth product
	$D^U{}_k$	Maximum demand of the kth product
	L	Total number of component qualities
	N_j	Number of input streams entering pool, j
	P	Total number of pools
	R	Total number of end products
	S_k	Unit selling price of the kth product
	Z_k	Quality requirement of the kth product
	λ_{ijw}	wth quality specification of the ith input stream entering the jth pool
Decision Variables	Q_{jw}	wth quality of the intermediate stream from the jth pool
	X_{ij}	Flow rate of ith input stream entering into the jth pool
	Y_{jk}	Flow rate from jth pool to the kth product (Intermediate stream)

The objective function is to minimize the total loss (negative profit), which is defined as the difference between the cost of required raw materials and the revenue from selling the final products as shown below.

$$\min \sum_{j=1}^{P+1} \sum_{i=1}^{N_j} (C_{i,j} X_{i,j}) - \sum_{k=1}^{R} S_k (\sum_{j=1}^{P+1} Y_{j,k}) \tag{7.1}$$

It is subject to the following constraints.

Mass balance on pools: $$\sum_{i=1}^{N_j} X_{i,j} - \sum_{k=1}^{R} Y_{j,k} = 0 \quad j = 1,\ldots,P+1 \tag{7.2}$$

Quality balance on pools:

$$Q_{j,w}(\sum_{k=1}^{R} Y_{j,k}) - \sum_{i=1}^{N_j} (\lambda_{i,j,w} X_{i,j}) = 0 \quad j = 1,\ldots,P; \; w = 1,\ldots,L \tag{7.3}$$

Product demand: $\sum_{j=1}^{P+1} Y_{j,k} - D_k \le 0 \quad k = 1,\dots,R$ (7.4)

Quality restrictions on products:

$$\sum_{j=1}^{P}(Q_{j,w}Y_{j,k}) + \lambda_{1,P+1,w}Y_{P+1,k} - Z_{k,w}\left(\sum_{j=1}^{P+1}Y_{j,k}\right) \le 0 \quad k = 1,\dots,R;\ w = 1,\dots,L \tag{7.5}$$

Bounds for the decision variables:

$$0 \le X_{i,j} \le \min\{A_i^U, B_j, \sum_{k=1}^{R} D_k^U\} \quad i = 1,\dots,N_j; \quad j = 1,\dots,P$$

$$0 \le Y_{j,k} \le \min\{D_k^U, B_j, \sum_{i=1}^{N_j} A_i^U\} \quad j = 1,\dots,P; \quad k = 1,\dots,R \tag{7.6}$$

$$\min\{\lambda_{i,j,w}\} \le Q_{j,w} \le \max\{\lambda_{i,j,w}\} \quad i = 1,\dots,N_j;\ j = 1,\dots,P;\ w = 1,\dots,L$$

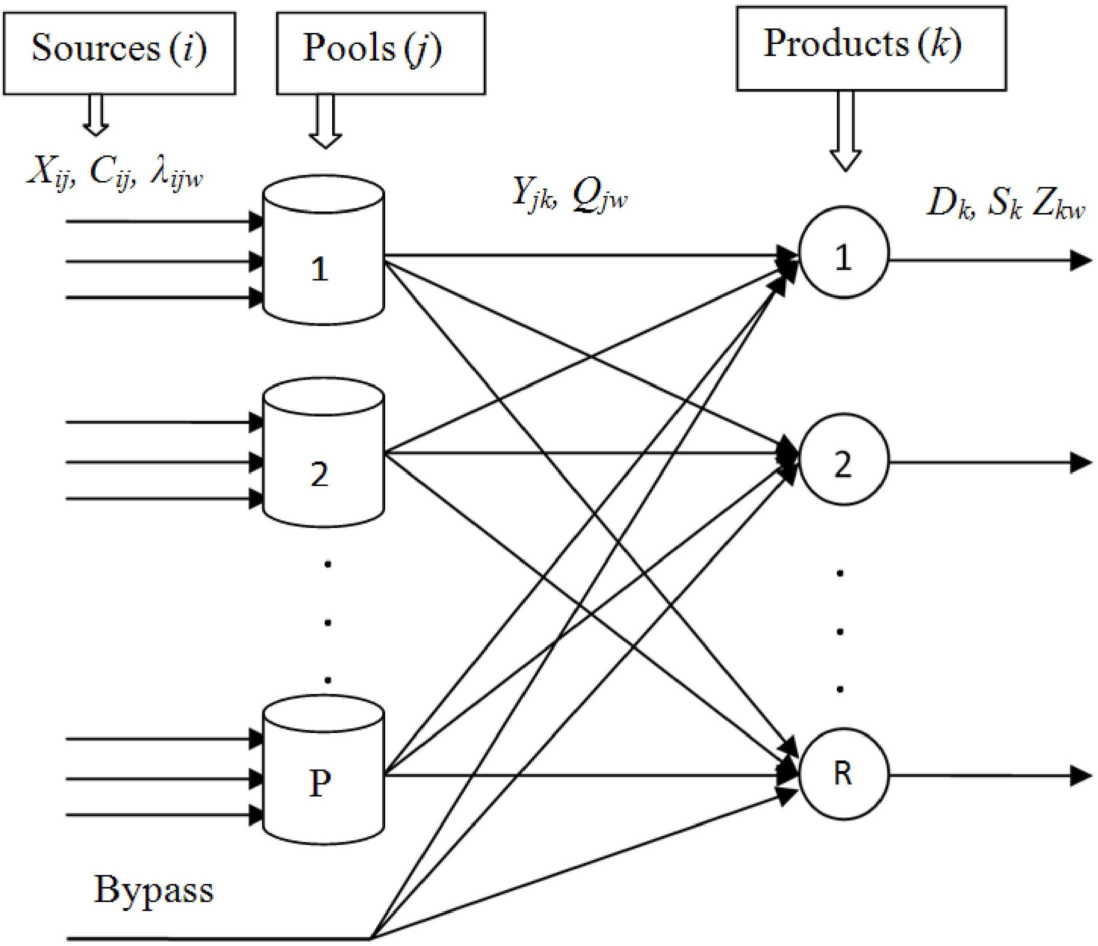


Figure 7.1. Blending Network for the Pooling Problem Formulation

It is obvious that the pooling problem becomes more complicated when a large number of pools, products and quality components are present in the blending network. These lead to pooling problems with a large number of equality and inequality constraints, and local optima. Therefore, a robust and reliable global optimization algorithm is required for solving such problems.

7.3 Application to Pooling Problems

In this section, IDE is evaluated on 13 widely used pooling problems with many variables and constraints for testing and comparing the optimization algorithms. Number of variables in these problems is from 7 to 168, the number of constraints (both equality and inequality constraints) is from 6 to 48, and the number of qualities is one or more. Tables 7.2a and 7.2b summarize the basic details of these pooling problems with single and multiple qualities respectively. The IDE algorithm is implemented in MATLAB platform because it is widely used by researchers and practitioners in diverse fields, for solving these pooling problems. It does not require any problem reformulation, prior knowledge of the problem and is easy to use.

Table 7.2a. Basic Details of Pooling Problems with Single Quality

Problem Name →	Haverly 1	Haverly 2	Haverly 3	Foulds 2	Foulds 3	Foulds 4	Foulds 5	Ben-Tal 4
Input streams	3	3	3	6	11	11	11	4
Pools	1	1	1	4	8	8	4	2
Products	2	2	2	4	16	16	16	2
Quality	1	1	1	1	1	1	1	1
Variables	7	7	7	22	168	168	100	8
Constraints (EC+IEC)	3+4	3+4	3+4	6+8	16+32	16+32	8+32	2+4
Global optimum	-400	-600	-750	-1100	-8	-8	-8	-450

Table 7.2b. Basic Details of Pooling Problems with Multiple Qualities

Problem Name →	Ben-Tal 5	Adhya 1	Adhya 2	Adhya 3	Adhya 4
Input streams	5	3	3	6	11
Pools	4	1	1	4	8
Products	5	2	2	4	16
Qualities	2	4	6	6	4
Variables	38	21	25	38	26
Constraints (EC+IEC)	16+20	12+15	14+28	21+28	10+25
Global optimum	-3500	-549.8	-549.8	-561.05	-877.65

7.3.1 Parameter Settings and Initialization

In the numerical experiments, the following parameters were used: population size, NP = 50, learning period, LP = 50, taboo list size = 50 and taboo radius = 0.001D. Stopping criterion is the satisfaction of either the maximum number of rejections, N_{max} = 10 or maximum number of function evaluations, G_{max} = 10000D. The latter is used as another stopping criterion to avoid indefinite loops. These parameter values are the same as those in Zhang and Rangaiah (2011), in order to show the robustness of the algorithm for solving pooling problems as well.

Since this is a stochastic optimization algorithm, random numbers affect convergence speed, reliability of finding the global optimum and solution quality. Hence, 100 independent runs, each time starting from a different random number seed, were performed on each of the pooling problems. A successful run means that the algorithm can achieve the objective function value very close to the known global optimum value, f(x*). Here, the algorithm in a run is considered to be successful if the objective function value found is ≤ [f(x*) + 1e-5] and the solution is feasible. Average number of (objective) function evaluations (NFE) and success rate (SR) respectively assess the convergence speed and reliability. Here, SR is the percentage of successful runs over the total runs, and NFE is the average over only the successful runs out of

100 trials. NFE is a good criterion for evaluating the computational efficiency of the algorithm and it is independent of computer system used. The NFE for IDE includes all the objective function evaluations used by the global and local optimizer. Feasibility rate (FR) is the percentage of runs, wherein the best solution found in the run satisfies all the constraints, and it can be more than SR.

The solution quality (in terms of the best objective function value from all runs, and median and standard deviation (SD) of final objective function value from each of the 100 runs), success rate and number of function evaluations for solving pooling problem by IDE are summarized and discussed in the following section. The computer system employed in this study is Intel Core 2 (Duo CPU 2.66 GHz, 3.25GB RAM) for which MFlops (million floating point operations per second) for the LINPACK benchmark program (at http://www.netlib.org/) for a matrix of order 500 are 594.

7.3.2 Results and Discussion

In order to illustrate the performance of IDE, the convergence for one single quality pooling problem (Haverly-1) is shown in Fig. 7.2. It is obvious that the constraint violation is decreasing very fast with the number of the generations, especially in the early stage. The steady decrease in constraint violation value indicates that the population is moving towards the feasible region. The objective function value of Haverly-1 is initially decreasing and then increases with the generations (Fig. 7.2) because constraints are relaxed greatly in the initial generations. As the generations increase, the constraint relaxation is less and solutions considered feasible in the previous generations become infeasible. This can be seen in the initial generations where the relaxation value is decreasing fast and the objective function value is fluctuating (Fig. 7.2). After 30 generations, the constraint relaxation becomes very small, and so the objective function is steadily improving to the global optimum.

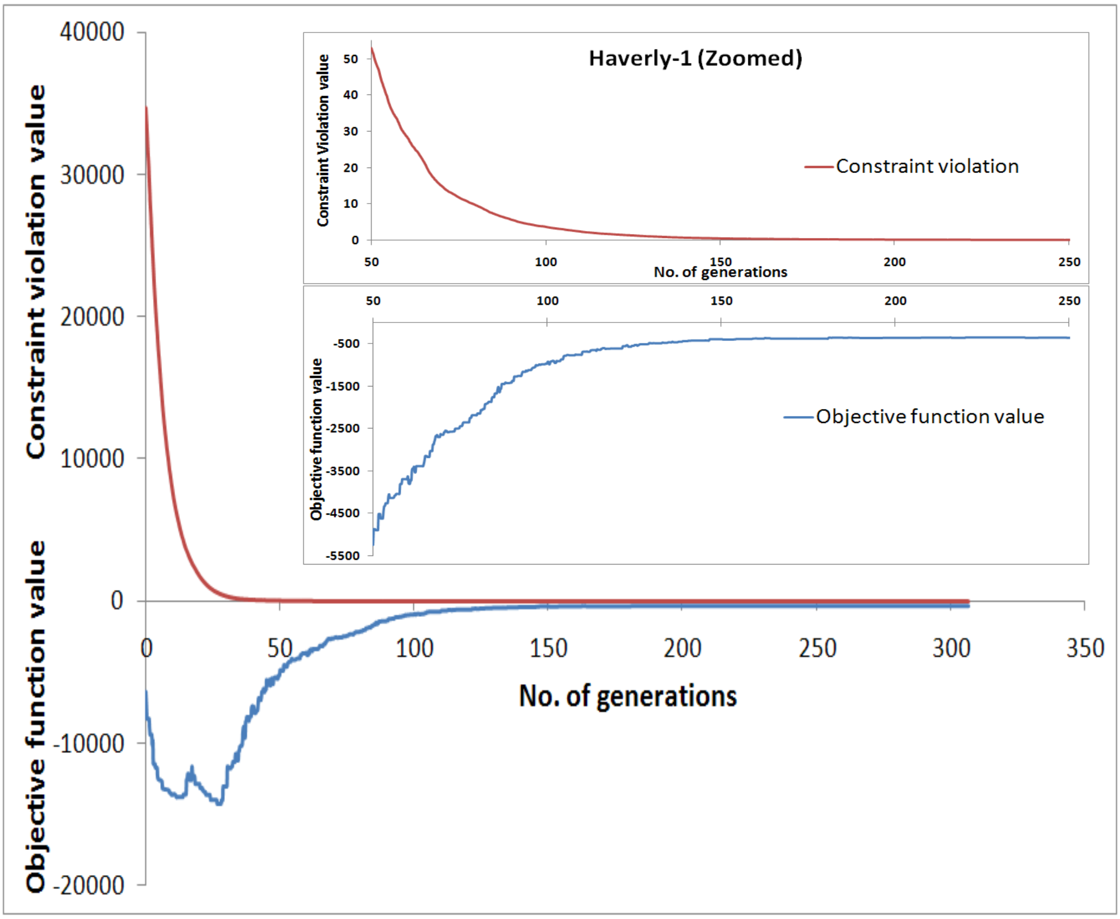


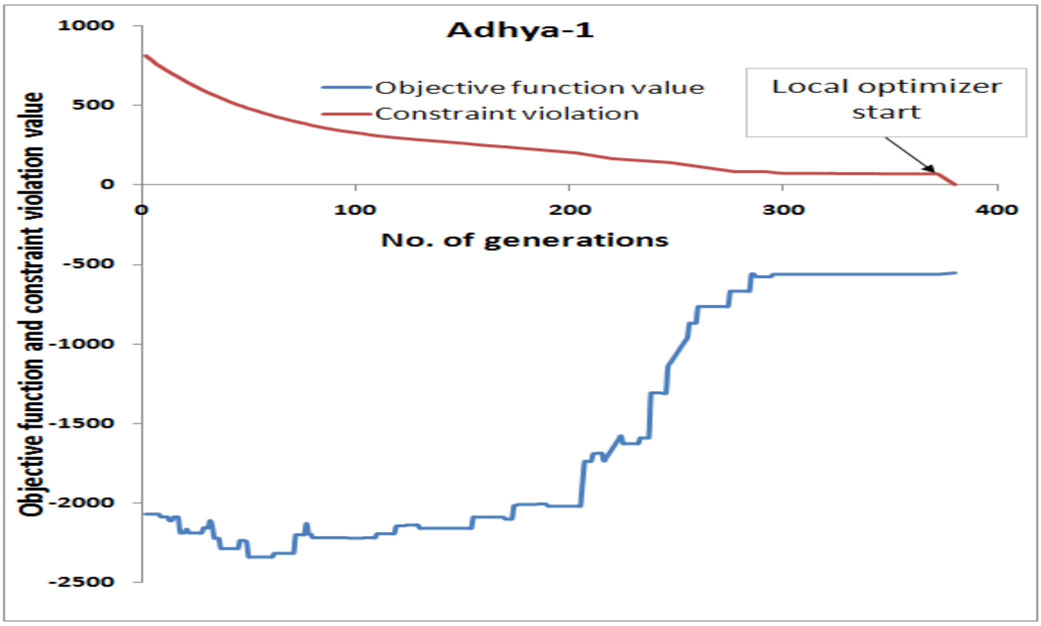
Figure 7.2. Objective Function and Constraint Violation Values Vesus the Number of Generations for Single Quality Problem (Haverly-1); Inset Shows the Profile for Smaller Range of Objective Function and Constraints Violation.

The convergence profiles for two pooling problems with multiple qualities (Adhya-1 and Adhya-3) are shown in Fig. 7.3. Constraint violation value is decreasing much slowly in Fig. 7.3 than in Fig. 7.2, which indicates that it is more difficult to find the feasible solution of Adhya-1 and Adhya-3 compared to Haverly-1. This is agreement with presence of more equality and inequality constraints in multi-quality pooling problems than in single-quality problems. As in Fig. 7.2, constraint violation is decreasing relatively quickly initially and then slowly in the later generations. This is because the relaxation of constraints is more in the initial generations, and more individuals in the population are treated feasible. When the relaxation is reduced faster, it will lead to more feasible individuals according to the selection criterion. Therefore, the constraint violation value is decreasing faster initially.

It can be seen in Fig. 7.3 that the objective function value is fluctuating. This is probably due to reduction in constraint relaxation and many local optima. Comparing Adhya-1 and Adhya-3 in Fig 7.3, the constraint violation of Adhya-1 is decreasing slower than that of Adhya-3; this may due to the more number of constraints in Adhya-3 and the overall constraint violation is large. For Adhya-1, both constraint violation and objective function value do not change after 300 generations, which indicates convergence of the algorithm. After the global search is terminated, local search starts from the best point found by the global search to refine the solution. It is clear from Figure 4 that, for Adhya-1, the global search has found the solution with objective function value -558.6, which is very close to the known global optimum (-549.8). Starting with the former, local search found the global optimum precisely.

For Adhya-3, the constraint violation value is reducing very slowly from 400 to 800 generations which indicates the difficulty to find feasible solutions when the relaxation is small. The objective function value of Adhya-3 is fluctuating even though the constraint violation is not improving. This indicates Adhya-3 has more local optima compared to Adhya-1 as the selection step chooses the better solution based on feasibility, constraint violation and objective function value. After the global search is terminated, the local search starts from the best point found by the global search to refine the solution as indicated in Fig 7.3. For Adhya-3, the global search has found the solution with objective function value of -779.8, which is far from the known global optimum (-561.05). So, local search found only the local optimum (-549.8) but it eliminated constraint violation.

Table 7.3 gives the summary of the results obtained by IDE for all the 13 pooling problems listed in Table 7.2. In Table 7.3, the best optimum found refers to the best solution found in 100 runs, the median and SD are the median and standard deviation of the final objective function values found in 100 runs. These are often used to show the quality of solutions obtained by a stochastic algorithm. As shown in Table 7.3, IDE is able to find the global optimum for all 13 problems in all or many of the 100 runs, and FR is 100%, which indicates that the best solution obtained in each run of IDE is feasible. For Haverly-1 and 3, Foulds-3, Ben-Tal 4 and 5, SR of IDE is 100%; for the remaining problems too, SR of IDE is very high at around 90% except for Adhya-3 with SR of 23%. The median values in Table 7.3 confirm that solutions obtained in most runs are the global solutions except for Adhya-3. Median value in case of Adhya-3 also is close to the global solution. As reported, Adhya-3 is a challenging problem having many local optima, which are very close to the global optimum. For example, Adhya et al. (1999) reported that the local optimizer could not find the global optimum even in 100 runs, each run starting from a different starting point. The SD of the Foulds-2, Adhya-1, Adhya-2, Adhya-3 and Adhya-4 is relatively large; this is because the second best optimum of these problems is very different from the global optimum. Therefore, only a few failures can result in a larger SD.



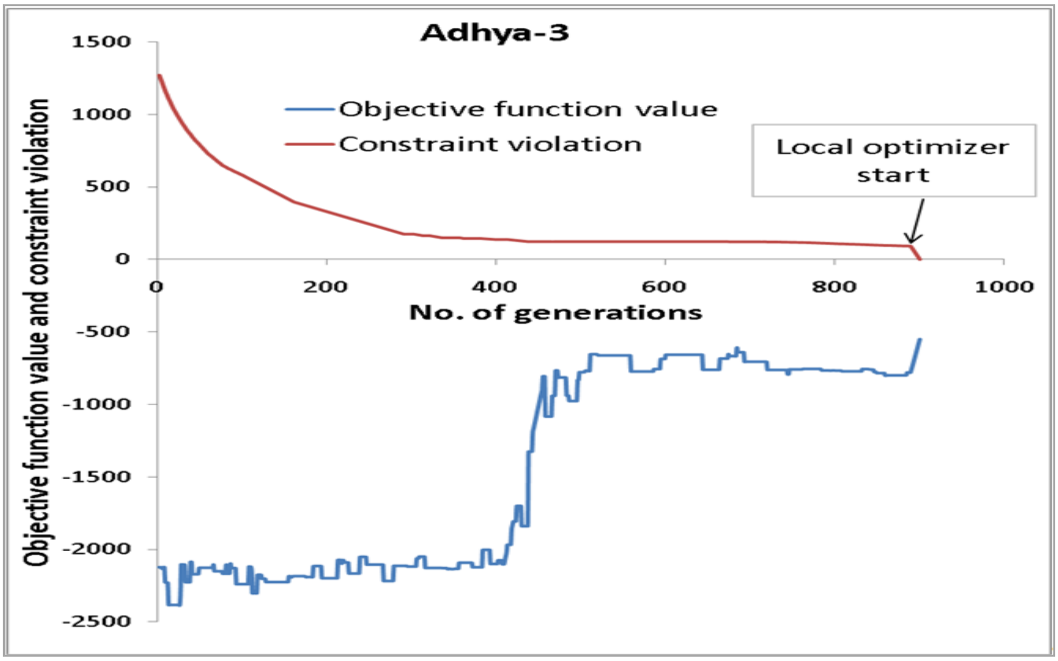


Figure 7.3. Profile of the Objective Function and Constraint Violation Vesus the Number of Generations for Pooling Problems with Multiple Qualities (Adhya-1 and Adhya-3).

Although SR of IDE for Adhya-3 is only 23%, the local solution obtained by IDE is close to the global solution (as can be seen by the median and SD values in Table 7.3). Values of decision variables of the best solution obtained in 100 runs on Adhyra-3 were found to be at or near the bounds. So, we tried a different re-initialization when a variable is outside its bounds. In IDE algorithm, boundary violation of a decision variable is handled by randomly generating a value within the bounds. The modified algorithm is to set the value of decision variable equal to its nearest bound when the decision variable bound is violated. Adhya-3 is tried with the modified algorithm, and the median value of the 100 runs is -559.6 which is very close to the global optimum solution -561.05. Even through SR is less than before (10% versus 23%), the modified algorithm gives a better median value of -559.6 and a better SD of 2.27 in 100 runs, which means that it has higher probability to find a solution very close to the global optimum.

Table 7.3. Performance of IDE for Solving Pooling Problems and its Comparison with Deterministic Methods Reported in Pham et al.(2009)

	IDE						Results from Pham et al.(2009)		
Problem	SR	NFE	CPU Time (s)	Best optimum found	Median of optima found	Standard deviation (SD)	Optimum found	Discretization Method Time (s)	Time (s) for Global Solver
Haverly-1	100	15116	5	-400	-400	4.5E-07	-400	< 1	38
Haverly-2	94	16901	6	-600	-600	1.3E+00	-600	< 1	< 1
Haverly-3	100	12101	4	-750	-750	6.8E-10	-750	< 1	< 1
Foulds-2	85	11534	6	-1100	-1100	6.9E+01	-1100	< 1	> 30000
Foulds-3	100	18533	14	-8	-8	1.9E-08	-8	< 1	> 30000
Foulds-4	92	29377	21	-8	-8	9.0E-02	-8	< 1	> 30000
Foulds-5	91	32667	22	-8	-8	2.6E-01	-8	< 1	> 4000
Ben-Tal 4	100	6738	3	-450	-450	4.1E-13	-450	< 1	> 10000
Ben-Tal 5	100	7285	4	-3500	-3500	1.4E-11	-3500	< 1	< 1
Adhya-1	94	15182	8	-549.8	-549.8	9.5E+01	-547.45	24	196
Adhya-2	75	19794	14	-549.8	-549.8	2.3E+02	-547.45	27	193
Adhya-3	23	29545	19	-561.05	-549.8	4.6E+00	-557.566	10	6446
Adhya-4	87	12448	9	-877.65	-877.65	3.3E+01	-854.492	20	143

The NFE and computational time are also reported in Table 3. For Ben-Tal 4 and 5, IDE requires least NFE and CPU time but Foulds-4 and 5 require much more NFE and CPU time. This is likely due to the larger number of the variables and constraints in Foulds-4 and 5 compared to Ben-Tal 4 and 5 (Table 2). In order to compare the performance of IDE with deterministic methods, results from Pham et al.(2009) are reported in Table 3. Pham et al. reported optimum found and CPU time for two methods: one is for the new convex-hull discretization with branch and bound method (DBBM) and another is for the global solver (LINGO). Note that the computer system used in Pham et al.(2009) is a desktop computer Dell Optiplex GX 620 with a Pentium IV, 3.6 GHZ processor and 1 GB of RAM. A similar computer, namely, Dell Optiplex GX620 with a Pentium IV, 3.6 GHZ and 2 GB of RAM has 297 MFlops for the LINPACK benchmark program for a matrix of order 500. So, the computer in Pham et al. is probably half slow compared to the computer used in this study. On the other hand, MATLAB program used in the present study is slow compared to compiled codes such as FORTRAN code. These differences should be kept in view for CPU time comparison.

In general, the global solver is computationally very expensive compared to both IDE and DBBM (Table 7.3). DBBM can solve the first nine problems efficiently and reliably. It uses much less CPU time compared to IDE. However, for Adhya-1, 2 and 4, DBBM requires more CPU time than IDE and cannot find the global solution accurately. In contrast, IDE can find much better solutions for the pooling problems with multiple qualities (Adhya-1, 2, 3 and 4) compared to DBBM of Pham et al. (2009). Especially, for Adhya-4, the optimum solution obtained by DBBM is very far from the known global optimum (Table 7.3). Therefore, we can conclude that IDE is more reliable and efficient for solving pooling problems with multiple qualities.

Gounaris et al.(2009) proposed a piecewise-linear relaxation with branch and bound method (PLRBBM) for solving pooling problems, and reported its efficiency and reliability for different formulations and partition levels. However, as can be seen in Table 7.4, there are many failures in the trials, and the same partition level is not suitable for all problems (Gounaris et al., 2009). Further, PLRBBM requires prior experience for setting parameter value (i.e., γ used in the selection of the grid points) in order to obtain good performance for a particular problem. In any case, performance of IDE for solving pooling problems is compared with the best/reported results for PLRBBM in Table 7.4. Computer system used is not stated in Gounaris et al.(2009). Further, CPU time for PLRBBM in Table 7.4 is the smallest CPU time reported; maximum CPU time allowed for a run in Gounaris et al. (2009) is 1 hour which means that any convergence failure will cause the CPU time to reach 3600 seconds. On the other hand, CPU time for IDE in Table 4 is the average in all 100 runs, which is more representative of the algorithm efficiency. Ignoring all these differences including the computer systems and programs used and their effect on CPU time, PLRBBM requires much less time than IDE for the single-quality pooling problems tested (Table 7.4). For pooling problems with multiple qualities, performance of IDE is comparable to PLRBBM.

Table 7.4. Comparison of IDE with the Deterministic Method of Gounaris et al., (2009) for Solving Pooling Problems

	IDE CPU(s)	Piecewise Linear Relaxation with Branch and Bound Method: CPU time (s) for Different Formulations and Partitioning Level = 15			
Problem		nf41	nf4r	nf6t	nf7r
Haverly-1	5	< 1	< 1	< 1	< 1
Haverly-2	6	< 1	< 1	< 1	< 1
Haverly-3	4	< 1	< 1	< 1	< 1
Foulds-2	6	< 1	< 1	< 1	< 1
Foulds-3	14	< 1	< 1	< 1	< 1
Foulds-4	21	< 1	< 1	< 1	< 1
Foulds-5	22	< 1	< 1	< 1	< 1
Ben-Tal 4	3	< 1	< 1	< 1	< 1
Ben-Tal 5	4	< 1	< 1	< 1	< 1
Adhya-1	8	4	3	4	5
Adhya-2	14	8	7	8	11
Adhya-3	19	95	76	42	43
Adhya-4	9	4	3	9	8

7.4 Conclusions

This study applied IDE to solving the pooling problems, and its performance is compared to a few deterministic methods recently reported in the literature. Our results indicate that IDE is capable of handling the difficult characteristics of this constrained global optimization problems. In particular, IDE is more robust and efficient in solving the multiple-quality pooling problems. Therefore, the study shows that the stochastic method, IDE is suitable and promising for pooling problems.

Chapter 8

Heat Exchanger Network Retrofitting Using Integrated Differential Evolution*

8.1 Introduction

One of the most frequent problems in industrial plants is excessive consumption of energy. Heat exchanger networks (HEN) are vital in the chemical process industries to reduce energy consumption (i.e., utilities). Although there are numerous papers addressing HEN synthesis, there is significantly less research on HEN retrofitting. Since 1970's, rise in energy costs and environmental concerns have led to extensive research in developing technologies for heat integration. Many approaches have been developed for HEN synthesis, and there are still some challenges in solving HEN synthesis problems.

HEN retrofitting is based on revamping the existing plant, and has greater industrial importance than synthesis. It is performed by changing structure (integer variables) and parameters (i.e., continuous variables) of HEN. The structural changes are mainly for the location of heat exchangers, splitters and mixers. The parameters are heat exchanger surface area, split ratio, etc. Additional factors that should be taken into account in retrofitting include structure constraints, relocation and re-piping costs, reassignment of existing heat exchanger units and effective use of existing heat exchanger areas. The main objectives of HEN retrofitting problems are reduction of utilities consumption, better utilization of existing exchangers, minimizing retrofitting

* This chapter is based on the paper - Zhang, H. and Rangaiah, G.P., One-step approach for heat exchanger network retrofitting using integrated differential evolution.–Under preparation.

cost and identification of required structural modifications (Shenoy, 1995). Owing to many possible structural changes, number of discrete variables will be high. This makes the retrofitting problems highly combinatorial, and the changes are interrelated. Additional complex logical conditions have to be included in the optimization problems. These features make HEN retrofit problems much more difficult to solve.

HEN retrofit methods can be classified into three broad categories: thermodynamic-based approach, mathematical programming methods and approaches combining both (Rezaei and Shafiei, 2009). The first retrofitting procedure was introduced by Tjoe and Linnhoff (1986), where cross-pinch exchangers are identified and eliminated from the existing network, followed by positioning of new exchangers or reusing eliminated exchangers from the previous step. Then, improvements can be made by considering heat-load loops between streams, process and utility heat exchangers. Yee and Grossmann (1986) proposed a mathematical formulation that targets three objectives for retrofitting: (1) maximum utilization of existing exchangers, (2) assigning the existing units to stream matches with minimum piping cost, and (3) minimum stream matches that require new units. However, this model does not consider all possible modification combinations and the optimal usage of area of each exchanger in order to reduce investment cost. To address these issues, Ciric and Floudas (1989) designed a two-step approach for HEN retrofitting. In the first stage, MILP formulation is used to optimize the cost associated with all possible stream combinations. Thus, information regarding how the existing exchangers should be assigned, area increase/decrease needed and how many units to be purchased are obtained. The second stage finds the network structure that has the minimum modification cost. The two-stage approach was later combined into a single stage, MINLP problem by Ciric and Floudas (1990). Due to the complexity of the formulation, Generalized Benders Decomposition technique was employed to divide the problem into smaller, more manageable sub-problems.

Asante and Zhu (1997) developed a technique for HEN retrofitting by combining pinch analysis with mathematical programming. The concept of network pinch was introduced by showing how network structure can affect the energy targets, creating a pinch point similar to process pinch. In their procedure, the network is first diagnosed using thermodynamics and mathematical programming to identify the best topology, which was further optimised in the later stage to obtain the final design. Although pinch analysis is easy to use and provides useful guidelines for implementing an efficient HEN, the outcome of the network depends largely on the decisions and experience of the engineer. Further, the design becomes much more complicated and difficult when a larger HEN is involved.

Ponce-Ortega et al. (2008) proposed a new formulation for retrofit of HENs considering process modifications. The complex MINLP was solved by a local method: GAMs/DICOPT. The authors claimed that the MINLP model is non-convex. Hence, the global optimum can only be found with global optimization methods. Nguyen et al. (2010) extended the MILP model developed by Barbao and Bagajewicz (2005) for HEN synthesis, to HEN retrofit problems; the MILP model considers the addition and location of heat exchangers as well as splitting. The retrofit model is solved using CPLEX in GAMS.

Li and Chang (2010) solved HEN retrofitting problems using simple pinch analysis. The proposed pinch-based approach is keeping additional capital investment to a reasonable level. In particular, every cross-pinch match is removed, and its heat loads on the hot and cold streams are both divided into two according to the pinch temperatures. At either side of the pinch, the divided heat loads on each stream are combined and then matched according to a systematic procedure derived from simple pinch analysis. Li and Chang (2010) solved only small size HEN retrofit problems containing a maximum of five streams in total.

Most formulations of HEN mathematical model involve nonlinear constraints and objective functions which might lead to non-convexities and more than one minimum. Such multi-modal functions pose a starting point problem; if a poor initial guess is provided, the solution might converge to a local solution (Price et al., 2005). In such cases, global optimization methods are required to find a better solution. The MINLP model of HEN retrofitting is NP-hard which makes it difficult for deterministic optimization methods, especially for larger problems (Furman and Sahinidis, 2001; Ponce-Ortega et al., 2008; Chen, 2008; Khorasany and Fesanghary, 2009).

Hence, in recent years, more studies have used stochastic approaches like simulated annealing and genetic algorithm to solve HEN optimization problems. Furthermore, unlike deterministic methods which require derivatives, stochastic global algorithms are applicable to any problem and simpler to implement. Silva et al. (2009) used particle swarm optimization for solving HEN retrofit problems. In this approach, HEN superstructure takes into account the operation and capital costs by maximizing energy recovery and minimizing installation costs. The case is treated as a constrained optimization problem whose objective is to minimize the total cost of HEN, composed of cost of utilities used and of new equipments. The constraints are thermodynamic limits and the obligation of reusing all the existing heat exchangers. Rezaei and Shafiei (2009) used GA with NLP and integer linear programming (ILP) for HEN retrofit problems. Their results show that the proposed method usually finds better solutions than those reported in the literature.

Smith et al (2010) developed a two-level pinch approach for optimization of HEN parameters and HEN structure. In this approach, Levenberg-Marquardt algorithm is used to optimize heat loads of existing matches for maximum heat recovery. The stream split fractions are optimized using sequential quadratic programming (SQP) algorithm. Simulated annealing with a feasibility solver is employed to solve HEN parameter (NLP) problem. Bochenek and Jezowski (2010) proposed a novel superstructure and HEN representation based on hot and cold nodes, and then used GA to solve the HEN retrofitting problem. Their results show that the proposed method is flexible and can find near global optimum efficiently.

All the above studies using stochastic global optimization are two level-approaches, and problem sizes are relatively small. In the first level, HEN structure (integer) variables are generated using a stochastic algorithm; in the second level, continuous variables such as heat exchanger areas and split ratios are optimized using a local optimizer. Owing to the complex nature of HEN retrofit problems, there can be many local solutions even with only continuous variables. However, using a stochastic global method for optimizing these continuous variables will significantly increase the computational time since there are many possible structures. Therefore, it is better to optimize both the integer and continuous variables simultaneously. This will not only reduce the computational time but also prevent the algorithm trapping at a local optimum. Hence, in this chapter, a single-step approach for HEN retrofitting using integrated differential evolution (IDE), is proposed and investigated for retrofitting problems.

The remainder of this chapter is organized as follows. HEN retrofit problem is stated in Section 8.2. The methodology used for global optimization of HEN retrofit problems is described in Section 8.3. Several HEN retrofitting problems and their solution using IDE are presented and discussed in Section 8.4. Finally, conclusions of this work are summarized in Section 8.5.

8.2 HEN Retrofit Problem Statement

In the general formulation of an HEN retrofitting problem, the topology of the existing HEN is fixed and known. Also, given are the parameters of the exiting heat exchangers, heaters and coolers, the initial and final temperatures of the process streams and the utilities. The objective of the problem is usually to minimize the total annual cost by modifying the locations and/or areas heat exchangers as well as adding new heat exchangers. The advantage of using the total annual cost as the objective function is to bring the balance (trade-off) between investment cost and utility cost. The problem is solved by performing a sequence of changes on the existing structural topology and parameters of the exchangers including heaters and coolers.

- ***Structural changes***: heat exchanger relocation by either changing the pairing of hot and cold streams or simply moving it without changing the pairing, adding or removing split streams and adding new heat exchangers.
- ***Parameter changes***: changes in split ratio at the split streams and heat exchange surface areas of the exchangers.

Structural changes usually result in discrete decisions while parameter changes are usually continuous in nature, except when standard heat exchanger areas are used. Furthermore, since the changes are often interrelated, logical conditions used in the formulation greatly increases the number of integer variables. Hence, the overall HEN retrofitting problem is a mixed integer non-linear programming (MINLP) problem and the parameter optimization is a non-linear programming (NLP) problem, both having multiple local minima.

8.2.1 Structural Representation

The HEN structural representation used in this work is adopted from Jezowski et al. (2007) and Bochenek and Jezowski (2010). In order to make the following description easy to understand, frequently used terms are defined here. A stream is defined as any flow that needs to be heated or cooled, but does not change in composition. Hot stream does not literally refer to the temperature of the stream but is defined as a stream that needs to be cooled. Similarly, a cold stream is a stream that needs to be heated. A HEN superstructure is conceptual network including all possible matches of hot and cold streams, with heat exchangers (Kemp, 2007). It basically defines the entire search space of the HEN retrofitting problem. Lastly, nodes are potential locations on the streams where a heat exchanger can be placed upon.

In contrast to the classical HEN structure representation where the superstructure is presented in the form of complex equalities and inequalities with large number of binaries, the structural representation used in this work defines only spaces for placing the heat exchangers using matrices. An example of the HEN structure representation using node-based model is illustrated in Fig. 1. To form the superstructure, side branches are first added to the main streams. As a rule of thumb, split streams are usually added to streams with a large product of flow-rate and heat capacity (FCP), in order to decrease the utility cost. This is because, under thermodynamic analysis, splitting a stream with large FCP helps to distribute the energy of the streams more effectively and thus less energy is wasted. See Kemp (2007) for a more detailed explanation of the FCP criteria in pinch analysis. Next, a number of nodes are distributed along all the hot and cold streams including split streams, if any.

This method of creating superstructure is requires inputs from the user; these are splitting one or more streams and number of nodes on each stream including split streams. This is perhaps an advantage of using this representation as it enables the user to define specific characteristics of the problem, taking into account his understanding of the problem. For example, the number of nodes on a stream can be defined according to the spatial constraints at the plant site. Also, heuristics and results of a more systematic analysis (e.g., from pinch analysis) are helpful.

8.2.2 Matrices Used in Superstructure Representation

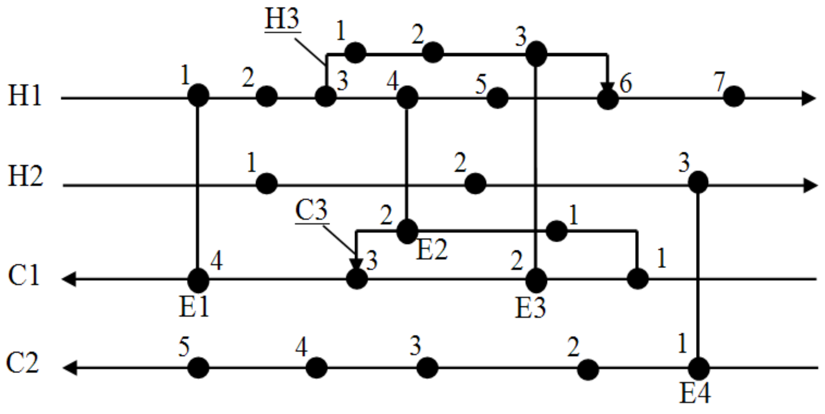
To form the matrices in the superstructure representation, start by numbering all the main streams first, followed by the split streams. Hot and cold streams are given separate set of numberings. Next, nodes on each stream are numbered separately, starting from the inlet of the stream. The superstructure can thus be represented using two node vectors $\mathbf{NOD^H}$ and $\mathbf{NOD^C}$, and two split matrices $\mathbf{SPL^H}$ and $\mathbf{SPL^C}$. These vectors have the follow structures.

- $\mathbf{NOD} = \begin{bmatrix} n_1 \\ \vdots \\ n_i \\ \vdots \\ n_{NB} \end{bmatrix}$ where elements of the vector is NB, the total number of hot (or cold) streams plus their split streams, and element n_i equals to the number of nodes on the ith stream.
- $\mathbf{SPL} = \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ \vdots & \vdots & \vdots \\ S_{NS,1} & S_{NS,2} & S_{NS,3} \end{bmatrix}$ where number of rows is NS, the total number of split streams on hot (or cold streams), and number of columns is 3. The 1st column shows the stream number where the split stream occurs, the 2nd column shows the node number where the split occurs and the 3rd column shows the node number where the split stream mixes with the main stream.

The HEN retrofitting superstructure is fixed and unique for each retrofitting problem. Hence, all genetic operators, i.e., mutation and crossover, in an evolutionary optimization algorithm will only be performed on the structural matrix, **SM**. This matrix shows a specific HEN topology, where each heat exchanger is connected to a unique pair of nodes on the hot and cold streams.

- $\mathbf{SM} = \begin{bmatrix} \text{hot stream}_1 & \text{hot node}_1 & \text{cold stream}_1 & \text{cold node}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \text{hot stream}_{\text{NA}} & \text{hot node}_{\text{NA}} & \text{cold stream}_{\text{NA}} & \text{cold node}_{\text{NA}} \end{bmatrix}$ with number of rows equal to NA, the maximum number of heat exchangers that can exist in the HEN, and four columns. First 2 columns involve the hot node address while the last 2 columns involve the cold node address.

In summary, HEN retrofitting superstructure is built by adding potential splits (split streams) and potential points (nodes) where heat exchangers can be placed in the existing HEN topology. Fig. 8.1 shows an example of how the HEN can be represented using the matrices. H1, H2, H3, C1, C2 and C3 are main streams/branches, and H4 and C4 are split/side streams/branches. Black dot and arrow head denotes a hot or cold node for placement of a potential heat exchanger. For instance, dot number 6 on H2 is a hot node because it lies on a hot stream. Construction of matrices **SM**, **NOD** and **SPL** in Figure 1 is as explained earlier.



$$\mathbf{SM}\begin{bmatrix} (1,1) & (1,4) \\ (1,4) & (3,2) \\ (3,3) & (1,2) \\ (2,3) & (2,1) \end{bmatrix}$$

$$\mathbf{NOD}^{\mathbf{H}} = [7,3,3]^T$$
$$\mathbf{NOD}^{\mathbf{C}} = [4,5,2]^T$$
$$\mathbf{SPL}^{\mathbf{H}} = [1,3,6]$$
$$\mathbf{SPL}^{\mathbf{C}} = [1,1,3]$$

Figure 8.1: Example of a HEN Structure and its Corresponding Matrix Representation

8.2.3 HEN Model Calculations

There are several ways to formulate the mathematical model for parameter calculations. Such formulations are tested and evaluated base on their effectiveness and ease of application in MATLAB used in this study. The model employed in this work is based on the work of Kotijabasakis and Linnhoff (1986). In this model, heat transfer surface area is used as the defining parameter for the heat exchanger instead of the usual heat load. Hence, following this model, decision variables used in IDE optimization will be heat transfer surface area of each exchanger in the HEN structure defined by the chosen discrete variables. Assumptions imposed on this model are similar to those used in other works:

- Heat exchangers are of shell-and-tube type
- Strictly counter-current flow scheme in all heat exchangers, i.e., temperature driving force is given by the logarithmic mean temperature difference
- Enthalpy changes of streams are linear functions of temperature, i.e. FCP values are constant in each heat exchanger

Under these assumptions, a heat exchanger can be modeled by the following equations:

$$A = \frac{Q}{U\,\Delta T_{lm}} \tag{8.1}$$

$$Q = FCP^{c}\left(T_{out}^{C} - T_{in}^{C}\right) \tag{8.2}$$

$$Q = FCP^{H}\,(T_{out}^{H} - T_{in}^{H}) \tag{8.3}$$

$$\Delta T_{lm}^{n} = \frac{(T_{in}^{H} - T_{out}^{C}) - (T_{out}^{H} - T_{in}^{C})}{\ln\frac{(T_{in}^{H} - T_{out}^{C})}{(T_{out}^{H} - T_{in}^{C})}} \tag{8.4}$$

$$\Delta T_{lm} \geq \Delta T_{min} \tag{8.5}$$

By combining Eqs (8.3) to (8.5), followed by some algebraic manipulations, the model used in this work can be represented as follows:

$$T_{out}^{i} = \alpha^{i} T_{in}^{i} + \left(1 - \alpha^{i}\right) T_{in}^{i*} \quad \forall i = 1, 2, \dots, n \tag{8.6}$$

where $\alpha^{i} = \frac{(\gamma^{i*} - \gamma^{i})}{\gamma^{i*} - \gamma^{i} e^{(\gamma^{i} - \gamma^{i*})}} \quad \forall i = 1, 2, \dots, n$ (8.7)

$$\gamma^{i} = \frac{UA}{FCP^{i}}, \ \gamma^{i*} = \frac{UA}{FCP^{i*}} \quad \forall i = 1, 2 \dots, n \tag{8.8}$$

$$0 < \alpha^{i} < 1 \tag{8.9}$$

$$A, FCP, U > 0 \tag{8.10}$$

Superscript (*) is used to differentiate one stream from the other in the heat exchanger, i.e. if the hot stream has a superscript (*), the cold stream does not and vice versa. n is used to represent the total number of heat exchangers in the network. Advantages of this model are: it is based on heat exchanger surface area instead of heat load, and it does not require thermodynamic constraints on temperature approaches (Bochenek and Jewozski, 2010).

In a typical HEN retrofitting optimization, exchanger equipment and stream data are usually available. Hence, FCP^{H}, FCP^{C}, U and the starting temperatures are known. Therefore, by setting the heat transfer surface area (A) as the decision variable in the optimization, we will have new values of A during the iterative calculations, making Eq. (8.6) linear. Thus, Eqs. (8.6) to (8.8) can be solved as a series of linear equations for the dependent variables, making the calculations much easier. Also, since heat transfer surface area is required directly in the calculation of investment costs, it is more convenient to base the calculations on it. In the typical formulation, additional constraints are added on temperature approaches to ensure a feasible HEN. However, in this formulation, such constraints are not required. For more information on this, refer to Bochenek and Jewozski (2010).

Each heat exchanger can be represented by a set of 2 equations similar to Eq. (8.6): one for hot stream and another for cold stream. Hence it is convenient to use a matrix notation:

$$\begin{bmatrix} T_H^{in} \\ T_{in}^C \end{bmatrix} \begin{bmatrix} \alpha^H & 1-\alpha^H \\ 1-\alpha^C & \alpha^C \end{bmatrix} = \begin{bmatrix} T_{out}^H \\ T_{out}^C \end{bmatrix} \tag{8.11}$$

Where the parameters α^H and α^C are defined as follows:

$$\alpha^H = \frac{(\gamma^C - \gamma^H)\mathrm{e}^{\gamma^C}}{\gamma^C\mathrm{e}^{\gamma^C} - \gamma^H\mathrm{e}^{\gamma^H}} \tag{8.12}$$

$$\alpha^C = \frac{(\gamma^H - \gamma^C)\mathrm{e}^{\gamma^H}}{\gamma^H\mathrm{e}^{\gamma^H} - \gamma^C\mathrm{e}^{\gamma^C}} \tag{8.13}$$

$$\gamma^H = \frac{UA}{FCP^H}\ ,\ \gamma^C = \frac{UA}{FCP^C} \tag{8.14}$$

The model so far assumes a fixed FCP, suggesting that the split ratios are fixed. However, in our HEN retrofitting model, the split ratios are variables and therefore equations at the splitters and mixers have to be added. By setting the split ratios as decision variables in the optimization, we can still get linear Eq. (8.11) for optimization calculations. Eqs. (8.15) - (8.18) and Fig. 8.2 shows the energy and mass balance of the splitters and mixers.

Energy balance of splitters: $T_{out}^1 = T_{in}\ ,\ \ T_{out}^2 = T_{in}$ (8.15)

Energy balance of mixers: $\frac{T_{in}^1 FCP_{in}^1}{FCP_{in}^1 + FCP_{in}^2} + \frac{T_{in}^2 FCP_{in}^2}{FCP_{in}^1 + FCP_{in}^1} = T_{out}$ (8.16)

Mass balance of splitter: $FCP_{in} = FCP_{out}^1 + FCP_{out}^2$ (8.17)

Mass balance of mixers: $FCP_{out} = FCP_{in}^1 + FCP_{in}^2$ (8.18)

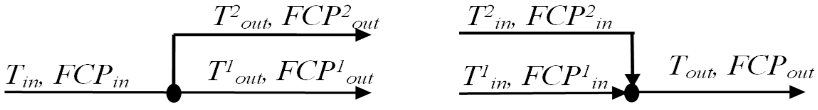


Figure 8.2: Symbols Used in Equations for Splitters (left plot) and Mixers (right plot).

After calculating the temperature at each node on each stream, a heater or cooler is added to the end of cold or hot stream if the temperature of the last node does not reach the required target temperature of the particular stream.

The decision variables include both integer and continuous variables. The number integer variables depends on the possible number of heat exchangers in the HEN, which is assumed to be number of existing exchangers including heaters and coolers plus two (Rezaei and Shafiei, 2009). Each exchanger requires 4 integer variables for its representation. Continuous variables include heat changers areas and stream split ratios, and so number of continuous variable is the number of heat exchangers plus number of stream splits assumed. The inequality constraints are that the node temperatures monotonically decrease (or increase) along each hot (or cold) stream. Also, final temperature of each hot (cold) stream should not be less (more) than or equal to the target temperature.

8.3 Methodology for Global Optimization of HEN Retrofit

Differential evolution (DE) is a global optimization technique classified under stochastic global optimization method. It is also developed using the idea of evolving an initial randomly generated population of potential solutions using operators like mutation, recombination and selection. However, in contrast to GA, which is often better suited for combinatorial problems as GA uses bit strings to encode the parameters and logical operators to modify them, DE is primarily a numerical optimizer (Price et al., 2005).

In this paper, IDE algorithm of Zhang et al. (2011) is modified and used in this study. It integrates DE with taboo list, parameter self-adaptive strategy, and a local optimizer. This algorithm uses four mutation strategies: DE/rand/1/bin, DE/rand to best/2/bin, DE/rand/2/bin and DE/current to rand/2/bin (Price et al., 2005), each with an initial probability of 0.25 and crossover probability Cr of 0.5. In subsequent generations, the probability of employing a particular strategy and its respective Cr changes according to the cumulative success rate from the previous generations. Hence, the program is termed “self-adaptive” as it learns and adapts itself to different kinds of problems by varying the mutation strategies and Cr values. The mutation/scaling factor F is also randomly chosen using normal distribution with mean of 0.5 and standard deviation of 0.3, to balance between the speed of convergence and the extent of search. The taboo list is used to ensure that the Euclidean distance between the trial vector and each vector on the list is larger than a specified taboo radius, to prevent the search from revisiting the same area. Finally, after the termination condition for global search is fulfilled, the best solution is optimized using SQP (a local optimizer) available in MATLAB.

8.3.1 Handling Integer Variables

Integers and binary variables may be encountered in chemical engineering problems. Within the optimization algorithm, these are also represented as continuous variables and converted into integers for evaluating the objective function and constraints. In this study, integers are handled by rounding the continuous variables to the nearest integers. Rounding off strategy is used for the generation of HEN structure. For example, in the SM in Fig. 7.1 to represent the HEN structure, the maximum number of hot (or cold) streams is 3 and minimum is 1; the maximum number of hot and cold stream node is 7 and 5 respectively, whereas the minimum number of hot/cold stream nodes is 0. So the generation of each element row of the SM matrix will be using the upper bound X^{ub} = [3.49 7.49 3.49 5.49] and lower bound X^{lb} = [0.51 -0.49 0.51 -0.49]. In the initialization stage, each row of SM is generated randomly between these lower and upper bounds followed by rounding off the continuous variables to the nearest integers. Note the use of 0.49 below the actual lower bound and also above the actual upper bound. This ensures equal probability of selecting any of the integers in the range of actual bounds.

8.3.2 Handling Constraint and Boundary Violations

In HEN retrofit problems, there are only inequality constraints on node temperatures along the streams. All these are handled using the popular penalty function method. The penalty function method converts the constrained problem into an unconstrained one by penalizing the infeasible solutions using penalty weights. A high value (1E+06) is used for the penalty weight for all the problems in this study. If any constraint is violated, the absolute value of the constraint violation is multiplied by the penalty weight and is added to the objective function, assuming that the problems are minimization type. If many constraints are violated, then each absolute violation is first multiplied with the penalty weight, and all of them are added to the objective function value.

8.3.3 Objective Function

Normally, the objective in HEN design is to design a cost-effective network. Therefore, total annualized cost, comprising operating cost and annualized investment cost (AIC), is taken as the objective function.

$$\text{Objective} = \text{Operating Cost} + \text{AIC} \tag{8.19}$$

$$\text{Operating Cost} = \sum_{j=1}^{N_{\text{heater}}} \mathrm{Q}_j \times \mathrm{C}_{\text{hot}} + \sum_{k=1}^{N_{\text{cooler}}} \mathrm{Q}_k \times \mathrm{C}_{\text{cool}} \tag{8.20}$$

$$\text{AIC} = \left\{\left(\sum_{l=1}^{N_{\text{ex}}} \mathrm{A}_l \times \mathrm{C}_{add}\right) + \left(\sum_{m=1}^{N_{\text{new}}} \mathrm{A}_m \times \mathrm{C}_{new}\right)\right\} \times \left(\frac{i \times (1+i)^n}{(1+i)^n - 1}\right) \tag{8.21}$$

The operating cost is the sum of hot and cold utilities cost. In Eq. 8.20, C_{hot} and C_{cool} are the unit cost of hot and cold utility respectively; N_{heater} and N_{cooler} are the number of heaters and coolers in the HEN; Q_j and Q_k are the heat loads of the each heater and cooler. AIC is calculated from the total investment cost considering interest rate (*i*) and plant life *(n* years). In Eq. 8.21, A_l and C_{add} are the additional area to an existing exchanger and its cost, and A_m and C_{new} are the new heat exchanger area and its cost.

8.3.4 Description of the Overall Methodology

The overall methodology for the global optimization of HEN retrofit problems is shown in Fig.8.3. It shows the main steps in IDE and also in the HEN model calculations for objective function. Decision variables used in HEN retrofit model includes **SM** (structural changes) and heat transfer surface areas and split ratios (parameter changes). First, users need to provide the superstructure such as NOD and SPL matrices, existing HEN topology and stream properties, maximum number of exchangers and number of split streams as well as the lower and upper bounds on decision variables. The IDE algorithm parameters are taken from Zhang et al. (2011). Values used for the 3 problems are same except maximum number of generations, G_{max}. Population size, NP = 50, tabu list size = 50, learning period = 50 and tabu radius = 0.01. G_{max} = 10000 for first and second problems, G_{max} = 20000 for third problem. In the initialization step of IDE, NP solutions/individuals in the initial population are randomly generated within bounds of the decision variables.

As the IDE used in this work is a general optimizer and not customized for HEN problems, solutions formed in the initialization stage or using its mutation and crossover operators could result in infeasible structures. Therefore, each of the individual in the population is sent for the structure feasible check. If the HEN structure generated (a) has more than one heat exchanger connected to the same node,

(b) has a heat exchanger on a node that is being used as a splitter or mixer for a split stream, or (c) has a heat exchanger on a node number that does not exist on the stream, then it will be corrected accordingly. For this, a correction function was created in the HEN model to modify the unacceptable heat exchanger(s), thus creating a feasible HEN structure. Then, the individual with its integer variables for the feasible HEN structure and the continuous variables (heat exchanger surface areas and split ratios) is sent for calculating the stream temperature at each node on each stream. Usually, a slightly change in the temperature of one node can affect the temperatures in other streams due to inter-connections through heat exchangers. Due to the strong correlation of the heat exchanger outlet temperatures at different nodes, convergence check and iterations, if required, are performed for all the nodes on all hot and cold streams.

In the classic formulation, stream temperatures are usually specified as constraints. However, in our model and program, heaters and coolers are automatically added at the end of the streams if the target temperatures are not met. This ensures that the resulting HEN structure always yield solutions where steam target temperatures are met. Furthermore, by reducing the number of constraints, we are able to decrease the complexity of the optimization problem. In this study, a relaxation of 0.1^{o}C on the final stream outlet temperature is used, i.e. no heaters/cooler is added if the stream outlet temperature falls within 0.1^{o}C from the target temperature. This relaxation is reasonable since the outlet temperatures in industry can vary within small ranges due to heat losses, disturbances etc. Further, optimization is based on estimation and reasonable assumptions.

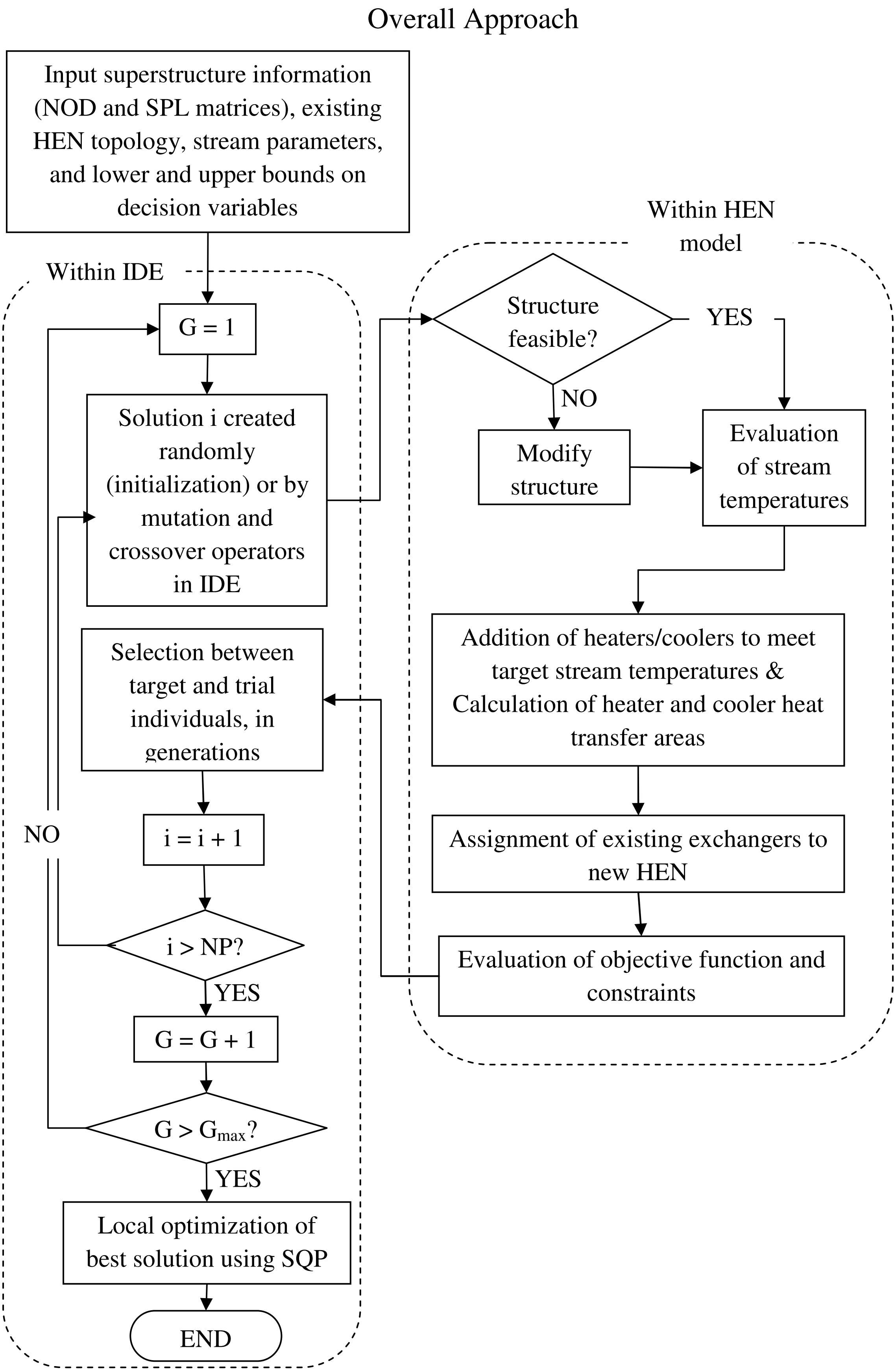

Overall Approach
Input superstructure information (NOD and SPL matrices), existing HEN topology, stream parameters, and lower and upper bounds on decision variables
Within IDE
G = 1
Solution i created randomly (initialization) or by mutation and crossover operators in IDE
Selection between target and trial individuals, in generations
i = i + 1
i > NP?
YES
NO
G = G + 1
G > G$_{max}$?
YES
Local optimization of best solution using SQP
END
Within HEN model
Structure feasible?
YES
NO
Modify structure
Evaluation of stream temperatures
Addition of heaters/coolers to meet target stream temperatures & Calculation of heater and cooler heat transfer areas
Assignment of existing exchangers to new HEN
Evaluation of objective function and constraints

Figure 8.3: Flowchart of the Overall Approach Employed in This Study for HEN Retrofitting Problems

After all the heat transfer areas of the heat exchanger, heaters and coolers are confirmed, the program will do a simple assignment task. Aim of this task is to reassign the existing heat exchangers to new locations in the retrofitted HEN, and at the same time minimize the increase in heat transfer area required in each existing exchanger. This task is achieved by sorting the existing heat exchangers, with respect to their heat transfer areas, in decreasing order. The same is done to the required exchangers in the retrofitted HEN. Finally, the objective function, which includes investment cost and utility cost, is evaluated. The investment cost needed can be calculated based on the cost of increase in heat transfer surface area of existing heat exchangers, cost of new heat exchangers and cost of relocation and re-piping. The utility cost is calculated based the heater and cooler duties. This process is repeated until the termination criterion is met. Then, the best point obtained over all generations is refined using the local optimizer with respect continuous variables only. SQP method in MATLAB is used as the local optimizer in order to find the precise solution with fixed HEN structure because of its limitation in handling integer variables.

8.4 Case Studies

The whole program is implemented in MATLAB because it is versatile and used by researchers and practitioners in diverse fields. It includes global optimization algorithm, IDE, and HEN retrofit models. The computer system employed in this study is Intel Core 2 (Duo CPU 2.66 GHz, 3.25GB RAM) for which MFlops (million floating point operations per second) for the LINPACK benchmark program (at http://www.netlib.org/) for a matrix of order 500 are 594. In order to show the reliability of the stochastic optimization method, each problem is solved 10 times with a different value for the random seed. The computational time and success rate of finding better solutions as well as the correct structure are reported.

Three examples from small to large scale from the literature are studied to show the effectiveness and reliability of the proposed approach. The first example is taken from Shenoy (1995), and it was recently studied by Rezaei and Shafiei (2009) using GA with non-linear programming and integer linear programming. The second and third examples are relatively large size problems which are for two different crude oil preheat trains. One was investigated by Briones and Kokossis (1999) and Rezaei and Shafiei (2009); and the other more complex one was recently studied by Smith et al. (2010).

8.4.1 Case Study 1

This HEN retrofit example is in a process scheme typically encountered in chemical industries. It involves a portion of a petrochemical process that includes an exothermic reactor and distillation column. From the HEN viewpoint, it has two hot and two cold streams. Details of streams, existing exchangers and other information are given in Table 8.1 and Fig. 8.4(a) (Shenoy, 1995).

Table 8.1. Streams and Cost Data for Case Study 1 (Shenoy, 1995).

Streams	T^{in} (^oC)	T^{out} (^oC)	FCP (kW/^oC)	Cost ($/kW-year)
H_1	175	45	10	
H_2	125	65	40	
C_1	20	155	20	
C_2	40	112	15	
Steam	180	179		120
Water	15	25		10

Notes: U = 0.1 kW/(m^2-K) for all exchangers; capital cost ($) = 30000 + $750A^{0.81}$ for all new exchangers with A in m^2; capital cost ($) = $750\Delta A^{0.81}$ for additional area of ΔA m^2 in an existing exchanger; plant lifetime: 2 years; interest rate = 10%; and LMTD is used for area calculations.

First, the proposed method is used for solve this problem without stream split. Number of integer and continuous variables in the optimization problem is 16 and 4 respectively. The optimal results obtained are compared with Shenoy (1995) which also did not involve the stream split. The results shown in Table 8.2 shows that the proposed method is able to obtain a much better objective function value (total annual cost). The operating and investment costs in Shenoy (1995) are $77863 and $164968 but these are respectively $77850 and $144513 in this work. It is clear that the proposed method obtained the lower investment cost using slightly lower operating cost.

The original HEN for case study 1 is shown in Fig. 8.4 along with the retrofitted network using the proposed approach. It is obvious that, with one additional heat exchanger, both hot and cold utilities are significantly reduced. The success rate of finding the correct structure is 100%, and that of finding a better solution is 90%. This indicates that even though the retrofitted HEN structure is correct, there are multiple local minima with respect to continuous variables alone. The average computation time for solving this problem is 1256 s.

Table 8.2. Comparison of Results for Case Study 1.

	Split?	Operating cost ($)	Investment ($)	Total annual cost ($/year)
Shenoy (1995)	No	77863	164968	172916
This work	No	77850	144513	161117
Rezaei & Shafiei (2009)	Yes	79810	129212	154260
This work	Yes	81150	125799	153634

Note: Operating cost in the existing network is $181200

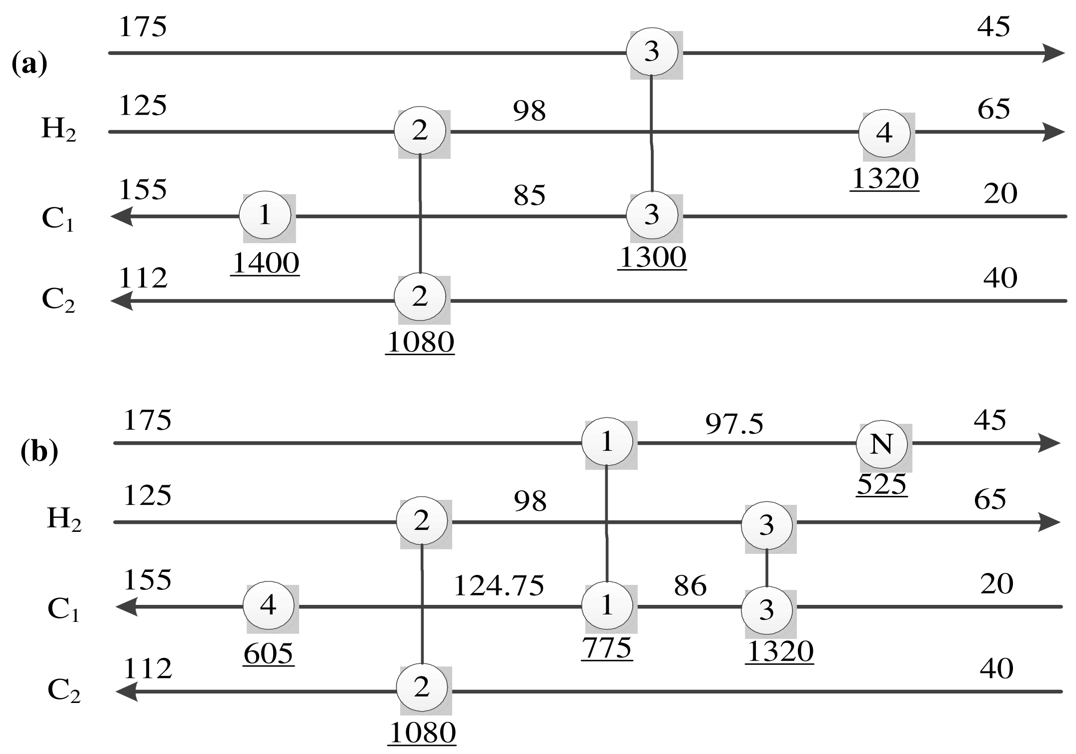


Figure 8.4. HEN in Case Study 1 without Stream Split: (a) Original Network, and (b) Retrofitted Network without Stream Split; Temperatures are in ^{0}C and Under-lined Values are Exchanger Duties

Rezaei and Shafiei (2009) studied the HEN problem in case study 1 considering stream spit. The results show that a much better solution is obtained with stream split compared with no stream split in Shenoy (1995); see Table 8.2. In order to test the proposed method, HEN retrofit with stream split is also studied. One stream split on hot stream 2 is assumed. The stream split ratio obtained from this work is 0.535 compared to 0.53 in Rezaei and Shafiei (2009). The retrofitted network by Rezaei and Shafiei (2009) and in this work are shown in Figs. 8.5(a) and (b) respectively. The structure of both these networks is very similar. However, in retrofitted network obtained in this work, one heat exchanger (E3) did not change location whereas all exchangers changed locations in the network of Rezaei and Shafiei (2009). The total additional area of the present solution is 336.2 m^2, slightly less than that in Rezaei and Shafiei (2009) as shown in Table 8.3. The results in Table 8.2 show that the proposed method obtained a better solution in terms of investment cost and total annual cost, but using slightly more utilities as indicated by the operating cost (Fig. 8.5).

The heat exchanger reassignments and area distribution after the retrofit by the proposed method and method in Rezaei and Shafiei (2009) are shown in Table 8.3. Success rate of finding the correct structure is 100%, and that of finding a better solution is 60%. This again indicates that, though the retrofitted HEN structure is correct, there are still multiple local minima with respect to continuous variables. The average computation time for solving HEN retrofit problem in case study 1 with stream split, is 1475 s which is more that for the problem without stream split. This is expected since HEN retrofit with stream split is more difficult to solve.

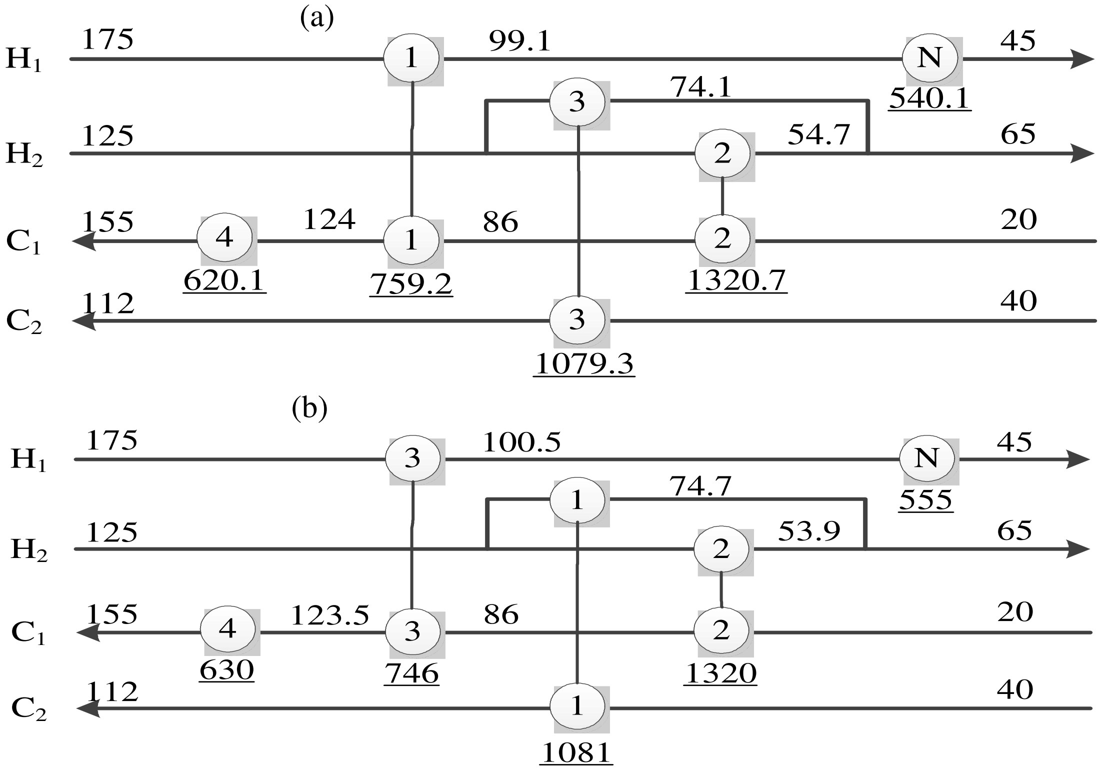


Figure 8.5. HEN in Case Study 1 with Stream Split: (a) Retrofitted Network of Rezaei and Shafiei (2009), and (b) Retrofitted Network in This Work

Table 8.3. Heat Exchanger Reassignments and Area Distribution in Case Study 1 with Stream Split

		Rezaei and Shafiei (2009)		This work	
Match number	Existing area	Retrofit area (m^2)	Additional area (m^2)	Retrofit area (m^2)	Additional area (m^2)
E1	268.7	272.6	3.9	488.6	219.9
E2	358.9	358.6	-	362.7	3.8
E3	256.2	493.6	237.4	256.7	-
E4	217.2	163.0	-	164.7	-
New	-	110.8	110.8	112.5	112.5
Total additional area (m^2)	-	-	352.1	-	336.2

8.4.2 Case Study 2

This example is crude oil preheat train of a petroleum refinery from Briones and Kokossis (1999). It involves six hot and one cold streams. Details of the streams, existing exchangers and other information used for calculations are given in Table 8.4. The original network for crude oil pre-heat train taken from Briones and Kokossis (1999) is shown in Fig. 8.6(a). The retrofitted network using the proposed method in Fig. 8.6(b) uses 3 coolers less than those in the original network and also less utility. Further, the retrofitted network has only 12 exchangers instead of 13 exchangers in the original network.

Table 8.4. Stream and Cost Data for Case Study 2 (Briones and Kokossis, 1999).

Streams	T^{in} (K)	T^{out} (K)	CP (kW/K)	Cost ($/kW-year)
H_1	622	368	86	
H_2	572	393	21.4	
H_3	549	523	184.7	
H_4	503	368	23.5	
H_5	479	451	129.4	
H_6	455	348	11.5	
C_1	316	633	147.9	
Steam	773	772		60
Water	293	313		5

Note: U = 0.265kW/(m^2-K) for all exchangers; capital cost ($) = 3460 + 300*A* for a new exchanger of area *A* m^2; capital cost ($) = 300Δ*A* for additional area of Δ*A* m^2 in an existing exchanger; capital cost($) = 300 for reassignment of an existing exchanger; plant lifetime = 5 years, rate of interest = 0%; LMTD is used for area calculations.

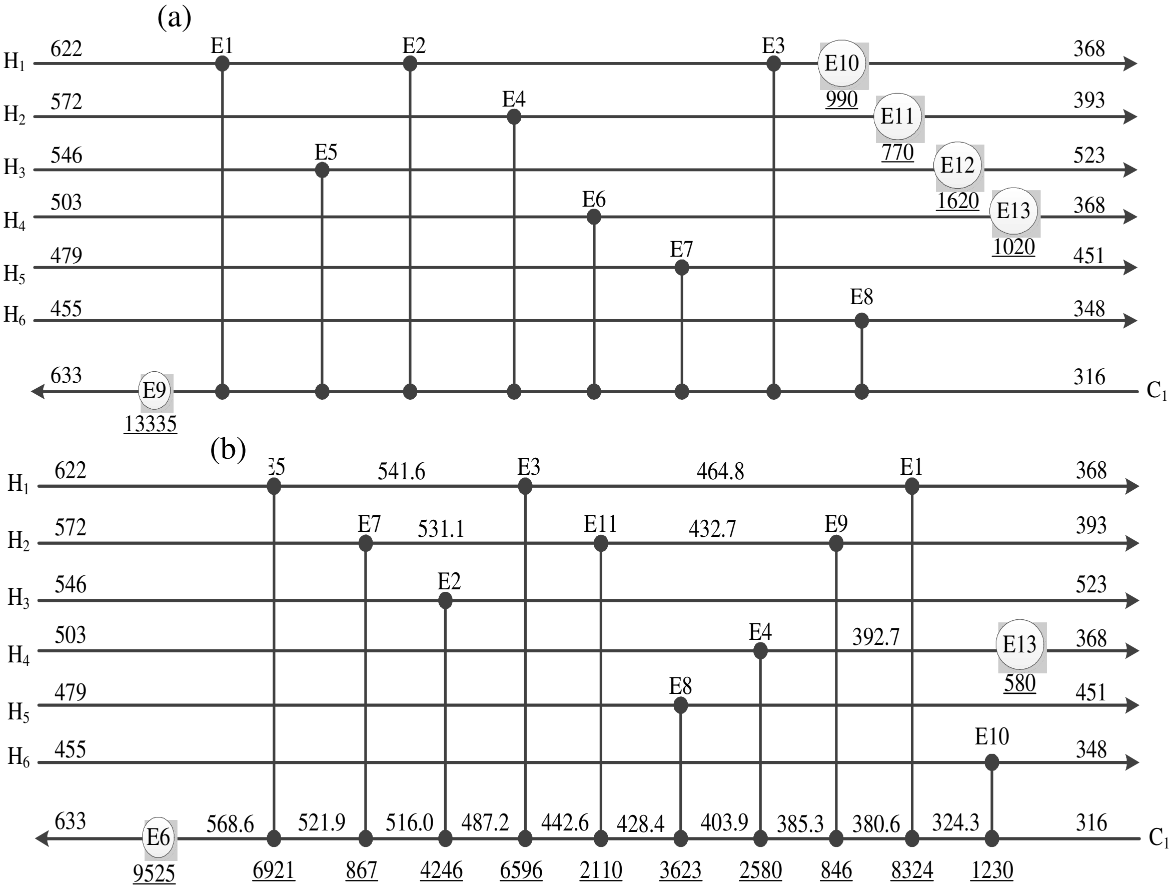


Figure 8.6. (a) Original Network of Case Study 2; (b) Retrofitted Network

Several researchers have studied this example without considering stream split (Briones and Kokossis, 1999; Ma et al., 2000; Rezaei and Shafiei, 2009). Accordingly, this study too does not consider stream split for comparison with these works. Number of integer and continuous variables in the optimization problem is 40 and 10 respectively. Results for solving the HEN retrofit problem in case study 2 by the proposed method are summarized in Table 5. They show that the proposed method is able to obtain a better objective function value by about 8.5%. This optimal solution incurs marginally higher operating cost but requires only half the investment compared to the reported solutions. This clearly shows that the proposed method can find a much better solution. Success rate of IDE to find the better solution is 60%, and the optimal network structure is found to be different in different runs. Average computation time for solving this problem is 13911 s.

Table 8.5. Comparison of Results for Case Study 2.

	Operating cost ($)	Investment ($)	Total annual cost ($/year)
Briones and Kokossis (1999)	571004	595842	690173
Ma et al. (2000)	571005	485139	668033
Rezaei and Shafiei (2009)	556630	547240	666078
This work	574400	288180	632036

Note: Operating cost in the existing network is $822100

Heat exchanger reassignments and area distribution after the retrofit by the proposed method and by the method in Rezaei and Shafiei (2009) are shown in Table 8.6. The total number of heat exchanger units used in the solution of Rezaei and Shafiei (2009) is 11 compared to 12 in this work. However, in the present solution, two exchangers (E11 and E13) did not change location and total additional area is just 949.6 m^2 whereas all heat exchangers have changed locations and total addition area is almost twice at 1818.14 m^2 in the solution of Rezaei and Shafiei (2009). Hence, as can be seen in Table 8.5, heat exchanger reassignments and area distribution after the retrofit by the proposed method require just about half the investment cost for the retrofit solution in Rezaei and Shafiei (2009).

Table 8.6. Heat Exchanger Reassignments and Area Distribution in Case Study 2.

Match number	Existing area (m^2)	Rezaei and Shafiei (2009)		This work	
		Retrofit area (m^2)	Exchanger assignment	Retrofit area (m^2)	Exchanger assignment
1	370	957.21	E2	508.6	E3
2	347	551.02	E6	485.9	E5
3	448	829.55	E1	680.7	E2
4	280	522.59	E7	287	E6
5	448	750.74	E3	759.1	E1
6	188	181.02	E5	211.8	E9
7	53	83.46	E10	114	E4
8	280	335.48	E4	278.9	E7
9	108	122.09	E8	123.5	E10
10	45	20.30	E11	66	E8
11	279	206.67	E9	228.6	E11
12	27	-	-	-	-
13	45	-	-	28.3	E13
Total additional area		1818.14		949.6	

8.4.3 Case study 3

Case study 3 is on retrofitting HEN for pre-heating crude in an existing atmospheric crude unit (Gadalla et al., 2003). It involves nine hot and three cold streams, and there are four stream splits in hot streams and one in cold streams. This example was also employed in the work of Chen (2008) as an example to demonstrate the application of HEN models. Stream and cost data for case study 3 are given in Table 8.7.

Table 8.7. Stream and Cost Data for Case Study 3 (Chen, 2008).

Streams	T^{in} (°C)	T^{out} (°C)	FCP (kW/ °C)	Cost ($/kW-year)
H_1	298	268	427.60	
H_2	339	100	211.13	
H_3	250	200	357.64	
H_4	257	50	25.10	
H_5	170	150	558.75	
H_6	282	40	83.88	
H_7	100	77	2081.09	
H_8	77	40	35.70	
H_9	189	40	41.52	
C_1	25	365	454.39	
C_2	271	282	798.45	
C_3	182	189	946.43	
Flue gas	1500	800		306.8
Water	10	40		5.25

Note: U = 0.5kW/(°C-m^2) for all exchangers; U = 0.667kW/(°C-m^2) for all heaters and U = 0.714 kW/(°C-m^2) for all coolers; capital cost ($) = 94093 + 1127$A^{0.9887}$ for each new exchanger of A m^2; capital cost ($) = 9665$\Delta A^{0.68}$ for additional area of ΔA in an existing exchanger; plant lifetime = 5 years; rate of interest = 0%; LMTD is used for area calculations.

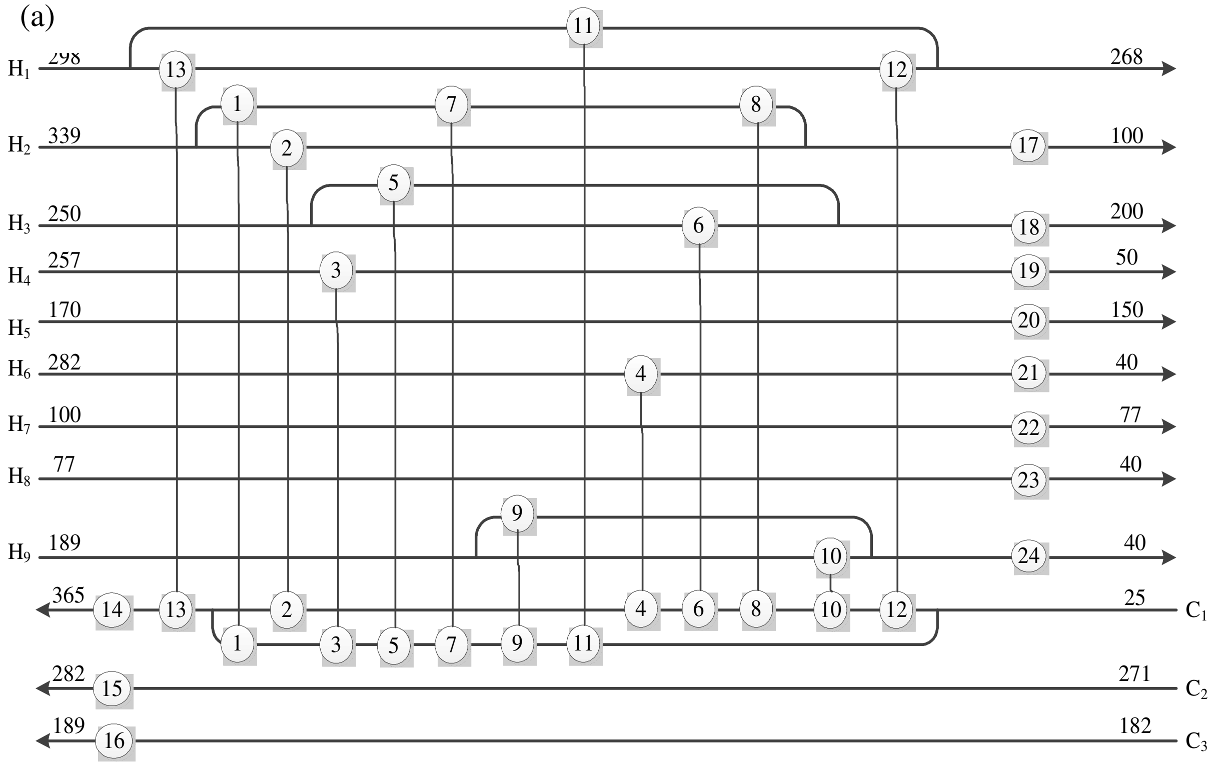
The existing crude oil pre-heat train of the refinery from Gadalla et al. (2003) is shown in Fig. 8.7(a). There are 13 heat exchangers, 3 heaters and 8 coolers in this network. The retrofitted network (Fig. 8.7b) using the proposed method has only 11 heat exchangers, 1 heater and 6 coolers. Furthermore, in this network, one of the stream splits is removed (hot stream 9 in Fig. 8.7b). Table 8.9 summarizes heat loads redistribution of the retrofitted HEN. Success rate of finding the better solution is 80%. However, the same network structure was not obtained in all runs. Number of integer and continuous variables in the optimization is 60 and 20 respectively. These continuous variables include heat exchanger areas and five streams split ratios which are on hot stream H1, H2, H3, H9 and cold stream, C1. Average computation time for solving this problem is 39086 s. Although this is quite large, it can be reduced by compiling the MATLAB code or using other software such as FORTRAN.

Recently, Smith et al. (2010) studied this HEN retrofit problem with variable FCP. Their results in Table 8.8 show that, even without HEN structure modification, the operating cost using variable FCP will decrease by 18%. With HEN structure modification, the operating cost for the variable FCP case will decrease by 23%. In the present work, we used constant FCP as in Gadalla et al. (2003). The optimization results for HEN retrofitting (last column in Table 8.8) show that operating cost reduces by 36%. This indicates that the proposed method finds a better HEN structure which significantly reduces utility consumption. However, the retrofitted structure requires more additional areas and so increased investment cost. Because of the large reduction in the operating cost, total annual cost of this work is significantly lower than others (Table 8.8).

Table 8.8. Comparison of Results for Case Study 3.

	Solutions in Smith et al. (2010)			This work
	Existing Network	Variable FCP (without modification)	Variable FCP (with modification)	Constant FCP (retrofitted)
Hot Utility (kW)	88,951	72,969	68,593	56,935
Cold Utility (kW)	92,300	76,288	71,908	60,265
Operating cost ($/year)	27,770,330	22,783,800	21,418,400	17,784,049
Additional area (m^2)	-	1334	1655	3794
Investment ($)	-	2,392,760	2,730,390	4,913,606
Total annual cost ($/year)	-	23,262,352	21,964,478	18,766,770

In order to investigate the effect of variable FCP of hot and cold streams, HEN structure of the retrofitted HEN (in Fig. 8.7b) obtained assuming constant FCP is used. With this structure and variable FCP of streams, local optimization using heat exchange areas and stream split ratios as decision variables was performed. This local and partial optimization reduces the operating cost to $20,884,430 and the investment cost is $4,527,344. This is because of the large FCP value for cold stream 1 at higher temperature which cause the higher heat load of the heat in the stream. The total annual cost of 21,789,899 ($/y) is higher than that using constant FCP, probably due to multiple minima even with only continuous variables and the inability of a local optimizer to find the global solution. However, this is better than the solution in Smith et al. (2010).



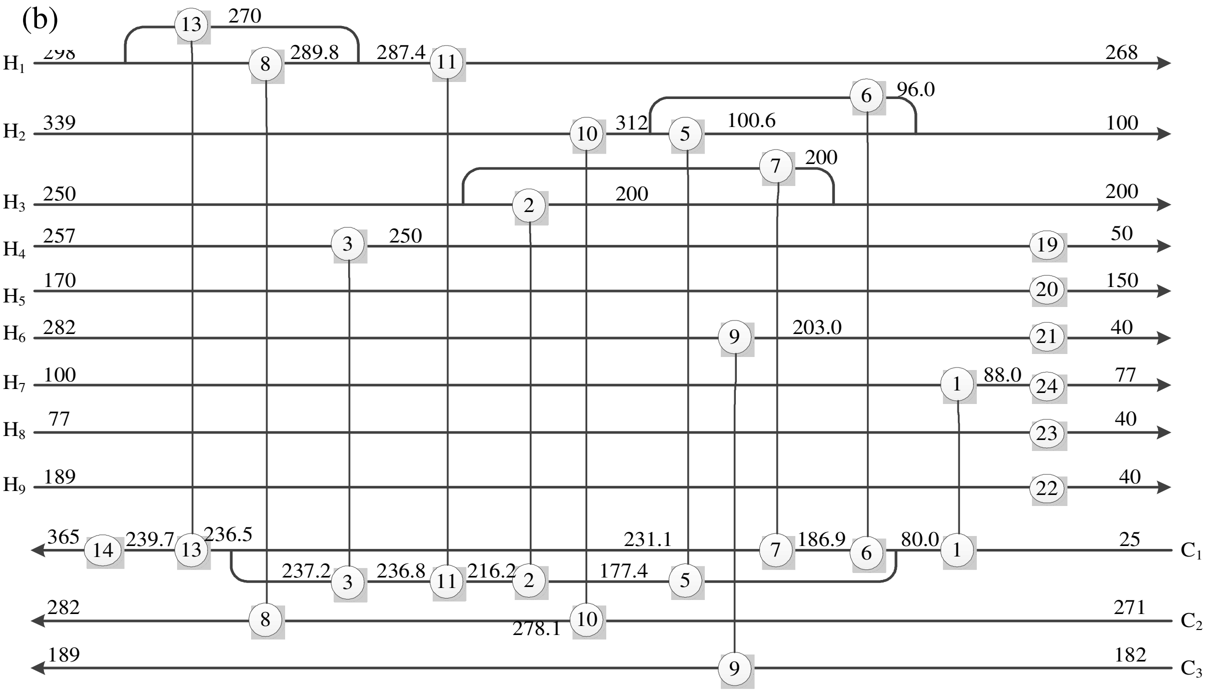


Figure 8.7. HEN in Case Study 3: (a) Original Network and (b) Retrofitted Network

Table 8.9. Heat Exchanger Reassignments and Heat Loads Distribution in Case Study 3.

Match number	Existing heat loads (MW)	From Smith et al. (2010)	This work
		Redistributed heat loads (MW)	Redistributed heat loads (MW)
1	13.25	11.87	24.99
2	13.19	14.12	15.62
3	0.86	0.93	0.18
4	0.11	10.1	-
5	7.47	16.29	39.26
6	7.33	1.63	5.49
7	11.42	13.74	2.27
8	11.52	10.76	3.11
9	0.89	1.7	1.45
10	2.05	2.91	5.67
11	2.37	1.32	8.30
12	1.74	3.87	-
13	8.76	7.64	6.63
14hu	73.54	57.61	56.93
15hu	8.78	8.78	-
16hu	6.63	6.63	-
17cu	1.09	-	-
18cu	3.08	-	-
19cu	4.33	4.26	5.02
20cu	11.18	11.18	11.18
21cu	20.19	10.2	6.19
22cu	47.87	47.86	22.89
23cu	1.32	1.32	1.32
24cu	32.44	1.58	13.67
Total heat Load (MW)	291.41	246.3	230.14

8.5 Conclusions

One-step optimization using the stochastic global optimizer, IDE is proposed for HEN retrofitting and tested it for 3 case studies. It optimizes the structure (discrete variables) and the heat exchanger areas and split ratios (continuous variables) simultaneously in one step. The new structure generated during IDE iterations is first be checked for feasibility before model calculations; thus, only feasible structures are evaluated. This approach not only increases the reliability of finding the better solution by preventing algorithm trapping at local optimum) but also increase the computational efficiency. Three examples from small to large-scale HEN retrofit problems with or without stream split are studied using IDE. The proposed one-step approach using IDE for HEN retrofit gave results better than those in the literature, and is very promising. It is planned to test this one-step approach for HEN retrofit problems with variable heat capacity in our future work.

Chapter 9

Conclusions and Recommendations

9.1 Conclusions of the Present Study

The stochastic global optimization algorithm, IDE is developed and its applications to chemical engineering processes have been studied in this doctoral thesis. In addition, adaptive constraint relaxation is proposed and studied for optimization problems with equality constraints. The major contributions and conclusions of this study are summarized in this section.

1. A review of global optimization algorithms with their applications in thermodynamic modeling is presented. Selected deterministic and stochastic global optimization algorithms are briefly described, and then their use for phase stability analysis, Gibbs free energy minimization and parameter estimation in phase equilibrium models, is reviewed. In short, a general overview of global optimization for modeling the phase behavior of systems with and without chemical reactions is presented.
2. The IDE algorithm, which integrates DE with tabu list of tabu search, self-adaptive strategies and a novel stopping criterion based on the number of rejected points, is developed. The effectiveness of the proposed stopping criterion and IDE is assessed on a wide range of benchmark functions. The performance of IDE is compared with state-of-the-art stochastic global optimization algorithms recently published in the literature. The results show that the reliability of finding the global optimum by IDE is better than many

other stochastic global optimizers, and that number of function evaluations required by IDE is significantly reduced due to the proposed stopping criterion.

3. IDE is used to solve parameter estimation problems such as in dynamic models and VLE data modeling. The performance of IDE for these problems is compared with that of DE, DETL, PSO, SA and a deterministic algorithm, BARON. IDE is shown to be better than or comparable to these algorithms for parameter estimation in both dynamic systems and VLE data modeling.
4. Phase equilibrium calculations and phase stability analysis are challenging problems due to the high non-linearity of thermodynamic models. IDE with and without tabu list, both with SCmax stopping criterion, are used for solving these problems. The performance of these stochastic algorithms is compared to ascertain their relative strengths for phase equilibrium and phase stability problems including without and with chemical reactions. Overall, IDE with tabu list has better performance for the phase equilibrium, chemical equilibrium and phase stability problems.
5. A novel constraint handling method, which incorporates adaptive constraint relaxation with feasibility approach, is proposed for constrained problems. In this, both the equality and inequality constraints are relaxed in order to increase temporarily the feasible region in the initial exploration. The feasible region is gradually shrunk according to the fraction of feasible individuals in the population. The proposed constraint handling technique with IDE is tested for solving benchmark problems with constraints, and subsequently applied to many chemical engineering problems with equality and inequality constraints. The results show that the proposed constraint handling method with IDE is

reliable and efficient for solving constrained optimization problems, even with equality constraints.

6. The pooling problem is an important optimization problem in petroleum industry. Owing to the bi-linear terms in the formulation, this is a non-convex optimization problem with multiple minima. The application of IDE to many standard pooling problems from the literature is investigated, for the first time by a stochastic optimizer. The results demonstrate that IDE algorithm is a good alternative for solving pooling problems.
7. IDE algorithm is modified for handling both integer and continuous variables, and then applied to solve HEN retrofitting problems by one-step approach, where both discrete and continuous variables are simultaneously optimized. This single-step approach not only avoids the algorithm trapping at a local optimum but also improves the computational efficiency. Three examples from small to large-scale HEN retrofit problems with and without stream split are studied. The results clearly show that the proposed one-step approach for HEN retrofit is very promising.

9.2 Recommendations for Future Work

Stochastic global optimization is an active research area and there is scope for many studies as well as its applications. Some of possible works for future study have been identified below.

Development of more efficient and reliable global optimization algorithms: Studies indicate that both deterministic and stochastic global optimization methods require further improvement for solving application problems robustly and efficiently. Compared to deterministic methods, stochastic optimization techniques involve simple concepts, do not require any assumptions and can be used for any type of problem. Hybridization to synergize selected features of different stochastic algorithms is a promising approach for developing better algorithms since reported results show that the performance of pure algorithms is almost always inferior to that of hybrid algorithms. Hence, further studies should be focused on the development of hybrid strategies to improve the reliability and efficiency of stochastic optimization methods. In addition, alternative termination criteria should be studied and tested for reliably determining the convergence of stochastic optimizers (Shoen, 2009). It is also desirable that these methods should have no or fewer tuning parameters.

Even though stochastic global optimizers have been successfully applied in many fields, most of the proposed algorithms have been tried and shown to be effective for small to medium size problems. Stochastic methods also face the “curse of dimensionality”, which means that the performance of the algorithm decreases significantly as the problem size increases (Wang, 2008). Several researchers have focused on developing stochastic algorithms for solving large global optimization problems (Wang et al., 2007; Yang et al., 2008; Cui et al., 2008; Yang et al., 2009). Hence, testing and improving IDE for such problems is important and should be studied.

Solution of large global optimization problems: Stochastic global optimization algorithms have been successfully employed in many fields, from science to engineering, due to their robustness, simple concept and easy to implement. But most of the proposed algorithms have been tried and shown to be effective for small to medium size problems (i.e., number of decision variables from 2 to 50). Stochastic methods also face the “curse of dimensionality”; this means that the performance of the algorithm decreases significantly as the problem size increases

(Wang, 2008). Optimization problems for some applications can contain a large number of variables; for example, Schulz et al. (2005) used DICOPT (a solver in GAMS) to solve supply chain optimization of large scale continuous processes with more than thousand variables and constraints. Although GAMs/DICOPT can handle high dimension and non-convex optimization problems, it is just a local method. Hence, development of stochastic global methods to handle large optimization problems is an important and challenging task.

Several researchers have focused on developing stochastic algorithms for solving large global optimization problems. Wang et al. (2007) proposed a hybrid method which combines SA with a gradient-based method for large global optimization problems. In addition, the "stretching" technique is incorporated in this hybrid method in order to achieve faster convergence. Results show that their algorithm has high success rate and high solution precision. Cui et al. (2008) proposed a PSO with fitness uniform selection strategy and random walk strategy. This proposed algorithm was tested on seven benchmark functions with dimension up to 3000. Their results show the good performance of the proposed algorithm compared to two other variants of PSO (namely, standard PSO and attractive & repulsive PSO). Yang et al. (2008) proposed two DE algorithms, named DECC-1 and DECC-2, for problems up to 1000 variables; the algorithms are based on the cooperative co-evolution framework. Their results show that these proposed algorithms have better performance on benchmark functions. Subsequently, a high dimension global optimization algorithm, JACC-G was proposed by Yang et al. (2009). This algorithm uses DE combined with cooperative co-evolution strategy and self-adaptive parameter strategy. Their results show the efficiency improvement for large problems compared to other algorithms. So, cooperative co-evolution and/or hybridization strategies should be investigated for solving large global optimization problems, in the near future.

IDE for multi-objective optimization: Many real world problems involve several objectives which are conflicting in nature. It is often difficult to formulate these problems into a single objective optimization problem. Solution of such problems is known as multi-objective optimization (MOO). The conflicting objectives in MOO problems lead to a set of optimal solutions called the Pareto-optimal solutions, which are equally good for the specified objectives and none of them can be said to be better than another without additional information about the problem. These solutions provide better understanding of the trade-off among objectives and also more choices to the decision-maker to choose a particular solution for implementation. MOO problems become even more complex if the objective functions are highly non-linear and non-convex. Over the last two decades, MOO field has grown significantly and many chemical engineering applications of it have been reported (Masuduzzaman and Rangaiah, 2009). Although IDE has only been used for single objective optimization, it has the capability of solving MOO problems too. Therefore, study of IDE for MOO problems is one of the interesting and recommended works.

IDE for generalized pooling and water management problems: The generalized pooling problem can model diverse applications in refineries, chemical plants and water treatment facilities (Misener et al., 2010). It differs from the regular pooling problem in that it includes connections/flows between the pools. Misener et al. (2010) and Faria and Bagajewicz (2011) applied the generalized pooling problem approach to a complex industrial wastewater treatment system using *p*-formulation, and solved the problem by deterministic global optimization methods. The IDE in

Chapter 7 has been successfully applied to pooling problems. Therefore, solving generalized pooling problems with stochastic global optimization algorithm, IDE, will be one of the potential future works.

IDE to HEN retrofitting problems with varying heat capacity: Several researchers have studied HEN retrofitting problems with constant heat capacities, as reviewed in Chapter 8. However, the thermal properties are highly dependent on temperature for some streams. Thus, varying heat capacities often arise when multi-component streams are cooled or heated, such as in refining preheat trains. Smith et al. (2011) presented a methodology which uses network pinch approach with structure modifications and cost optimization to solve HEN retrofitting problems with varying FCP. The proposed one-step approach using IDE for HEN retrofitting with constant FCP has shown good reliability and efficiency (see Chapter 8). Therefore, this approach with IDE should be investigated further for HEN retrofitting problems with varying FCP values.

References

Adhya, N.; Tawarmalani, M.; Sahinidis, N. V., A Lagrangian approach to the pooling problem. *Ind. Eng. Chem. Res.***,** vol. 38, pp. 1965-1972, 1999.

Angira, R. and Babu, B.V. Optimization of process synthesis and design problems: A modified differential evolution approach. *Chem. Eng. Sci.*, 61, 4707-4721, 2006.

Angira, R. and Santosh, A. Optimization of dynamic systems: A trigonometric differential evolution approach. *Comp. & Chem. Eng.*, 31, 1055-1063, 2007.

Ali, M.M., Torn, A. and Viitanen, S., Numerical comparison of some modified controlled random search algorithms. *J. Glob. Optim.,* vol. 11, pp. 377-385, 1997.

Ali, M. M.; Kajee-Bagdadi, Z., A local exploration-based differential evolution algorithm for constrained global optimization. *Appl. Math. Comput.* **2009,** *208*, 31-48.

Almutairi, H. and Elhedhli, S., A new Lagrangean approach to the pooling problem. *J. Global Opt.***,** vol.*45*, pp.237-257, 2009.

Alvarez, V.H., Larico, R., Ianos, Y. and Aznar, M., Parameter estimation for VLE calculation by global minimization: the genetic algorithm. *Braz. J. Chem. Eng.,* vol. 25, pp. 409-418, 2008.

Asante, N.D.K., Zhu, X.X., An Automated and Interactive Approach for Heat Exchanger Network Retrofit. *Chem. Eng. Res. and Design*, 75, 349-360, 1997.

Aslam, N. and Sunol, A.K., Reliable computation of binary homogeneous azeotropes of multi-component mixtures at higher pressures through equation of states. *Chem. Eng. Sci.*, vol. 59, pp. 599-609, 2004.

Aslam, N. and Sunol, A.K., Sensitivity of azeotropic states to activity coefficient model parameters and system variables. *Fluid Phase Equilib.*, vol. 240, pp. 1-14, 2006.

Audet, C.; Brimberg, J.; Hansen, P.; Digabel, S. b. L.; Mladenovi, N., Pooling Problem: Alternate Formulations and Solution Methods. *Manage. Sci.***,** vol. *50* (6), pp. 761-776, 2004.

Babu, B., and Angira, R., Modified differential evolution (MDE) for optimization of non-linear chemical processes. *Comp. & Chem. Eng.,* 30, pp.989-1002, 2006.

Babu, B.; Munawar, S. A., Differential evolution strategies for optimal design of shell-and-tube heat exchangers. *Chem. Eng. Sci. 62*, 3720-3739, 2007.

Babu, K. S., Kumar, M.V.P. and Kaistha, N., Controllable optimized designs of an ideal reactive distillation system using genetic algorithm. *Chem. Eng. Sci.,* vol. 64, pp. 4929-4942, 2009.

Bagajewicz, M., A review of recent design procedures for water networks in refineries and process plants. *Comput. & Chem. Eng.,* vol.24, pp. 2093-2113, 2000.

Baker, J. E. Reducing Bias and Inefficiency in the Selection Algorithm, *in Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pp 14-21, 1987.

Baker, T. E. and Lasdon, L. S., Successive Linear Programming at Exxon. *Manage. Sci.* vol 31, pp.264-274, 1985.

Balogh, J., Csendes, T. and Stateva, R.P., Application of a stochastic method to the solution of the phase stability problem: cubic equations of state. *Fluid Phase Equilib*., vol. 212, pp. 257-267, 2003.

Balsa-Canto, E., Peifer, M., Banga, J.R., Timmer, J. and Fleck, C., Hybrid optimization method with general switching strategy for parameter estimation. *BMC Systems Biology,* vol. 2, p. 26, 2008.

Barbaro, A. and Bagajewicz, M., New regorous one-step MILP formulation for heat exchanger network synthesis. *Comp. & Chem. Eng*., 29, pp. 1945-1976, 2005.

Ben-Tal, A.; Eiger, G. and Gershovitz, V., Global minimization by reducing the duality gap. *Math. Program.*, vol.*63*, pp.193-212, 1994.

Bergamini, M.L., Scenna, N.J., Aguirre. P.A., Global optimal structures of heat exchanger networks by piecewise relaxation. *Ind. Eng. Chem. Res.*, 46, pp. 1752-1763, 2007.

Blum, C. and Roli, A., Metaheuristics in combinatorial optimization: overview in conceptual comparison. *ACM Comp. Surv.*, vol. 35, pp. 268-308, 2003.

Bochenek, R., and Jezowski, J.M., Genetic Algorithms Formulation for Retrofitting Heat Exchanger Network, in: Rangaiah, G.P. (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*. World Scientific Pub Co Inc, pp. 545-589, 2010.

Bollas, G.M., Barton, P.I. and Mitsos, A., Bilevel optimization formulation for parameter estimation in vapor-liquid(-liquid) phase equilibrium problems. *Chem. Eng. Sci.*, vol. 64, pp. 1768-1783, 2009.

Bonilla-Petriciolet, A., Vazquez-Roman, R., Iglesias-Silva, G.A. and Hall, K.R., Performance of Stochastic Global Optimization Methods in the Calculation of Phase Stability Analyses for Nonreactive and Reactive Mixtures. *Ind. Eng. Chem. Res.*, vol. 45, pp. 4764-4772, 2006.

Bonilla-Petriciolet, A., Bravo-Sanchez, U.I., Castillo-Borja, F., Zapiain-Salinas, J.G. and Soto-Bernal, J.J., The performance of Simulated Annealing in parameter estimation for vapor-liquid equilibrium modeling. *Braz. J. Chem. Eng.,* vol. 24, pp. 151-162, 2007.

Bonilla-Petriciolet, A., Iglesias-Silva, G.A. and Hall, K.R. Calculation of homogeneous azeotropes in reactive and non-reactive mixtures using a stochastic optimization approach. *Fluid Phase Equilib.,* vol. 281, pp. 22–31, 2009.

Bonilla-Petriciolet, A. and Segovia-Hernandez, J. G., A comparative study of particle swarm optimization and its variants for phase stability and equilibrium calculations in multicomponent reactive and non-reactive systems. *Fluid Phase Equilib.,* vol. 289, pp. 110-121, 2010.

Bonilla-Petriciolet, A., Rangaiah, G.P. and Segovia-Hernández, J.G., Evaluation of stochastic global optimization methods for modeling vapor-liquid equilibrium data. *Fluid Phase Equilib*. vol. 287, pp. 111-125, 2010a.

Bonilla-Petriciolet, A., Segovia-Hernández, J.G. and Soto-Bernal, J.J., Harmony search for parameter estimation in vapor-liquid equilibrium modeling", *In Proceedings of 5th International Symposium on Design, Operation and Control of Chemical Processes*, pp. 719-726, 2010b.

Bonilla-Petriciolet, A., Rangaiah, G.P., Segovia-Hernandez, J.G. and Jaime-Leal, J.E., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 413-463, 2010c.

Bonilla-Petriciolet, A., Rangaiah, G.P. and Segovia-Hernández, J.G., Constrained and unconstrained Gibbs energy minimization in reactive systems using genetic algorithms and differential evolution with tabu list. *Fluid Phase Equilib*., vol. 300, pp. 120-134, 2011.

Briones, V., Kokossis, A. C., Hypertargets: A Conceptual Programming approach for the optimization of industrial heat exchanger networks. II. Retrofit design. *Chem. Eng. Sci.*, 54, pp. 541–561, 1999.

Brest, J., Greiner, S., Boskovic, B., Merjan, M. and Zumer, V. Self-adaptive control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comp.*,10(6), 646-657, 2006.

Brest, J., Zamuda, A., Boskovic, B., Maucec, M.S. and Zumer, V. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. *IEEE Congress on Evolutionary Computation*, 2032-2039. 2008.

Brooks, S., A discussion of random methods for seeking maxima. *Oper. Res.,* vol. 6, pp. 244-251, 1958.

Burgos-Solorzano, G.I., Brennecke, J.F., Stadtherr, M.A., Validated computing approach for high-pressure chemical and multiphase equilibrium. *Fluid Phase Equilib*., vol. 219, pp. 245-255, 2004.

Cetin, B. C., Barhen, J. and Burdick, W.J., Terminal Repeller Unconstrained Subenergy Tunneling for Fast Global Optimization. *J. Optimiz. Theory App.,* vol. 77, pp. 97-126, 1993.

Chaikunchuensakun, S., Stiel, L. I. and Baker, E.L., A Combined Algorithm for Stability and Phase Equilibrium by Gibbs Free Energy Minimization", *Ind. Eng. Chem. Res.,* vol. 41, pp. 4132-4140, 2002.

Chen, L., Heat-integrated crude oil distillation system design, *PhD Thesis*. The University of Manchester, UK, 2008.

Chen, S.Q., Rangaiah, G.P. and Srinivas, M., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, 2010.

Cheung, A., Adjiman, C.S., Kolar, P. and Ishikawa, T., Global optimization for clusters of flexible molecules-solvent-solute interaction energy calculations. *Fluid Phase Equilib*., vol. 194, pp. 169–183, 2002.

Chibante, R., *Simulated Annealing, Theory with applications,* Intech, India, 2010.

Ciric, A.R., Floudas, C.A., A retrofit approach for heat exchanger networks. *Comp.s & Chem. Eng.*, 13, pp. 703-715, 1989.

Ciric, A.R., Floudas, C.A., A comprehensive optimization model of the heat exchanger network retrofit problem. *Heat Recovery Systems and CHP*, 10, pp. 407-422,1990.

Coello Coello, C. A., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comp. Meth. Appl. Mech. & Eng.,* 191, pp.1245-1287, 2002.

Cordero, J.C., Davin, A., Floquet, P., Pibouleau, L. and Domenech, S., Synthesis of optimal reactor networks using mathematical programming and simulated annealing. *Comp. & Chem. Eng.,* vol. 21, pp. 47-52, 1997.

Costa, A.L.H., Silva, F.P.T. and Pessoa, F.L.P., Parameter estimation of thermodynamic models for high-pressure systems employing a stochastic method of global optimization. *Braz. J. Chem. Eng*., vol. 17, pp. 349-353, 2000.

Cui, Z., Cai, X., Zeng, J. and Sun, G., Particle swarm optimization with FUSS and RWS for high dimensional functions. *Appl. Math. Comp.*, 205, pp.98-108, 2008.

Das, H., Cummings, P. T., and Levan, M. D., Scheduling of serial multiproduct batch processes via simulated annealing. *Comp.s & Chem. Eng.,* 14, pp.1351-1362, 1990.

Deb, K., An efficient constraint handling method for genetic algorithms. *Comp. Meth. Appl. Mech. & Eng.,* 18, pp.311-338, 2000.

Dominguez, A., Tojo, J. and Castier, M., Automatic implementation of thermodynamic models for reliable parameter estimation using computer algebra. *Comp. & Chem. Eng*., vol. 26, pp. 1473-1479, 2002.

Dorigo, M., Maniezzo, V. and Colorni, A., Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man. Cybernet. Part B*, vol. 26, pp. 29-41, 1996.

Dumas, L.; Druez, B.; Lecerf, N., A fully adaptive hybrid optimization of aircraft engine blades. *Comp.Appl. Math. 232*, 54-60, 2009.

Englezos, P. and Kalogerakis, N., *Applied parameter estimation for chemical engineers*, Marcel-Dekker, New York, 2001.

Esposito, W.R. and Floudas, C.A., Global optimization in parameter estimation of nonlinear algebraic models via the error-in-variables approach. *Ind. Eng. Chem. Res.*, vol. 37, pp. 1841-1858, 1998.

Esposito, W. R. and Floudas, C. A. Global optimization for the parameter estimation of differential algebraic systems. *Ind. Eng. Chem. Res.*, 39, 1291–1310, 2000.

Exler, O., Antelo, L. T., Egea, J. A.; Alonso, A. A.; Banga, J. R., A Tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design. *Comp. & Chem. Eng.***,** *32*, 1877-1891. 2008.

Faria, D.C., and bagajewicz, M.J., A new approach for global optimization of a class of MINLP problems with applications to water manement and pooling problems, *AIChE J.*, In press, 2011.

Fernandez-Vargas, J.A., *Phase equilibrium modeling in non-reactive systems using ant colony optimization*. Thesis of Master, Universidad de Guanajuato, 2011.

Ferrari, J.C., Nagatani, G., Corazza, F.C., Oliveira, J.V. and Corazza, M.L., Application of stochastic algorithms for parameter estimation in the liquid-liquid phase equilibrium modeling. *Fluid Phase Equilib.*, vol. 280, pp. 110-119, 2009.

Fidkowski, Z.T., Malone, M.F. and Doherty, M.F., Computing azeotropes in multicomponent mixtures", *Comp. & Chem. Eng.*, vol. 17, pp. 1141-1155, 1993.

Floudas, C. A., Aggarwal, A., and Ciric, A. R., Global optimum search for non-convex NLP and MINLP problems. *Comp. & Chem. Eng.,* 13, pp.1117-1132, 1989.

Floudas, C. A.; Aggarwal, A., A decomposition strategy for global optimum search in the pooling problem. *ORSA J.Comp.***,** vol. 2, pp. 225-235, 1990.

Floudas, C.A., *Deterministic global optimization: Theory, methods and applications*, Kluwer Academic, Boston, 1999.

Floudas, C. A.; Akrotirianakis, I. G.; Caratzoulas, S.; Meyer, C. A.; Kallrath, J., Global optimization in the 21st century: Advances and challenges. *Comp. & Chem. Eng.***,** vol. *29*, pp. 1185-1202, 2005.

Floudas, C. A.; Gounaris, C. E., A review of recent advances in global optimization. *J. Glob. Optim.***,** vol. 45, pp. 3-38, 2009.

Foulds, L. R.; Haugland, D.; Jornsten, K., A Bilinear Approach to the Pooling Problem. *Optim.* **,** vol. 24, pp. 165-180, 1992.

Freitas, L., Platt, G. and Henderson, N., Novel approach for the calculation of critical points in binary mixtures using global optimization. *Fluid Phase Equilib*., vol. 225, pp. 29-37, 2004.

Furman, K. C., and Sahinidis, N. V., Computational complexity of heat exchanger network synthesis. *Comp. & Chem. Eng.*, vol.*25*(9–10), pp.1371–1390, 2001.

Gadalla, M., Jobson,M., Smith, R., Optimization of existing heat-integrated refinery distillation systems, *Trans IChemE,* 81, pp. 147-152, 2003.

Gau, C.Y., Brennecke, J.F. and Stadtherr, M.A., Reliable nonlinear parameter estimation in VLE modeling. *Fluid Phase Equilib*., vol. 168, pp. 1-18, 2000.

Gau, C.Y. and Stadtherr, M.A., Deterministic global optimization for error-in-variables parameter estimation. *AIChE J.*, vol. 48, pp. 1192-1197, 2002.

Gecergormez, H. and Demirel, Y., Phase stability analysis using interval Newton method with NRTL model. *Fluid Phase Equilib.*, vol. 237, pp. 48-58, 2005.

Geem, Z.W., Kim, J.H. and Loganathan, G.V., A new heuristic optimization algorithm: harmony search. *Simulation*, vol. 76, pp. 60-68, 2001.

Geem, Z.W., *Music-inspired harmony search algorithm theory and applications*, First Ed., Springer, 2009.

Glover, F. and Laguna, M., *Tabu search*, Kluwer Academic Publishers, 1997.

Golubitsky, M. and Schaeffer, D.G., *Singularities and groups in bifurcation theory*, Springer-Verlag, New York, 1985.

Gounaris, C. E.; Misener, R.; Floudas, C. A., Computational Comparison of Piecewise-Linear Relaxations for Pooling Problems. *Ind. Eng. Chem. Res.***,** vol. 48, pp. 5742-5766, 2009.

Guus, C.; Boender, E.; Romeijn, H. E., *Stochastic methods*. Kluwer Academic Publishers: 1995.

Hanke, M. and Li, P., Simulated annealing for the optimisation of batch distillation processes. *Comp. & Chem. Eng.,* vol. 24, pp. 1-8, 2000.

Hansen, E.R., *Global optimization using interval analysis*, Marcel Dekker, New York, 1992.

Hansen, N. and Kern, S. "Evaluating the CMA evolutionary strategy on multimodel test functions," Springer-Verlag, 2004. http://www.springer.de/comp/lncs/index.html.

Harding, S.T., Maranas, C.D., McDonald, C.M. and Floudas, C.A., Locating all homogeneous azeotropes in multicomponent mixtures. *Ind. Eng. Chem. Res*., vol. 36, pp. 160-178, 1997.

Harding, S.T. and Floudas, C.A., Phase Stability with Cubic Equations of State: A Global Optimization Approach. *AIChE J.***,** vol.*46* (7), pp.1422-1440, 2000a.

Harding, S.T. and Floudas, C.A., Locating all heterogeneous and reactive azeotropes in multicomponent mixtures. *Ind. Eng. Chem. Res*., vol. 39, pp. 1576-1595, 2000b.

Haverly, C. A., Studies of the Behaviour of Recursion for the Pooling Problem. *ACM SIGMAP Bull.***,** vol.*25*, pp.29-32, 1978.

He, Q.; Wang, L., A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl. Math. & Comp.***,** vol. 186, pp. 1407-1422, 2007.

Heidemann, R.A. and Khalil, A.M., The calculation of critical points. *AIChE J*., vol. 26, pp. 769-779, 1980.

Henderson, N., Freitas, U. and Platt, G.M., Prediction of critical points: a new methodology using global optimization. *AIChE J.***,** vol. 50, pp. 1300–1314, 2004.

Henderson, N., Sacco, W.F., Barufatti, N.E. and Ali, M.M., Calculation of critical points of thermodynamic mixtures with differential evolution algorithms. *Ind. Eng. Chem. Res*., vol. 49, pp. 1872-1882, 2010.

Holland, J.H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, 1975.

Horst, R. and Tuy, H., *Global optimization: Deterministic approaches*. Springer: Berlin, 1996.

Hua, J.Z., Brennecke, J.F. and Stadtherr, M.A. Enhanced Interval Analysis for Phase Stability: Cubic Equation of State Models. *Ind. Eng. Chem. Res*., vol. 37, pp. 1519-1527, 1998.

Huang, V. L., Qin, A. K., and Suganthan, P. N., Self-adaptive Differential Evolution Algorithm for Constrained Real-Parameter Optimization. *In 2006 IEEE Congress on Evolutionary Computation*. Vancouver, BC, Canada, 2006.

Jalali, F., Seader, J.D. Homotopy continuation method in multi-phase multi-reaction equilibrium systems. *Comp. & Chem. Eng*., vol.23, pp.1319-1331, 1999.

Jalali, F., Seader, J.D. and Khaleghi, S., Global solution approaches in equilibrium and stability analysis using homotopy continuation in the complex domain. *Comp. & Chem. Eng*., vol. 32, pp. 2333-2345, 2008.

Jayaraman, V. K., Kulkarni, B. D., Karale, S. and Shelokar, P., Ant colony frame work for optimal design and scheduling of batch plants. *Comp. & Chem.Eng.,* 24, pp.1901-1912, 2000.

Jayaraman, V.K., Shelokar, P.S., Shingade, P., Pote, V., Baskar, R. and Kulkarni, B.D., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 237-270, 2010.

Jezowski, J., Bochenek, R. and Ziomek, G., Random search optimization approach for highlymulti-model nonlinear problems. *Adv. Eng. Softw.,* vol. 36, pp. 504-517, 2005.

Jezowski,J., Bochenek, R. and Poplewski, G., On application of stochastic optimization techniques to designing heat exchanger and water networks, *Chem. Eng. Proc.*, 46, pp. 1160-1174, 2007

Jezowski, J., Poplewski, G. and Bochenek,R., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 57-110, 2010.

Jiang, H., Cai, W. and Shao, X., A random tunneling algorithm for the structural optimization problem. *Phys. Chem. Chem. Phys*., vol. 4, pp. 4782-4788, 2002.

Jourdan, L., Basseur, M. and Taibi, E.G., Hybridizing exact methods and metaheuristics: A taxonomy. *Eur. J. Oper. Res.,* vol. 199, pp. 620-629, 2009.

Junior, N.S., Cardozo, N.S.M., Secchi, A.R. and Wada, K., Adaptive Random Search: A Promising Method for Determining the Stability of Mixtures. *Comp. Aided Chem. Eng.*, vol. 27, pp. 321-326, 2009.

Justo-García, D.N., García-Sánchez, F., Águila-Hernández, J. and Eustaquio-Rincón, R., Application of the simulated annealing technique to the calculation of critical points of multicomponent mixtures with cubic equations of state. *Fluid Phase Equilib*., vol. 264, pp. 164-173, 2008.

Kapadi, M. D. and Gudi, R. D. Optimal control of fed-batch fermentation involving multiple feeds using differential evolution. *Process Biochemistry*, 39, 1709-1721, 2004.

Katare, S., Bhan, A., Caruther, J.M., Delgass, W.N., and Venkatasubramanian, V. A hybrid genetic algorithm for efficient parameter estimation of large kinetic models. *Comp. & Chem. Eng.*, 28, 2569-2581, 2004.

Kemp I.C., *Pinch Analysis and Process Integration: A User Guide on Process Integration for the Efficient Use of Energy*, Elsevier, 2007.

Kennedy, J. and Eberhart, R.C., Particle Swarm Optimization. *In Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

Kennedy, J., Eberhart, R.C. and Shi, Y., *Swarm Intelligence,* Academic Press, 2001.

Kheawhom, S., Efficient constraint handling scheme for differential evolutionary algorithm in solving chemical engineering optimization problem. *Ind. Eng. Chem. Res.*, 16, pp.620-628, 2010.

Khorasany, R.M., Fesanghary, M., A novel approach for synthesis of cost-optimal heat exchanger networks, *Comp. & Chem. Eng.*, 33, pp. 1363-1370, 2009.

Kirkpatrick, S., Gelatt, C. and Vecci, M., Optimization by Simulated Annealing. *Science*, vol. 220, pp. 671-680, 1983.

Kocis, G. R. and Grossmann, I. E., Global optimization of non-convex mixed-integer non-linear programming (MINLP) problems in process synthesis. *Ind. Eng. Chem. Res.*, 27, pp.1407-1421, 1998.

Kotjabasakis, E., Linnhoff, B., Sensitivity tables for the design of flexible processes (1) — How much contingency in heat exchanger networks is cost effective? *Chem. Eng. Res. and Design*, 64, pp. 197–211, 1986.

Kundu, M., Chitturi, A. and Bandyopadhyay, S.S., Prediction of equilibrium solubility of CO_2 in aqueous alkanolamines using differential evolution algorithm. *Can. J. Chem. Eng.*, vol. 86, pp. 177-126, 2008.

Lampinen, J., A constraint handling approach for the differential evolution algorithm, *In Proceedings of the Congress on Evolutionary Computation*, Washington,DC, 2002.

Lasdon, L. S.; Waren, A. D.; Sarkar, S.; Palacios-Gomez, F., Solving the Pooling Problem using Generalized Reduced Gradient and Successive Linear Programming Algorithms. *ACM SIGMAP Bull.*, vol. 27,pp. 9-15, 1979.

Lazzus, J.A. Optimization of activity coefficient models to describe vapor-liquid equilibrium of (alcohol+water) mixtures using a particle swarm algorithm. *Comp. Math. Appl.*, vol. 60, pp. 2260-2269, 2010.

Lee, Y.P., Rangaiah, G.P. and Luus, R., Phase and chemical equilibrium calculations by direct search optimization. *Comp.& Chem. Eng.,* vol. 23, pp. 1183-1191, 1999.

Lee, E.S.-Q., Rangaiah, G. P. and Agrawal, N., Optimization design of chemical processes for multiple economic and environmental objectives. G. P. Rangaiah

(Ed.), *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering* , pp.301-337. Singapore: World Scientific.*2010.*

Leguizamon, G. and Coello-Coello, C. A. C., Boundary Search for Constrained Numerical Optimization Problems With an Algorithm Inspired by the Ant Colony Metaphor. *IEEE Trans. Evol. Comp.,* 13, pp.350-368, 2009.

Levy, A.V. and Gomez, S., in: Boggs, P.T., Byrd, R.H. and Schnabel, R.B., (Eds.), *Numerical Optimization,* SIAM, pp. 213-244, 1985.

Li, B.H. and Chang, C.T., Retrofitting heat exchanger networks based on simple pinch analysis. *Ind. Eng. Chem. Res*., 49, pp. 3967-3971, 2010.

Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P., Coello Coello, C. A. and Deb, K., Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. *In Technical Report.* Singapore: Nanyang Technological University, 2006.

Liang, J. J. and Suganthan, P. N., Comparison of results on the 2006 CEC benchmark function set. In. http://www3.ntu.edu.sg/home/EPNSugan/.

Liao, T.W. Two hybrid differential evolution algorithms for engineering design optimization. *Appl. Soft Comp.*, 10, pp.1188-1199, 2010.

Liberti, L.and Kucherenko, S., Comparison of deterministic and stochastic approaches to global optimization. Int. Trans. in Operat. Res., 12 (3), pp. 263-286. 2005.

Lin, B. and Miller, D. C., Tabu search algorithm for chemical process optimization. *Comp. & Chem. Eng.***,** *28*, pp. 2287-2306. 2004.

Lin, K. and Chen, D., Hybrid artificial immune system and its application to calculation of chemical reaction and phase equilibrium. *J. Chem. Ind. Eng.,* vol. 58, pp. 1348-1352, 2007.

Liu, H., Cai, Z. and Wang, Y., Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl. Soft Comp.,* 10, pp.629-640, 2010.

Liu, P.K. and Wang, F.S., Hybrid differential evolution with geometric mean mutation in parameter estimation of bioreaction systems with large parameter search space. *Comp. & Chem. Eng.,* vol. 33, pp. 1851-1860, 2009.

Lucia, A., Padmanabhan, L. and Venkataraman, S., Multiphase equilibrium flash calculations. *Comp. & Chem. Eng*., vol. 24, pp. 2557-2569, 2000.

Lucia, A., DiMaggio, P.A., Bellows, M.L. and Octavio, L.M., The phase behavior of n-alkane systems. *Comput. & Chem. Eng*., vol. 29, pp. 2363–2379, 2005.

Lucia, A.; Gattupalli, R. R.; Kulkarni, K.; Linninger, A., A barrier-terrain methodology for global optimization. *Ind. Eng. Chem. Res. 47*, pp. 2666-2680, 2008.

Luo, Y., Yuan, X. and Liu, Y., An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints. *Comp. & Chem. Eng.,* 31, pp.153-162, 2007.

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

Luus, R. and Brenek, P., Incorporation of gradient into random search optimization. *Chem. Eng. Tech.,* vol. 12, pp. 309-318, 1989.

Luus, R., in: Floudas, C.A. and Pardalos, P.M., (Eds.), *Encyclopedia of Optimization,* Kluwer Academic Publishers, pp. 440-444, 2001.

Luus, R., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 17-56, 2010.

Ma, K. L., Hui, C. W., Yee, T. F., Constant approach temperature model for HEN retrofit. *Appl. Therm. Eng.*, *20*, pp. 1505–1533, 2000.

Maier, R.W., Brennecke, J.F. and Stadtherr, M.A., Reliable computation of homogeneous azeotropes. *AIChE J.*, vol. 44, pp. 1745-1755, 1998.

Maier, R.W., Brennecke, J.F. and Stadtherr, M.A., Reliable computation of reactive azeotropes. *Comp. & Chem. Eng.*, vol. 24, pp. 1851-1858, 2000.

Main, R. A., Large Recursion Models: Practical Aspects of Recursion Techniques. In Optimization in Industry. John Wiley & Sons Ltd: New York, 1993.

Mani, A. and Patvardhan, C., A Novel Hybrid Constraint Handling Technique for Evolutionary Optimization. *In Proceedings of the Eleventh conference on Congress on Evolutionary Computation,* pp. 2577-2583. Trondheim, Norway, 2009.

Mashinchi, M.H., Orgun, M.A. and Pedrycz, W., Hybrid optimization with improved tabu search. *Appl. Soft Comp.,* vol. 11, pp. 1993-2006, 2011.

Masuduzzaman and Rangaiah, G.P., in: Rangaiah, G.P., (Ed.), *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 27-59, 2009.

McDonald, C. M., Floudas, C. A., Global Optimization for the Phase and Chemical Equilibrium Problem: Application to the NRTL Equation. *Comp.& Chem. Eng.,* vol. 19, pp. 1111-1141, 1995.

Meyer, C.; Floudas, C. A., Global optimization of a combinatorially complex generalized pooling problem. *AIChE J.***,** vol. *52*, pp. 1027-1037, 2006.

Michelsen, M.L., The isothermal flash problem. Part I. Stability. *Fluid Phase Equilib.*, vol. 9, pp. 1-19, 1982.

Miettinen, K.; Makela, M. M.; Maaranen, H., Efficient hybrid methods for global continuous optimization based on simulated annealing. *Comp.Operat. Res. 33*, pp. 1102-1116, 2006.

Misener, R. and Floudas, C. A., Advances for the pooling problem: modeling, global optimization, and computational studies. *Appl. Comp. Math.*, vol.8, pp.3-22, 2009.

Misener, R.; Floudas, C. A., Global Optimization of Large-Scale Generalized Pooling Problems: Quadratically Constrained MINLP Models. *Ind. Eng. Chem. Res.***,** vol. *49*, pp. 5424-5438, 2010.

Misener, R.; Gounaris, C. E.; Floudas, C. A., Mathematical modeling and global optimization of large-scale extended pooling problems with the (EPA) complex emissions constraints. *Comp. & Chem. Eng.***,** vol. *34*, pp. 1432-1456, 2010.

Misener, R.; Thompson, J. P.; Floudas, C. A., APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Comp. & Chem. Eng.***,** vol. 35, pp. 876-892, 2011.

Mitsos, A. and Barton, P.I., A dual extremum principle in thermodynamics**.** *AIChE J.***,** vol. 53, pp. 2131-2147, 2007.

Mladenovic, N.; Drazic, M.; Kovacevic-Vujcic, V.; Cangalovic., M., General variable neighborhood search for the continuous optimiation. *Euro. J. Opera. Res.***,** vol. 191, pp. 753-770, 2008.

Moles, C.G., Mendes, P., Banga, J.R., Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods. *Genome Res., 13* (11), pp. 2467-2474. 2003.

Nguyen, D.Q., Barbaro, A., Vipanurat, N. and Bagajewicz, M.J., All-at-once and step-wise detailed retrofit of heat exchanger networks using an MILP model. *Ind. Eng. Chem. Res.*, 49, pp. 6080-6103, 2010.

Nichita, D.V., Gomez, S. and Luna, E., Multiphase equilibria calculation by direct minimization of Gibbs free energy with a global optimization method. *Comp. & Chem. Eng.*, vol. 26, pp. 1703-1724, 2002a.

Nichita, D.V., Gomez, S. and Luna, E., Phase stability analysis with cubic equations of state by using a global optimization method. *Fluid Phase Equilib.*, vol. 194-197, pp. 411-437, 2002b.

Nichita, D.V., Gomez, S. and Luna-Ortiz, E., Multiphase equilibria calculation by direct minimization of Gibbs free energy using tunneling global optimization method. *J. Can. Petrol. Tech.*, vol. 43, pp. 13–16, 2004.

Nichita, D.V., Valencia, C.A.D. and Gomez, S. Volume-based thermodynamics global phase stability analysis. *Chem. Eng. Commu.,* vol. 193, pp. 1194-1216, 2008a.

Nichita, D.V., Garcia-Sanchez, F. and Gomez, S., Phase stability analysis using the PC-SAFT equation of state and the tunneling global optimization method. *Chem. Eng. J.*, vol. 140, pp. 509-520, 2008b.

Nichita, D.V. and Gomez, S., Efficient location of multiple global minima for the phase stability problem. *Chem. Eng. J.*, vol. 152, pp. 251-263, 2009.

Nichita, D.V. and Gomez, S., Efficient and reliable mixture critical points calculation by global optimization. *Fluid Phase Equilib.*, vol. 291, pp. 125-140, 2010.

Niewierowicz, T., Escarela-Perez, R., Campero-Littlewood, E., Hybrid genetic algorithm for the identification of high-order synchronous machine two-axis equivalent circuits. *Comp. Elect. Eng.,* vol.**29**, pp.505-522, 2003.

Nobakhti, A. and Wang, H. A simple self-adaptive differential evolution algorithm with application on ALSTOM gasifier. *Appl. Soft Comp.*, 8, pp. 350-370, 2008.

Nocedal, J.; Wright., S. J., *Numerical optimization* Springer: New York, 2006.

Okasinski, M.J. and Doherty, M.F., Thermodynamic behavior of reactive azeotropes. *AIChE J.*, vol. 43, pp. 2227-2238, 1997.

Omran, M.G.H., Engelbrecht, A.P. and Salman, A. Bare bones differential evolution. *Euro. J. Oper. Res.*, 196(1), pp. 128-139, 2009.

Pant, M., Thangaraj, R. and Singh, V.P. A New Differential Evolution Algorithm for Solving Global Optimization Problems. *Proceedings of the 2009 International Conference on Advanced Computer Control*, pp. 388-392, 2009.

Papamichail, I. and Adjiman, C. S. Global optimization of dynamic systems. *Comp. & Chem. Eng.*, 28, pp. 403–415, 2004.

Pardalos, P.M., Romeijin, H.E., Tuy, H., Recent developments and trends in global optimization. *J. Comp. App. Math.*, 124, pp. 209-228. 2000.

Pereira, F.E., Jackson, G., Galindo, A. and Adjiman, C.S., A duality-based optimisation approach for the reliable solution of (*P*, *T*) phase equilibrium in volume-composition space. *Fluid Phase Equilib.*, vol. 299, pp. 1-23, 2010.

Pham, V.; C.Laird; El-Halwagi, M., Convex Hull Discretization Approach to the Global Optimization of Pooling Problems. *Ind. Eng. Chem. Res.***,** vol. 48, pp. 1973-1979, 2009.

Pintarič, Z. N. and Kravanja, Z., Selection of the Economic Objective Function for the Optimization of Process Flow Sheets. *Ind. Eng. Chem. Res.***,** 45, pp.4222-4232, 2006.

Pinter, J., *Global Optimization in Action. Continuous and Lipschitz Optimization Algorithms, Implementations and Applications.* Kluwer Academics: Netherlands, 1996.

Platt, G.M. Henderson, N. and Medeiros, J.L., Determination of critical conditions of the esterification of acetic acid with ethanol in the presence of carbon dioxide. *Braz. J. Chem. Eng*., vol. 23, pp. 359-364, 2006.

Polishuk,I., Wisniak, J. and Segura, H., Prediction of the critical locus in binary mixtures using equation of state. I. Cubic equations of state, classical mixing rules, mixtures of methane-alkanes. *Fluid Phase Equilib*., vol. 164, pp. 13-47, 1999.

Ponce-Ortega, J.M., Jimenez, A. and Grossmann, I.E., Simultaneous retrofit and heat integration of chemical process. *Ind. Eng. Chem. Res.,* 47, pp.5512-5528, 2008.

Price, K., Storn, R.M. and Lampinen, J.A., *Differential evolution: A practical approach to global optimization*. Verlag Berlin Heidelberg: Springer, 2005.

Qi, Z. and Sundmacher, K., Geometrically locating azeotropes in ternary systems. *Ind. Eng. Chem. Res*., vol. 44, pp. 3709-3719, 2005.

Qin, A.K., Huang, V.L. and Suganthan, P.N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comp.*, 13(2), pp. 398-417, 2009.

Rahman, I., Das, A.K., Mankar, R.B. and Kulkarni, B.D., Evaluation of repulsive particle swarm method for phase equilibrium and phase stability problems. *Fluid Phase Equilib*., vol. 282 , pp. 65–67, 2009.

Rangaiah, G.P., Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems. *Fluid Phase Equilib.,* vol. 187-188, pp. 83-109, 2001.

Rangaiah, G.P., Stochastic *Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific: Singapore, 2010.

Reynolds, D., Mulholland, A.J., Gomatam, J., Calculation of Chemical and phase equilibrium via simulated annealing. *J.Math. Chem.,* vol.22, pp.25-37, 1997.

Rezaei, E., & Shafiei,S., Heat exchanger networks retrofit by coupling genetic algorithm with NLP and ILP methods. *Comp. & Chem. Eng.*, vol.33, pp.1451-1459, 2009.

Riggs, J.B., *An introduction to numerical methods for chemical engineers*, 2nd ed., Lubbock: Texas Tech University Press, 1994.

Rosenthal, R. E., GAMS - A User's Guide. 2010, http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf

Rossi, C.C.R.S., Cardozo-Filho, L. and Guirardello, R., Gibbs free energy minimization for the calculation of chemical and phase equilibrium using linear programing. *Fluid Phase Equilib.*, vol. 278, pp. 117-128, 2009.

Rossi, C.C.R.S., Berezuk, M.E., Cardozo-Filho, L. and Guirardello, R., Simultaneous calculation of chemical and phase equilibria using convexity analysis. *Comp. & Chem. Eng.*, in press, 2010.

Rudolph, G. , Convergence Analysis of Canonical Genetic Algorithms. *IEEE Trans. Neur.Network*, vol. 5, pp. 96-101, 1994.

Ryoo, H. S. and Sahinidis, N. V., Global optimization of non-convex NLPs and MINLPs with applications in process design. *Comp. & Chem. Eng.,* 19, pp.551-566, 1995.

Saber, N. and Shaw, J.M., Rapid and robust phase behaviour stability analysis using global optimization. *Fluid Phase Equilib.*, vol. 264, pp. 137-146, 2008.

Sahinidis, N. V.; Grossmann, I. E., Convergence Properties of Generalized Benders Decomposition. *Comp. & Chem. Eng.***,** vol. 15, pp. 481-491, 1991.

Sahinidis, N.; Tawarmalani, M., BARON Global Optimization Software. 2010, http://archimedes.cheme.cmu.edu/baron/manuse.pdf

Salomone, E. and Espinosa, J., Prediction of homogeneous azeotropes with interval analysis techniques exploiting topological considerations. *Ind. Eng. Chem. Res.*, vol. 40, pp. 1580-1588, 2001.

Sanchez-Mares, F. and Bonilla-Petriciolet, A., Cálculo de puntos críticos empleando uma estratégia de optimización global estocástica. *Afinidad*, vol. 63, pp. 396-403, 2006.

Schnepper, C.A. and Stadtherr, M.A., Robust process simulation using interval methods. *Comp. & Chem. Eng.*, vol. 20, pp. 187-199, 1996.

Schoen, F., Stochastic Global Optimization: Stopping Rules. Floudas, C.A., Pardalos, P.M., (edit) *Encyclopedia of optimization*, Springer, pp. 3743-3746, 2009.

Schulz, E.P., Diaz, M.S. and Bandoni, J.A., Supply chain optimization of large-scale continuous processes. *Comp. & Chem. Eng.*, 29, pp.1350-1316, 2005.

Schwaab, M., Biscala, E.C., Monteiro, J.L. and Pinto, J.C. Nonlinear parameter estimation through particle swarm optimization. *Chem. Eng. Sci.,* vol. 63, pp. 1542-1552, 2008.

Scurto, A.M., Xu, G., Brennecke, J.F. and Stadtherr, M.A., Phase behavior and reliable computation of high pressure solid-fluid equilibrium with cosolvents. *Ind. Eng. Chem. Res.*, vol. 42, pp. 6464–6475, 2003.

Seider, W.D. and Widagdo, S., Multiphase equilibria of reactive systems. *Fluid Phase Equilib.*, vol. 123, pp. 283-303, 1996.

Shenoy, U. V., *Heat Exchanger Network Synthesis,* Gulf Publishing Company, Houston, Texas, 1995.

Shi, Y., Eberhart, R.C., A modified particle swarm optimization. *IEEE International Conference on Evolutionary Computation*, 1998.

Sieniutycz, S., Jezowski, J.M., *Energy Optimization in Process Systems*. Elsevier, 2009.

Sim, M.K., Rangaiah, G.P. and Srinivas, M., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 147-201, 2010.

Silva, A.P., Ravagnaani, M.A.S.S., Biscaia, E.C., Particle Swarm Optimisation Applied in Retrofit of Heat Exchanger Networks. *10th International Symposium on Process Systems Engineering - PSE2009*, pp. 1035-1040, 2009.

Skolpap, W., Nuchprayoon, S., Scharer, J.M., Grisdanurak, N., Douglas, P.L. and Moo-Young, M., Fed-batch optimization of αα-amylase and protease-producing Bacillus subtilis using genetic algorithm and particle swarm optimization. *Chem. Eng. Sci.,* vol. 63, pp. 4090-4099, 2008.

Smith, R., Jobson, M., Chen, L., Recent development in the retrofit of heat exchanger networks. *Appl. thermal Eng.*, 30, pp. 2281-2289, 2010.

Soliman, O.S. and Bui, L.T. A self-adaptive strategy for controlling parameters in differential evolution. *IEEE Congress on Evolutionary Computation,* 2837-2842. 2008.

Srinivas, M. and Rangaiah, G.P., Implementation and evaluation of random tunneling algorithm for chemical engineering applications. *Comp. & Chem. Eng.,* vol. 30, pp. 1400-1415, 2006.

Srinivas, M. and Rangaiah, G.P., A study of differential evolution and tabu search for benchmark, phase equilibrium and phase stability problems. *Comp. & Chem. Eng*., vol. 31, pp. 760-772, 2007a.

Srinivas, M. and Rangaiah, G.P., Differential Evolution with Tabu List for Global Optimization and Its Application to Phase Equilibrium and Parameter Estimation Problems. *Ind. Eng. Chem. Res.*, vol. 46, pp. 3410-3421, 2007b.

Srinivas, M. and Rangaiah, G.P. Differential evolution with tabu list for solving nonlinear and mixed-integer nonlinear programming problems. *Ind. Eng. Chem. Res.*, 46, 7126-7135, 2007c.

Srinivas, M. and Rangaiah, G.P., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, pp. 465-504, 2010.

Stateva, R.P. and Wakeham, W.A., Phase equilibrium calculations for chemically reacting systems. *Ind. Eng. Chem. Res.*, vol. 36, pp. 5474-5482, 1997.

Staudt, P.B. and Soares, R.P., Reliability vs. Efficiency When Solving Multiphase Equilibrium Problems with Hybrid Optimization Codes. *Comp. Aid. Chem. Eng.*, vol. 27, pp. 585-590, 2009.

Steyer, F. and Sundmacher, K. VLE and LLE data for the system cyclohexane + cyclohexene + water + cyclohexanol", *J. Chem. Eng. Data*, vol. 49, pp. 1675-1681, 2004.

Stokes, L. and Plummer, J., Using spreadsheet solvers in sample design. *Comp. Statistics and Data Analysis,* 44, pp.527-546, 2004.

Storn, R. and Price, K., Differential evolution- a simple and efficient heuristic for global optimization over continuous space. *J. Glob. Optim.*, vol. 11, 341-359, 1997.

Stradi, B.A., Brennecke, J.F., Kohn, J.P. and Stadtherr, M.A. Reliable computation of mixture critical points. *AIChE J.*, vol. 47, pp. 212-221, 2001.

Sun, A.C. and Seider, W.D., Homotopy-continuation for stability analysis in the global minimization of the Gibbs free energy. *Fluid Phase Equilib.*, vol. 103, pp. 213-249, 1995.

Takahama, T. and Sakai, S., Constrained Optimization by the e Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. *In IEEE Congress on Evolutionary Computation*. Vancouver, BC, Canada, 2006.

Talbi, E.G., A taxonomy of hybrid metaheuristics. *J. Heuristics,* vol. 5, pp. 541-564, 2002.

Tawarmalani, M. and Sahinidis, N., Convexification and global optimization in continuous andmixed-integer nonlinear programming: theory, algorithms, software, and applications Kluwer Academic: Dordrecht, 2002.

Teh, Y.S. and Rangaiah, G.P. A study of equation-solving and Gibbs free energy minimization methods for phase equilibrium calculations. *Chem. Eng. Res. Des.*, vol. 80, pp. 745-759, 2002.

Teh, Y.S. and Rangaiah, G.P., Taboo search for global optimization of continuous function to phase equilibrium calculation. *Comp. & Chem. Eng.,* 27, pp.1665-1679. 2003.

Tessier, S.R., Brennecke, J.F. and Stadtherr, M.A., Reliable phase stability analysis for excess Gibbs energy models. *Chem. Eng. Sci.,* vol. 55, pp. 1785-1796, 2000.

Tjoe, T. N., Linnhoff, B., Using pinch technology for process retrofit. *Chem. Eng.*, 93, pp. 47-60, 1986.

Visweswaran, V.; Floudas, C. A., A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: II. Application of theory and test problems. *Comp. & Chem. Eng.***,** vol. 14, pp. 1419-1434, 1990.

Visweswaran, V.; Floudas, C. A., New Properties and Computational Improvement of the GOP Algorithm for Problems with Quadratic Objective Functions and Constraints. *J. Global Optim.***,** vol. 3, pp. 439-462, 1993.

Visweswaran, V.; Floudas, C. A., *New Formulations and Branching Strategies for the GOP Algorithm*. Kluwer Academic Publishers: Dordrecht, 1996.

Wakeham, W.A. and Stateva, R.P., Numerical solution of the isothermal, isobaric phase equilibrium problem. *Rev. Chem. Eng*., 20, pp. 1-56. 2004.

Wang, M.C., Wong, D.S.H., Chen, H., Yan, W. and Guo, T.M., Homotopy continuation method for calculating critical loci of binary mixtures. *Chem. Eng. Sci*., vol. 54, pp. 3873-3883, 1999.

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.

Wang, Y.J. and Zhang, J. S., An efficient algorithm for large scale global optimization of continuous functions. *J. Comp. Appl. Math.*, 206, pp.1015-1026, 2007.

Wasylkiewicz, S.K., Ung, S., Global phase stability analysis for heterogeneous reactive mixtures and calculation of reactive liquid–liquid and vapor–liquid–liquid equilibrium. *Fluid Phase Equilib.,* vol. 175, pp.253-272, 2000.

Weise, T. Global optimization algorithm: Theory and application**.** 2008. http://www.it-weise.de/.

Williams, T. J. and Otto, R. E., A generalized chemical processing model for the investigation of computer control. *AIEE Trans,* 79, pp.458-473, 1960.

Xu, G., Brennecke, J.F. and Stadtherr, M.A., Reliable computation of phase stability and equilibrium from the SAFT equation of state”, *Ind. Eng. Chem. Res*., vol. 41, pp. 938, 2002.

Yang, Z., Tang, K. and Yao, X. Large scale evolutionary optimization using cooperative coevolution, *Inform. Sci.*, 178, pp. 2985-2999, 2008.

Yang, Z., Zhang, J., Tang, K., Yao, X. and Sanderson, A.C., An Adaptive Coevolutionary Differential Evolution Algorithm for Large-Scale Optimization. CEC'09. IEEE Congress on. Evolutionary Computation, pp.102-109, 2009.

Yerramsetty, K.; Murty, C., Synthesis of cost-optimal heat exchanger networks using differential evolution. *Comp. & Chem. Eng.***,** vol. 32 (8), pp. 1861-1876, 2008.

Yuan, Q. and Qian, F., A hybrid genetic algorithm for twice continuously differentiable NLP problems. *Comp. & Chem. Eng.,* 34, pp.36-41, 2010.

Younes, A., Elkamel, A. and Areibi, S., in: Rangaiah, G.P., (Ed.), *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*, World Scientific, 2010.

Zahara, E. and Kao, Y.-T., Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problem**s**. *Expert Systems with Appl.s,* 36, pp.3880-3886, 2009.

Zhang, H. and Rangaiah, G.P., A hybrid global optimization algorithm and its application to parameter estimation problems. *Asia Pac. J. Chem. Eng.,* vol. 6, pp. 379-390, 2011.

Zhang, H., Kennedy, D.D., Rangaiah, G.P. and Bonilla-Petriciolet, A., Novel bare-bones particle swarm optimization and its performance for modeling vapor-liquid equilibrium data. *Fluid Phase Equilib.,* vol. 301, pp. 33-45, 2011.

Zhang, H., Fernandez-Vargas, J.A., Rangaiah, G.P., Bonilla-Petriciolet, A. and Segovia-Hernandez, J.G., Evaluation of integrated differential evolution and unified bare-bones particle swarm optimization for phase equilibrium and stability problems, *J. Fluid Phase Equilib.*, 310, pp. 129-141, 2011.

Zhang, H., Rangaiah, G. P. and Bonilla-Petriciolet, A., Integrated Differential Evolution for Global Optimization and Its Performance for Modeling Vapor-Liquid Equilibrium Data. *Ind. Eng. Chem. Res.,* 50, pp. 10047-10061, 2011.

Zhang, X., Chen, W., Dai, C. and Guo, A. Self-adaptive Differential Evolution Algorithm for Reactive Power Optimization. *Fourth International Conference on Natural Computation*, pp. 560-564, 2008.

Zhigljavsky, A.; Zilinskas, A., *Stochastic global optimization* Springer: New York, 2007.

Zhu, Q.; Qin, A.; Suganthan, P.; Huang, G., Evolutionary extreme learning machine. *Pattern Recognition***,** vol. 38 (10), pp. 1759-1763, 2005.

Zhu, Y., Wen, H. and Xu, Z., Global stability analysis and phase equilibrium calculations at high pressures using the enhanced simulated annealing algorithm. *Chem. Eng. Sci.,* vol. 55, pp. 3451-3459, 2000.

Zhu, Y. and Inoue, K., Calculation of chemical and phase equilibrium based on stability analysis by QBB algorithm: application to NRTL equation. *Chem. Eng. Sci*., vol. 56, pp. 6915-6931, 2001.

Zielinski, K. and Laur, R. Stopping criteria for a constrained single-objective particle swarm optimization algorithm. *Informatica*, 31, pp. 51-59, 2007.

Publications

The contributions in this thesis have been published/will be published in several international journals and international/regional conferences; these are listed below.

Publications in international journals

1. Zhang, H., Rangaiah, G.P. and Bonilla-Petriciolet, A., Integrated differential evolution for global optimization and its performance for modeling vapor-liquid equilibrium data, *Ind. Eng. Chem. Res.*, 50, pp. 10047-10061, 2011a.
2. Zhang, H., Kennedy, D.D., Rangaiah, G.P. and Bonilla-Petriciolet, A., Novel bare-bones particle swarm optimization and its performance for modeling vapor-liquid equilibrium data, *Fluid Phase Equilib.*, 301, pp. 33-45, 2011b.
3. Zhang, H. and Rangaiah, G.P., A hybrid global optimization algorithm and its application to parameter estimation problems, *Asia-Pacific J. Chem. Eng.*, 6, pp. 379-390, 2011a.
4. Zhang, H., Fernandez-Vargas, J.A., Rangaiah, G.P., Bonilla-Petriciolet, A. and Segovia-Hernandez, J.G., Evaluation of integrated differential evolution and unified bare-bones particle swarm optimization for phase equilibrium and stability problems, *Fluid Phase Equilib.*, 310, pp. 129-141, 2011c.
5. Zhang, H., Bonilla-Petriciolet, A. and Rangaiah, G.P., A review on global optimization methods for phase equilibrium modeling and calculations, *The Open Thermodynamic Journal*, 5, pp. 71-92, 2011d.
6. Zhang, H. and Rangaiah, G.P., An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization, *Computers & Chemical Engineering*, 37, pp. 5542-5551, 2012.

Articles in preparation

1. Zhang, H., Lee, K.L. and Rangaiah, G.P., New formulation and approach for global optimization of pooling problems. Submitted. 2011.
2. Zhang, H. and Rangaiah, G.P., One-step approach for heat exchanger network retrofitting using integrated differential evolution. Submitted, 2012.

Contributions to international/regional conferences

1. Zhang, H. and Rangaiah, G.P., A hybrid global optimization algorithm, *The 5th International Symposium on Design, Operation and Control of Chemical Processes* (PSE-ASIA 2010), July 25-28, 2010, Singapore.
2. Kennedy, D.D., Zhang, H. and Rangaiah, G.P., Novel bare-bones particle swarm optimization, *The 5th International Symposium on Design, Operation and Control of Chemical Processes* (PSE-ASIA 2010), July 25-28, 2010, Singapore.
3. Zhang, H. and Rangaiah, G.P., Self-adaptive differential evolution with taboo list for constrained optimization problems and its application to pooling problems, *21st European Symposium on Computer Aided Process Engineering* (ESCAPE 21), May 29- June 1, 2011, Chalkidiki, Greece.
4. Zhang, H., Lee, K.L. and Rangaiah, G.P., New approach for optimizing pooling problems using integrated differential evolution, *14th Asia Pacific Confederation of Chemical Engineering Congress* (APCChE 2012), February 21-24, 2012, Singapore.
5. Rangaiah, G.P., Zhang, H. and Sharma, S., (Invited Presentation) Stochastic global optimization techniques: progress, challenges and prospects for chemical engineering applications, *14th Asia Pacific Confederation of Chemical Engineering Congress* (APCChE 2012), February 21-24, 2012, Singapore.