STOCHASTIC GLOBAL OPTIMIZATION METHODS AND THEIR APPLICATIONS IN CHEMICAL ENGINEERING

MEKAPATI SRINIVAS

NATIONAL UNIVERSITY OF SINGAPORE

2007

STOCHASTIC GLOBAL OPTIMIZATION METHODS AND THEIR APPLICATIONS IN CHEMICAL ENGINEERING

MEKAPATI SRINIVAS

(B.Tech., National Institute of Technology, Warangal, India)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

To My Parents

&

Late Sri Ch. Deekshitulu

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. G. P. Rangaiah for his invaluable guidance, continuous support and encouragement throughout this work, particularly at difficult times. I would also thank him for teaching me the important things: positive attitude, critical thinking and analysis while attacking any problem during my research work.

I would like to thank Prof. I. A. Karimi, Prof. A. K. Ray, Prof. M. S. Chiu, Prof. X. S. Zhao and Prof. T. C. Tan for teaching me the fundamentals and advanced topics in chemical engineering. I would also like to thank Prof. Raj Rajagopalan, Prof. S. Lakshminarayanan, Prof. P. R. Krishnaswamy 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.

Many thanks to my friends Sudhakar, Murthy, Sathish, Velu, Prakash, Yelneedi, Satyanarayana, Umapathi and B.V. Srinivas for their continuous support and encouragement over the years. Special thanks to my colleagues and other friends in the department who made my life at NUS more enjoyable.

I would like to express my endless gratitude to my beloved parents and family members for their everlasting love, support and encouragement in my life particularly in

pursing graduate studies. I would like to extend my gratitude to my master Sri Ch. Satyanarayana and his son Late Ramalinga Deekshitulu for their guidance and help in achieving my goals.

I sincerely acknowledge the National University of Singapore for providing me the opportunity and financial support to pursue doctoral studies.

TABLE OF CONTENTS

SUMMARY

Stochastic global optimization is an active research area due to its ability to provide the best possible solutions to the highly non-linear, non-convex and discontinuous objective functions. The broad objective of this study is to develop, apply and evaluate reliable and efficient stochastic global optimization methods for chemical engineering applications. Two benchmark problems similar to phase equilibrium calculations have also been proposed and studied in this thesis.

After reviewing the literature, a method, namely, random tunneling algorithm (RTA) is selected, implemented and evaluated for benchmark problems involving 2 to 20 variables and a few to hundreds of local minima. Its potential is then tested for chemical engineering applications, namely, phase equilibrium calculations via Gibbs free energy minimization and parameter estimation in models. Phase equilibrium problems considered include vapor-liquid, liquid-liquid and vapor-liquid-liquid examples involving several components and popular thermodynamic models, and parameter estimation problems consist of up to 34 variables. RTA successfully located global minimum for most the examples but its reliability is found to be low for problems with a local minimum comparable to the global minimum.

Subsequently, two methods, namely, differential evolution (DE) and tabu search (TS) have been evaluated for benchmark problems, phase equilibrium calculations and phase stability problems. The results show that DE is more reliable in locating the global

minimum compared to TS whereas the latter is computationally more efficient than the former for the examples tested.

Guided by the insight obtained from DE and TS, a new method, namely, differential evolution with tabu list (DETL) is proposed by integrating the strong features of DE and TS. DETL is initially tested over many of benchmark problems involving multiple minima. The results show that the performance of DETL is better compared to DE, TS and the recent modified differential evolution (MDE). DETL is then evaluated for challenging phase equilibrium calculations and parameter estimation problems in differential and algebraic systems. It is also evaluated for many non-linear programming problems (NLPs) having constraints and different degrees of complexity, and several mixed-integer non-linear programming problems (MINLPs) which involve process design and synthesis problems. Overall, the performance of DETL is found to be better than that of DE, TS and MDE.

Finally, a transformation for the objective function is proposed to enhance the reliability of stochastic global optimization methods. The proposed transformation is implemented with DE, and is evaluated for several test functions involving multiple minima. Although the proposed transformation is found to be better than a recent transformation in the literature, further investigation is required to improve its effectiveness for problems with more variables.

Comprehensive evaluation of the selected stochastic optimization methods, and the new method, benchmark problems and transformation proposed in this study are of interest and use to the global optimization community.

NOMENCLATURE

Abbreviation	Explanation
DE	: Differential Evolution
DETL	: Differential Evolution with Tabu List
ECTS	: Enhanced Continuous Tabu Search
ES	: Evolutionary Strategies/Easom function
FAEA	: Fast Annealing Evolutionary Algorithm
FB	: Forcing to Bounds
GA	: Genetic Algorithms
GM	: Global Minimum
GP	: Goldstein and Price function
GRDT	: Gradient Descent with Dynamic Tunneling
GW	: Griewank function
H	: Hartmann function
HDE	: Hybrid Differential Evolution
IDE	: Improved Differential Evolution
LLE	: Liquid-Liquid Equilibrium
LM	: Local Minimum
mHB	: modified Himmelblau Function
mNDT	: modified N-dimensional Test Function
mROS	: modified Rosenbrock Function
MBB	: Modified Bouncing Ball

MDE : Modified Differential Evolution

MINLPs : Mixed-Integer Non-Linear Programming problems

MTT : Monotonic Transformation

NFE : Number of Function Evaluations

NFG : Number of Function and Gradient evaluations

NLPs : Non-Linear Programming problems

NM : Nelder-Mead

NRTL : Non-Random Two Liquid

OA/ER : Outer approximation/Equality Relaxation

PEC : Phase Equilibrium Calculations

PS : Phase Stability

PSO : Particle Swarm Optimization

QN : quasi-Newton

RA : Rastrigin function

RG : Random Generation

ROS : Rosenbrock function

RT : Random Tunneling

RTA : Random Tunneling Algorithm

SA : Simulated Annealing

SC1 : Stopping Criterion 1

SC2 : Stopping Criterion 2

SF : Snyman-Fatti

SH : Shubert function

SPT	: Stochastic Pijavaskij Tunneling
SR	: Success Rate
SSE	: Sum of Squared Errors
TPDF	: Tangent Plane Distance Function
TRUST	: Terminal Repeller and Unconstrained Subenergy Tunneling
TS	: Tabu Search
UNIQUAC	: Universal Quasi-Chemical
VLE	: Vapor-Liquid Equilibrium
VLLE	: Vapor-Liquid-Liquid Equilibrium
WOT	: Without Transformation
WT	: With Transformation
ZAK	: Zakharov function

Symbol	**Explanation**
A	Amplification factor
c	Parameter of the proposed transformation
CR	Crossover constant
$E(x, x^*)$	Virtual objective function
$E_{rep}(x, x^*)$	Repeller term
$E_{sub}(x, x^*)$	Subenergy tunneling function
f	Objective function
f_{trans}	Transformed objective function
F	Estimated objective function using Lipschitz constant, or Total moles of feed

g	Inequality constraint
G	Gibbs free energy
Gen_{max}	Maximum number of generations
$\overline{G}$	Partial molar Gibbs free energy
$\frac{G}{RT}$	Dimensionless Gibbs free energy
h	Equality constraint
h_n	Length of the n^{th} hyper-rectangle
H	Homotopy function, or Dimensionless tangent plane distance Function, or Heavyside function
Iter	Iterations
$Iter_{max}$	Maximum number of iterations
k	Rate constant
L	Lipschitz constant
m_1	Number of equality constraints
m_2	Number of inequality constraints
n	Number of moles
n^k	Total number of moles in phase k
n_i^k	Number of moles of component i in phase k
nc	Number of components
N	Number of real/continuous variables of the given problem
N_{neigh}	Number of neighbors in each iteration
N_p	Number of promising points
N_t	Number of taboo points

NP	Population size
NP_{init}	Initial population size
p	Total number of continuous, binary and integer variables
ps	Number of perturbations in a particular tunneling phase
P	System pressure
R	Universal gas constant
s	Centroid of the hyper-rectangle
ss	Step size
S	Sign variable for the direction of tunneling
Sc	Successive iterations/generations
Sc_{max}	Maximum number of successive generations without improvement in the best function value
t	Homotopy parameter, or Tangent plane
tls	Tabu list size
tp	Maximum number of tunneling phases
tr	Tabu radius
T	Temperature
$\tilde{\mathbf{T}}$	m – dimensional temperature vector
U	Under-estimator or Trial vector/individual
V	Mutated vector/individual
$\mathbf{x}$	Decision variable, or Mole fraction in liquid phase
$\dot{x}$ and $\ddot{x}$	System of first and second order differential equations
$\tilde{\mathbf{x}}_1$	m - dimensional composition vector of component 1
$\dot{x}_i$	i^{th} component of the dynamic system

x_i^l and x_i^u	Lower and upper bounds of i^{th} element of x
$x_{1,i}$	Measured x_1 in i^{th} experiment
X	Target vector/individual
y	Binary variable, or Mole fraction in vapor phase
z_i	Moles of i^{th} component in the feed

Greek Letters	**Explanation**
α	Constant
β_i	i^{th} component of the decision variable
γ_i	Activity coefficient of component i
θ	Energy interaction parameter vector, or Decision variable in parameter estimation problems
λ	Magnitude of perturbation
ϕ_i	Fugacity coefficient of component i
σ	Standard deviation
ε	Radius

Superscripts	**Explanation**
k	Phase index
l	Lower bound
L	Liquid phase
u	Upper bound
V	Vapor phase

Subscripts	Explanation
G/J/gen	Generation number
init	Initial stage
max	Maximum number
neigh	Neighbor point
P	Promising point
t	Tabu point

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Importance of Global Optimization

Many practical engineering problems can be posed as optimization problems for which a global optimal solution is desired. Global optimization methods are important in many applications because even slightly better solutions can translate into large savings in time, money and/or resources. In several modeling applications, they are required for correct prediction of process behavior. Literature abounds a range of applications of global optimization from estimating the amount of chlorophyll concentration in vegetation (Yatsenko et al., 2003) to the design of Earth-Mars transfer trajectories (Vasile et al., 2005). Global optimization techniques are typically general and can be applied to complex problems in engineering design (Grossmann, 1996), business management (Arsham, 2004), bio-processes (Banga et al., 2003), computational biology (Klepeis and Floudas, 2003), computational chemistry (Floudas and Pardalos, 2000), structural optimization (Muhanna and Mullen, 2001), computer science (Sexton et al., 1998), operations research (Faina, 2000), exploratory seismology (Barhen et al., 1997), process control and system design (Floudas, 2000a) etc.

The aim of global optimization is to find the solution in the region of interest, for which the objective function achieves its best value, the global optimum. Global minimization aims at determining not just a local minimum but the minimum among the minima (smallest local minimum) in the region of interest. In contrast to local

optimization for which the attainment of the local minimum is decidable (via gradient equal to zero and Hessian matrix is positive definite), no such general criterion exists in global optimization for asserting that the global minimum has been reached.

Significant advances in optimization were made in the early 1940's and many of them are limited to linear programming until late 1960s. However, the assumption of linearity is restrictive in expressing the real world applications due to the need for non-linear expressions in the models. Initially, non-linear programming problems are addressed and solved to global optimality using local optimization techniques under certain convexity assumptions (Minoux, 1986). However, many of these problems are often modeled via non-convex formulations that exhibit multiple optima. As a result, application of traditional local optimization techniques to such problems fails to achieve the global solution, thus requiring global optimization.

For example, consider a nonlinear function $f = -\left\{\sum_{i=1}^{5} \sin[(i+1)x + i]\right\}$, where x is a one dimensional vector with bounds -5 to 2. As shown in Figure 1.1, the function has 5 local minima (points a, b, c, d and e in the figure) among which the global minimum is at $x = -1.97$ (point 'e') with function value $f = -2.8851$. Any local optimization technique like steepest descent method, conjugate gradient method, Newton method or quasi-Newton method can provide only the local minimum (a or b or c or d) but not the global minimum unless the initial guess is near the global minimum.

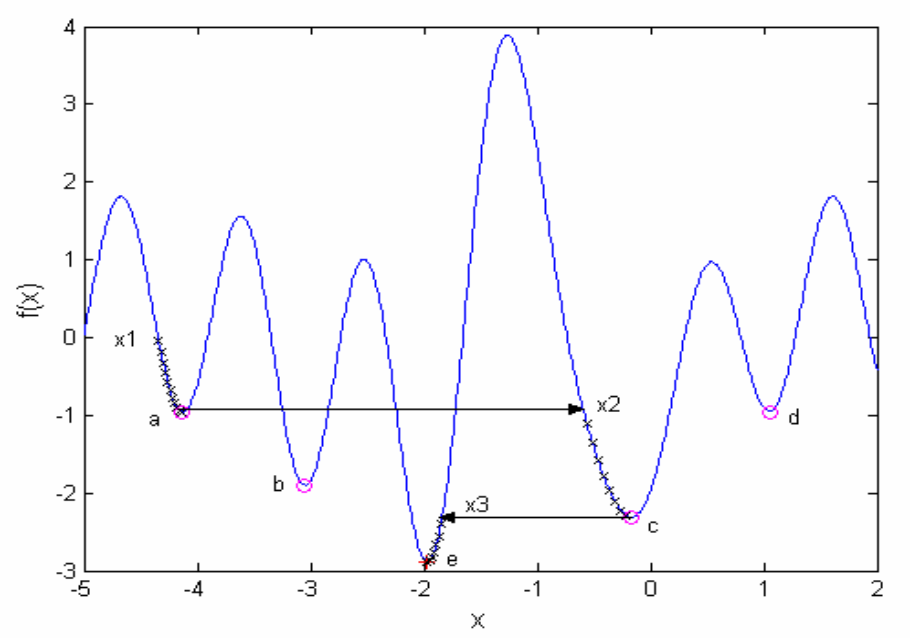


Figure 1.1 Schematic of the non-linear function, f(x) with multiple minima

Let us consider a global optimization technique, namely, tunneling method (Levy and Montalvo, 1985). The method starts with an initial guess 'x1' and uses a local minimization technique such as gradient descent or Newton's method to find the nearest local minimum 'a'. Then the tunneling phase will be switched on and it calculates the zeros of the tunneling function such that $x_1 \neq x_2$, but $f(x_1) = f(x_2)$ (as shown in Figure 1.1). The zero of the tunneling function 'x_2' is used as the starting point for the next minimization phase and the second lowest minimum (point c) will be located. The cycle of minimization phase and tunneling phase is repeated until it finds the global minimum at 'e'. Generally, the tunneling algorithm will terminate after a specified maximum number of iterations.

The first collection of mathematical programming papers in global optimization (in English literature) is published in the seventies (Dixon and Szego, 1975 and 1978). The benefits that can be obtained through global optimization of non-

convex problems motivated several researchers in this direction (e.g., van Laarhoven and Aarts, 1987; Floudas and Pardalos, 1996; Glover and Laguna, 1997; Bard, 1998; Tuy, 1998; Haupt and Haupt, 1998; Sherali and Adams, 1999; Floudas, 2000a and b; Horst et al., 2000; Tawarmalani et al., 2002; Zabinsky, 2003; and Price et al. 2005) and the publication of the journal, "Journal of Global Optimization" by Kluwer Academic Publishers since 1991.

1.2 Classification of Global Optimization Methods

Global optimization methods can be broadly classified into two categories, namely, deterministic methods and stochastic methods (Pardalos et al., 2000; Horst and Pardalos, 1995). The former methods provide guaranteed solution under certain conditions whereas stochastic methods provide a weak asymptotic guarantee but often generate near-optimal solutions quickly. Deterministic methods exploit analytical properties of the function such as convexity and monotonic feature whereas stochastic methods require little or no assumption over the optimization problem such as continuity of function. Deterministic methods generate a deterministic sequence of points in the search space whereas stochastic methods generate random points in the search space. These two classes of optimization methods illustrate the trade-off between the ability to find the exact answer quickly and the ability to generate near-optimal solutions quickly. There are many deterministic and stochastic methods, which can be further classified into several groups as shown in Figure 1.2. Each group of methods is briefly introduced below.

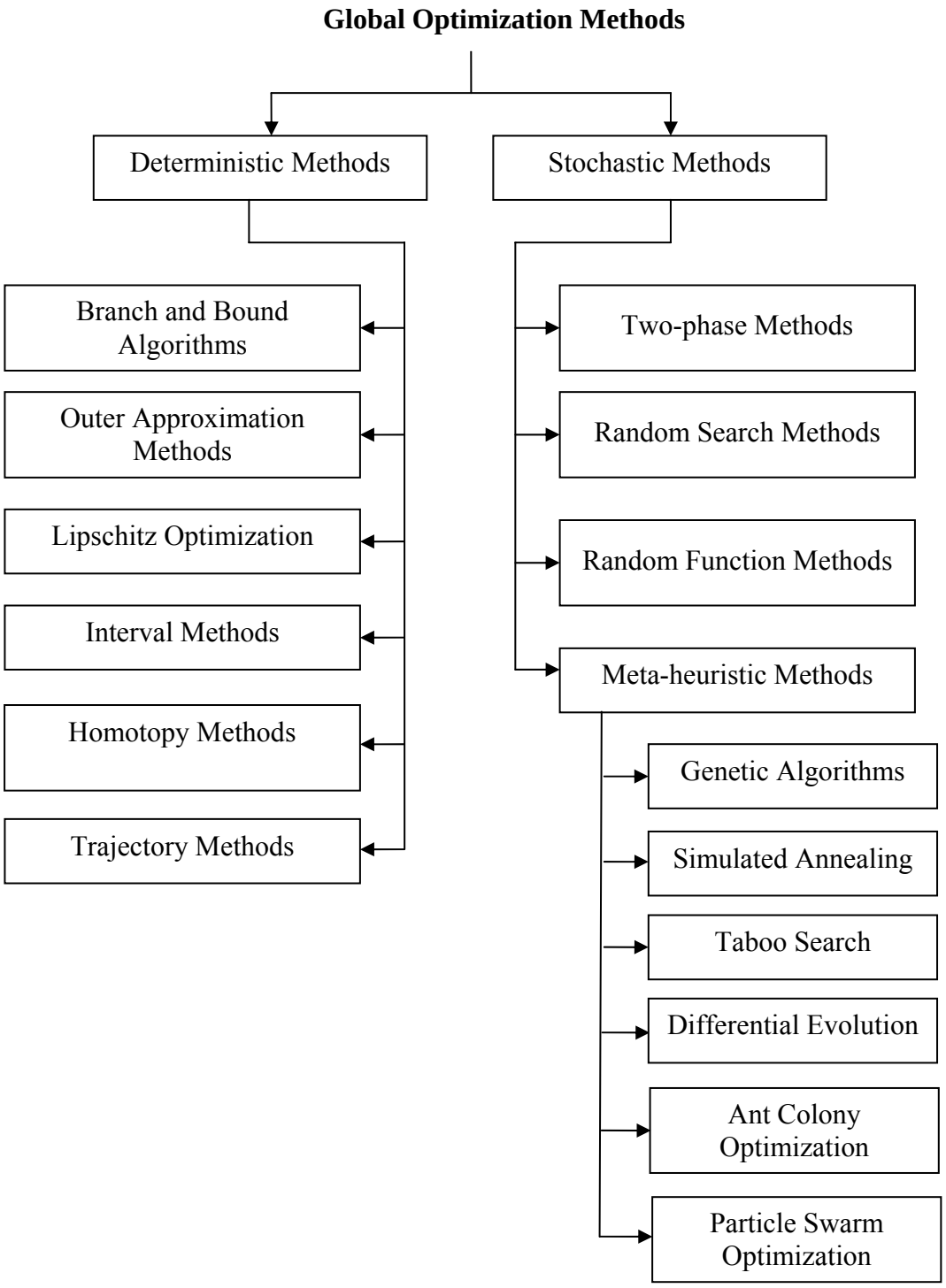


Figure 1.2: Classification of global optimization methods

Besides global optimization model in the methods shown in Figure 1.2, there are different types of models, which use some special properties of the objective function (Horst and Pardalos, 1995). They are difference of convex optimization problems (objective function is expressed as a difference of two convex functions), monotonic optimization problems (function is monotonically increasing or decreasing), quadratic optimization problems (quadratic objective function), multiplicative programming problems (objective function is expressed as a product of convex functions), fractional programming problems (ratio of two functions) etc.

Deterministic methods

There are six types of methods under this category namely, branch and bound algorithms, outer approximation methods, Lipschitz optimization, interval methods, homotopy methods and trajectory methods.

Branch and Bound Algorithms: The main principle behind branch and bound (B&B) algorithms is the "divide and conquer" technique known as branching and bounding (Adjiman et al., 1998; and Edgar et al., 2001). The method starts by considering the original problem with the complete feasible region known as the root problem. Branching is carried out by making the total feasible region into different partitions wherein objective function in each partition is bounded using a suitable convex underestimation function. The algorithm is applied recursively to the sub-problems until a suitable termination condition is satisfied. For example, consider a non-linear function, f(**x**), where **x** is a two dimensional vector with bounds

$0 \le x_1, x_2 \le 2$. Let the initial feasible region is partitioned into four smaller rectangles thus forming B&B tree shown in Figure 1.3.

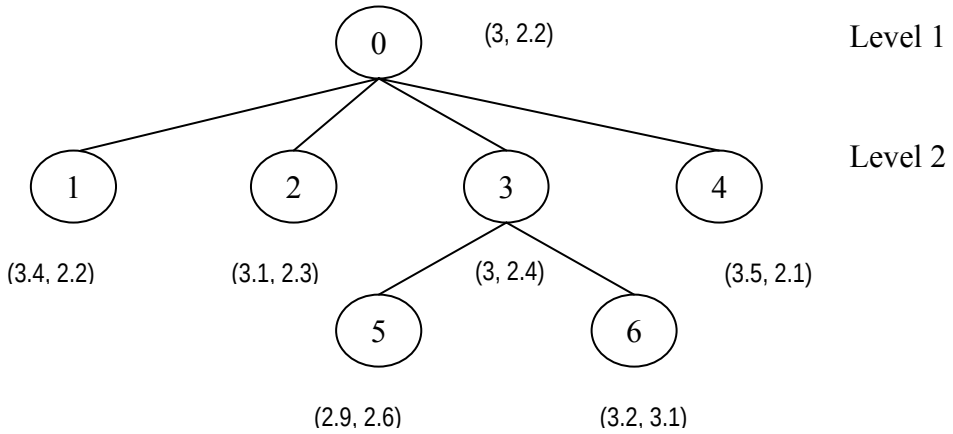


Figure 1.3: Branch and bound tree

The root problem with the original feasible region corresponds to the node '0' in Level 1. Then the problem is partitioned into four sub problems (branching) representing four nodes in the Level 2. The numbers in the brackets represent the upper and lower bounds of the objective function corresponding to each node respectively. The lower and upper bounds for each node are obtained respectively by minimizing the corresponding convex underestimation function over its partition and evaluating the original function at the minimum of the underestimation function. The smallest upper bound obtained over all partitions is retained as the "incumbent", the best point found so far. Thus the upper bound corresponding to node '3' is the best point obtained in Level 2. The branching is again repeated by selecting a node with the smallest upper bound in the current level (node '3' in Level 2) thus resulting nodes '5' and '6'. The lower bounds over these partitions are obtained again by solving the corresponding convex underestimation functions. It is clear from Figure 1.3 that the lower bound at node 6 (3.1) exceeds the incumbent value that has been

stored previously (3.0), a function value lower than 3 cannot be found by further branching from node 6 and is not considered for branching again. The branching step is repeated from node 5 until the difference between the incumbent f value and the lower bound at each node is smaller than the user defined tolerance.

The advantage of B & B method is that it provides mathematical guarantee to the global optimum under certain conditions whereas the main difficulty of this algorithm lies in finding a suitable convex underestimation function. For a general non-convex function f(**x**) over the domain [$\mathbf{x}^u$, $\mathbf{x}^l$], the convex under-estimator, U(**x**), is given by

$$U(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{N} \alpha_i (x_i^l - x_i)(x_i^u - x_i) \qquad 1.1$$

where **x** is an N-dimensional state vector and α_i is a positive scalar value. Since the quadratic term in the above function is convex, all the non-convexities in the original function can be subdued by choosing a large value for α_i. The major difficulty comes in choosing the value for α_i. There are many applications of B&B methods in both combinatorial and continuous optimization (Horst and Pardalos, 1995) including phase equilibrium problems (McDonald, 1995a, b and c).

Outer Approximation Methods: Outer Approximation (OA) is a technique in which the original feasible set is approximated by a sequence of simpler relaxed sets (Horst and Tuy, 1996). In this technique, the current approximating set is improved by a suitable additional constraint. For example, let the initial feasible set, D is relaxed to a simpler set, D_1 containing D, and the original objective function is minimized over the relaxed set (D_1). If the solution of the relaxed problem is in 'D', then the problem

is solved; otherwise, an appropriate portion of D_1 is cut off by an additional constraint resulting a new relaxed set D_2, which is a better approximation of D compared to D_1. Then the earlier relaxed set (D_1) is replaced by a new one (D_2) and the procedure is repeated. These methods are used as basic methods in many fields of optimization particularly in combinatorial optimization. The main disadvantage of these methods is that the size of the sub-problem increases from iteration to iteration, since in each step a new constraint is added to the existing set of constraints. The applications of OA methods include minimizing concave functions over convex sets (Hoffman, 1981; and Tuy, 1983) and mixed-integer non-linear programming problems (Duran and Grossmann, 1986).

Lipschitz Optimization: Lipschitz optimization solves global optimization problems in which the objective function and constraints are given explicitly and have a bounded slope (Hansen and Jaumard, 1995). In other words, a real function, f, defined on a compact set, X, is said to be Lipschitz if it satisfies the condition

$$\left|f(x_1) - f(x_2)\right| \le L\left\|x_1 - x_2\right\|;\ \ \forall x_1, x_2 \in X \qquad 1.2$$

where L is called as Lipschitz constant and ||.|| denotes the Euclidean norm. The first, best known and most studied algorithm for univariate Lipschitz optimization is called as Piyavskii's algorithm. It is a sequential algorithm which constructs a saw-tooth cover iteratively for the objective function (Figure 1.4), f, and evaluates f at a point corresponding to a minimum of this saw-tooth cover.

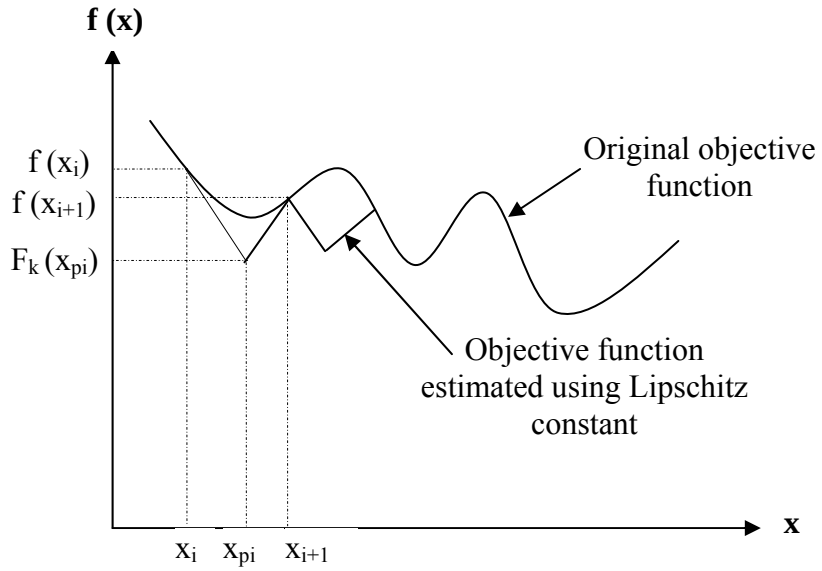


Figure 1.4: Saw-tooth cover in Lipschitz optimization

Thus at each iteration, the lowest tooth is found and the function is evaluated at the lowest point of this tooth, which is then split into two smaller teeth. The algorithm stops when the difference between estimated objective function value (F) using Lipschitz constant and the incumbent value does not exceed the tolerance 'ε'. Pinter has published several papers (Pinter, 1992) on Lipschitz optimization addressing the convergence issues, algorithms for the n-dimensional case and applications. The main draw back of these methods is that the number of sub-problems (i.e., the number of teeth in the lower bounding function) may become very large and scanning their list to find the sub-problem with the lowest lower bound is time consuming, particularly for large dimensional problems. The typical situation where Lipschitz optimization algorithms are most adequate is i) if the problem has only a few variables, and (ii) if specific knowledge of the problem that allows finding Lipschitz constant or fairly accurate estimate of Lipschitz constant is available. The applications (Wingo, 1984; Love et al., 1988; and Hansen and Jaumard, 1995) include

parameterization of statistical models, black-box design of engineering systems, location and routing problems.

Interval Methods: Interval methods (Hansen, 1992) mainly deal with interval numbers, interval arithmetic and interval analysis instead of real numbers, real arithmetic and real analysis used in general. The initial interval should be large enough to enclose all feasible solutions. There are many applications of interval methods such as parameter estimation problems, circuit analysis problems (Horst and Pardalos, 1995) and phase stability analysis (Burgos-Solórzano et al., 2003) problems. The main advantage of this method is it provides guaranteed global minimum and it controls all kinds of errors (round-off errors, truncation errors etc.).

The interval method starts with searching for stationary points in the given initial interval with the powerful root inclusion test based on interval-Newton method. This test determines with mathematical certainty if an interval contains no stationary or a unique stationary point. If neither of these results is proven, then the initial interval is again divided into two subintervals and the same procedure is repeated. The method terminates if the bounds over the solutions are sufficiently sharp or no further reduction of the bounds occurs. On completion, the algorithm provides the narrow enclosures of all the stationary points in the given initial interval with sharp bounds so that global optimum can easily be determined. This method finds all the global and local optima in the given feasible region unlike finding only one solution by many other methods.

The main drawbacks of interval methods are large computational time and inconvenient to deal with intervals in the programming. The reason for large computational time is due to the Newton method which finds all the stationary points in the region. Some eliminating techniques (Hansen, 1992) are used in order to reduce the computational time. These eliminating procedures reduce the search region by eliminating all or part of the current box (interval) using different techniques such as calculating the gradient, performing non-convexity check using Hessian matrix and calculating upper bound over the global minimum. Besides these techniques, the number of iterations can also be reduced by introducing slope function in the Newton method which is called as slope interval Newton method (Hansen, 1992).

Homotopy Methods: Homotopy methods are important in solving a system of non-linear equations (Forster, 1995). They start with solving a simple problem and then deforming it into the original complicated problem. During this deformation or homotopy, the solution paths of the simple problem and deformed problems are followed to obtain the solution for the original complicated problem. The homotopy function consists of a linear combination of two functions: the original function, $f(\mathbf{x}) = 0$ whose solution is to be calculated and a function, $g(\mathbf{x}) = 0$ for which a solution is known or can be readily calculated. The homotopy function can be defined as $H(\mathbf{x}, t) = t\ f(\mathbf{x}) + (1 - t)\ g(\mathbf{x}) = 0$, where t is the homotopy parameter which allows the tracking of solution path from the simple problem to the solution of the original complex problem. As the parameter, t is gradually varied from 0 to 1 and $H(\mathbf{x}, t) = 0$ is solved using a suitable method, the series of solutions to $H(\mathbf{x}, t) = 0$ traces a path to the solution of the original function $f(\mathbf{x}) = 0$. Both the original and simple functions ($f(\mathbf{x})$ and $g(\mathbf{x})$) are combined and formulated into an initial value problem in ordinary

differential equations, and is solved by suitable method to obtain solution to the original function. Based on the selection of g(**x**), there are different types of homotopy methods: Newton homotopy and fixed point homotopy. These methods provide guaranteed convergence to a solution if it exists and a continuous path for the homotopy function (i.e., H (**x**,t)) from t = 0 to t = 1 (Sun and Seider, 1995) also exists. However, homotopy methods does not provide guarantee to all multiple solutions. The main advantage of homotopy methods is that they can be applied to complicated systems where nothing is known *a priori* about the solution, and the disadvantage is computationally expensive. Applications of these methods include solving economic equilibrium models, generalized plane stress problem (Forster, 1995) and phase equilibrium problems (Sun and Seider, 1995).

Trajectory Methods: These methods construct a set of trajectories in the feasible region in such a way that at least some of the solutions of the given problem lie on these paths (Diener, 1995). In most of the cases, these trajectories are obtained by solving ordinary differential equations of first or second order. Based on the definition of the trajectory, there are several methods such as Griewank's method, Snyman-Fatti method, Branin's method and Newton leaves method. For example, Snyman-Fatti algorithm solves the following second order differential equation

$$\ddot{x}(t) = -\nabla f(x), \quad \text{with } x(0) = x_0^i \text{ and } \dot{x}(0) = \dot{x}_0^i \tag{1.3}$$

Here, the trajectory is the motion of a particle of unit mass in an N-dimensional conservative force field. The algorithm starts with an initial value taken in the feasible region and solves the differential equations by searching along the trajectory. The trajectory will be terminated whenever it finds a function value which is equal to or approximately equal to the function value at the starting point i.e., the trajectory

terminates before it retraces itself. The best point found along the trajectory is recorded and is used as the next starting point from which the differential equations are solved again. The algorithm terminates after a specified number of samplings and declares the current overall minimum as the global minimum. The trajectory methods escape from the local minimum by climbing up-hills which is inherent in the physical models that they approximate. They can also be modified by including stochastic elements like random choices of initial data. The main disadvantage of trajectory methods is they do not work well if many local minima are present. Reported applications (Diener, 1995) of trajectory methods include parametric optimization, artificial neural network training and broadcasting.

Stochastic methods

Four main classes of stochastic methods are: two-phase methods, random search methods, random function methods and meta-heuristic methods (Boender and Romeijn, 1995).

Two-phase Methods: Two-phase methods consist of a global phase and a local phase (Pardalos et al., 2000). In the global phase, the objective function is evaluated at a number of randomly sampled points in the feasible region. In the local phase these sample points are improved by using any one of the local optimization techniques. The best local optimum found will be the estimate of the global optimum. Two-phase methods are most successful for the problems with only a few local minima and the problem should have enough structure which facilitates the use of efficient local

search. Their main difficulty is that they may find the same local minimum many times.

Tunneling methods (Levy and Montalvo, 1985) also come under two-phase methods which consist of a tunneling phase (global phase) and a local phase. The local phase finds an improved point in the neighborhood of the initial guess by using a standard local optimization technique. The tunneling phase explores the new regions of attraction by calculating the zeros of the tunneling function, which are then used as a new initial guess for the local optimization technique. The cycle of tunneling and local phases are repeated for specified number of iterations and the current overall optimum to date is declared as a global optimum.

Most of the two-phase methods can be viewed as the variants of the multi-start algorithm, which consists of generating a sample of points from a uniform distribution over the feasible region. The multi-start algorithms provide asymptotic convergence to the global minimum, which is the fundamental convergence that is provided by all stochastic methods. Clustering methods (Boender and Romeijn, 1995), developed mainly to improve efficiency of the multi-start algorithms, try to identify the different regions of attraction of the local optima, and start a local search from each region of attraction. It identifies different regions of attraction by grouping mutually close points in one cluster. Clusters are formed in a step-wise fashion, starting from a seed point, which may be the unclustered point with the lowest function value or the local optimum found by applying a local optimization technique to the starting point. Points are then added to the cluster through application of a clustering rule. The two most popular clustering techniques are density clustering (Rinnooy Kan and Timmer,

1987a and b) and single linkage clustering (Timmer, 1984). In density clustering, the objective function is approximated by a quadratic function whereas the single linkage clustering does not fix any shape to the clusters *a priori*.

Multi-Level Single-Linkage (MLSL) algorithm combines the computational efficiency of clustering methods with the theoretical merits of multi-start algorithms. In this method, local search procedure is applied to every sample point, except if there is another sample point within some judiciously chosen critical distance with a better function value. The main task in the MLSL is choosing the critical distance such that the method converges with minimum effort. The above two-phase methods aim at finding all the local optima and then select the best one as the global optimum. Hence, these methods do not work well when the function have a large number of local optima.

Random Search Methods: These methods consist of algorithms which generate a sequence of points in the feasible region following some pre-specified probability distribution or sequence of probability distributions (Boender and Romeijn, 1995). They are very flexible such that they can be easily applied to ill-structured problems for which no efficient local search procedures exist. Pure random search (PRS) is the simplest algorithm among the random search methods. It consists of generating a sequence of uniformly distributed points in the feasible region, while keeping track of the best point that was already found. This algorithm offers a probabilistic asymptotic guarantee that the global minimum will be found with probability one as the sample size grows to infinity. Next to PRS in random search methods is the pure adaptive search, which differs from PRS in the way that it forces

improvement in each iteration. In this algorithm, a better region is identified and a uniformly distributed iteration point is generated in it. Number of iteration points needed to find the global optimum by pure adaptive search increases only linearly with dimension. The main difficulties of PRS are constructing the better region and generating a point that is uniformly distributed in it.

The third important algorithm in random search methods is the adaptive search algorithm which differs from the pure adaptive search by generating iteration points from a sequence of Boltzmann distributions. The advantage of this algorithm is that sampling is done from the feasible region instead of from the better regions as defined in the pure adaptive search. This avoids the two main difficulties (i.e., constructing better regions and generating a uniformly distributed point in it) of pure random search. The fourth one is a direct search algorithm proposed by Luus and Jaakola (1973), which uses random search points and systematic region reduction in locating the global optimum. The algorithm has been shown to have high reliability in locating the global optimum for numerous optimization problems (Luus, 2000). Even though there are improvements of each random search algorithm, these algorithms generally require high computational effort.

Random Function Methods: In the Random Function approach (Boender and Romeijn, 1995), the objective function is assumed to be a stochastic process, which is defined as a probability distribution over a class of functions. The stochastic process is mainly characterized by a correlation function between the function values at each pair of points in the feasible region. This correlation function is used to choose the next point to evaluate the function value. The parameters of the correlation function

are estimated during each cycle of the algorithm. Random function approach is useful for problems where the evaluation of objective function is computationally intensive. The difficulty of this algorithm lies in specifying the stochastic process that is consistent with the properties of the actual function (i.e., continuity and/or differentiability) and also in determining the explicit expressions for the correlation function.

Meta-heuristic Methods: Meta-heuristic methods are mainly developed based on the processes observed in physics and biology. Among the many such methods, the most important are Simulated Annealing, Tabu Search, Genetic Algorithms, Differential Evolution, Ant Colony Optimization and Particle Swarm Optimization. *Simulated Annealing* (SA) (Kirkpatrick et al., 1983) is a stochastic method based on analogy to the annealing of metals. Consider a solid crystal that has been heated to molten state and then cooled until it solidifies. If the temperature is reduced rapidly, irregularities appear in the crystal structure of the cooling solid and the energy level of the solid is higher than in perfectly structured crystal. If the material is cooled slowly keeping the temperature at a steady level for some time so that the material can reach thermal equilibrium with its environment, the energy level attained by the solid will be at its minimum. Based on this concept, SA diversifies the search, both in selecting a move/point to evaluate and in deciding whether or not to accept a move. The basic SA algorithm uses the Metropolis criterion to accept a move, in which down hill is always accepted and uphill moves are accepted with a probability of $\exp(-\text{movevalue}/T)$. When the temperature is high, many uphill moves are accepted, thereby preventing the method being trapped in local minima. The advantages of SA are very easy to implement, robust and applicable to a very general

class of optimization problems. The other advantage of SA is it converges to the global minimum in asymptotic time (Yang, 2000); however, care should be taken while choosing the parameter values, which are problem-dependent.

Taboo (or tabu) search (TS) is a meta-heuristic method which has been successfully applied to combinatorial optimization problems (Glover, 1989 and 1990). There are relatively few applications of this method to continuous problems. Taboo search consists of forbidden (taboo) moves which guide and improve the search in the solution space. The main idea is that the set of all candidate solutions generated in a given iteration, should not only depend on the current iteration point but also be modified by comparing them with the candidate solutions in the tabu list and excluding the candidate solutions that are nearer to the points in the tabu list. Thus TS avoids repeated visits to the same points in the search space. After a few iterations, several promising areas are identified and intensive search is carried out from the most promising area. Although TS has computationally proven successful (Pardalos et al., 2000), it does not guarantee the global optimum.

Genetic Algorithms (GA) are developed based on the mechanics of natural genetics and natural selection (Holland, 1975). Their main principle is the 'survival of the fittest', which means that good points are allowed to continue to the next generation while the less desirable points are discarded from further calculations. Initially, GA starts with a population of random points in the feasible region. These random points will undergo three main operations, namely, reproduction, crossover and mutation. In the reproduction, a mating pool, in which the points with good fitness will have more copies than the ones having lower fitness value, is formed.

Crossover is an operation which introduces some randomness into the population such that the algorithm can escape from the local optima. In this operation, new members (points or strings) are formed by exchanging the information among two members of the mating pool. Mutation operation involves making changes in the selected member directly; the main idea is to change the member by a small amount such that local searches will be promoted if there is any local optimum nearby. After reproduction, crossover and mutation, the new members obtained represent the next generation and the same process is repeated many times. Several attempts have been made to improve the efficiency of GA such as compact GA (Harik et al., 1999) which uses less memory compared to the original GA and there are many applications of GA in different disciplines (Babu, 2004).

Differential Evolution (DE) (Storn and Price, 1997) is similar to GA in principle but it is simpler in its structure and easy to implement. It is a population based direct search algorithm, and each generation/cycle of DE consists of mutation, crossover and selection steps as in GA. The basic difference between GA and DE is that the search is guided by crossover in the former whereas it is governed by the mutation in the latter. In DE, a mutant individual for the selected member of the current population (also known as target individual) is generated by adding the weighted difference of two random individuals to a third random individual in the population. The elements of the mutant individual thus obtained are copied to the target individual using crossover constant to produce a trial individual, known as crossover operation. The better one between the trial and target individuals is selected based on a greedy criterion such as the objective function value, for the next generation, and is called as selection operation. The fitness (objective function value)

of the individuals is improved over the generations, finally converging to an optimum. DE is found to be faster compared to GA (Karaboga and Cetinkaya, 2004), and its applications include digital filter design, neural network learning, crystallographic characterization and optimal design of shell and tube heat exchangers (Price and Storn, 2006).

Ant Colony Optimization (ACO) (Dorigo et al., 1996; and Mathur et al., 2000) is developed by mimicking the way ants establish a shortest route from the nest to the food and back. While moving from one place to another in an ant colony, ants communicate themselves by laying down a chemical substance known as pheromone, on its path. For example, an ant lays down varying quantities of pheromone while searching for its destination (foodstuff point). Another randomly moving ant in the colony can detect this path and follows it with a high probability while simultaneously depositing its own pheromone on its path. More pheromone on this path increases the probability of the path being followed by other ants, thus a shortest route is established from the nest to the food and back.

The above concept is implemented in ACO by introducing artificial ants called agents. Initially, the given feasible region of the optimization problem is divided into a specific number of regions, called as nodes. The fitness of these nodes is calculated by evaluating objective function at various randomly generated points, and nodes are sorted based on their fitness. Each ant starts from a randomly selected node and chooses a path based on the amount of pheromone present on possible paths from the starting node (i.e., higher probability for the paths with more pheromone). Each ant then reaches the next node and selects the next path in a similar way and continues

until it reaches starting node, and the whole journey is referred as a cycle. The improvement in the fitness value is analyzed for each cycle and the quantity of pheromone is adjusted accordingly (i.e., path with better fitness values receives more pheromone and vice versa). The number of cycles is repeated until most of the ants select the same path in every cycle i.e., until a converged solution, optimum, is obtained. The reported applications of ACO in chemical engineering include design and scheduling of batch plants (Jayaraman et al., 2000), and dynamic optimization of chemical processes (Zhang et al., 2005).

Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 2001) is a recent bio-inspired technique originated from the social behavior of bird flocking and fish schooling since these mechanisms are found to have an optimizing characteristic. In PSO, a swarm or population of particles fly through the given N-dimensional problem space according to stochastic update rules. Initially, a swarm is generated randomly over the given feasible region. The objective function is evaluated at each particle in the swarm, and as the particles fly through the solution space (i.e., from one iteration to next iteration), each particle remembers its own best position (in terms of function value) that it has ever found, known as local best solution. Each particle also remembers the best solution obtained by any particle in the swarm, known as the global best solution. In every iteration, velocity of each particle in the swarm is updated using both the local and global best solutions obtained from the previous iteration. As the number of iterations increases, the swarm settles down near to a better solution, and is considered as the optimum value obtained. The chemical engineering applications of PSO include dynamic analysis of chemical processes

(Ourique et al., 2002) and process synthesis and design problems (Yiqing et al., 2007).

1.3 Motivation and Scope of the Work

Since the advent of non-linear optimization in late 60's, several developments and applications of global optimization can be found in engineering discipline including chemical engineering. The reason is that mathematical programming is an active and vibrant research in chemical engineering. This trend has existed for several decades and appears to be increasing (Grossmann, 1996). Global optimization methods play a significant role (Floudas, 2000a; Grossmann and Biegler, 2004; Biegler and Grossmann, 2004; and Floudas et al., 2005) in many practical chemical engineering applications, which include heat exchanger network synthesis (Zamora and Grossmann, 1998; Bjork and Westerlund, 2002; and Lin and Miller, 2004a), phase equilibrium problems (McDonald and Floudas, 1997; Nichita et al., 2002a and b; Teh and Rangaiah, 2003; and Burgos-Solorzano et al., 2004), parameter estimation problems (Esposito and Floudas, 1998; Gau and Stadtherr, 2000; Esposito and Floudas, 2000; and Katare et al., 2004), optimization of integrated process design and control (Moles et al., 2003), design of alternative refrigerants (Sahinidis et al., 2003) and solvents (Sinha et al., 1999; and Wang and Achenie, 2002;), phase stability analysis (Hua et al., 1998; Harding and Floudas, 2000; Henderson et al, 2001; and Balogh et al., 2003), process network problems (Queseda and Grossmann, 1995; and Lee and Grossmann, 2001), computer aided molecular design (Patkar and Venkatasubramanian, 2003; and Lin et al., 2005), global optimization of dynamic systems (Jockenhoevel et al., 2003; and Papamichail and Adjiman, 2004;) and process

synthesis, design and control problems (Ryoo and Sahinidis, 1995; Adjman et al., 2000; Floudas, 2000a; Costa and Oliveira, 2001; and Angira and Babu, 2006).

Though deterministic techniques have been used for many applications in chemical engineering, they are mostly problem specific and require additional properties (such as continuity) of the objective function/constraints. On the other hand, stochastic methods are attracting greater attention and interest from several researchers (e.g., Ku and Karimi, 1991; Patel et al., 1991; Salcedo, 1992; Floquet et al., 1994; Garrard and Fraga, 1998; Sundaram and Venkatasubramanian, 1998; Yu et al., 2000; Rangaiah, 2001; Banga et al., 2002; Patkar and Venkatasubramanian, 2003; Banga et al., 2003; Teh and Rangaiah, 2003; Bhushan and Karimi, 2004; Lin and Miller, 2004a and b; and Babu and Angira, 2006) as they (i) generally require little or no additional assumptions on the optimization problem, (ii) can handle non-linear and non-convex objective functions and constraints, and (iii) provide acceptable solutions with probabilistic convergence guarantee to the global optimum. Stochastic methods are particularly attractive for challenging global optimization problems which does not have any known structure, known as black-box optimization models (Pardalos et al., 2000) and problems with expensive objective function evaluations (Jones et al., 1998). Inspired by the potential of stochastic global optimization techniques to address generic, complex optimization problems, this research is devoted to the study of several stochastic optimization methods for chemical engineering applications. The two main objectives and scope of the study are outlined in the following.

(i) Evaluation of Promising Stochastic Global Optimization Techniques for Chemical Engineering Applications

Many chemical engineering problems involve optimization of highly non-linear and non-convex objective function/constraints. As a result, these problems will have multiple solutions and the use of local optimization methods fail to find the global optimum. On the other hand, several stochastic global optimization methods have been proposed and successfully applied to many applications of interest in engineering. In this study, promising stochastic global optimization algorithms namely, RTA (Jiang et al., 2002), TS (Chelouah and Siarry, 2000) and DE (Storn and Price, 1997), which are shown to be efficient and reliable in the literature, have been evaluated for benchmark and chemical engineering applications. Evaluation has been carried out fairly, systematically and thoroughly. Chemical engineering applications studied include phase equilibrium calculations via Gibbs free energy minimization, phase stability analysis using tangent plane distance function and parameter estimation in models using error-in-variables approach. Phase equilibrium problems are important in the design, simulation and optimization of chemical processes whereas parameter estimation problems play a vital role in developing good mathematical models. The examples considered for phase equilibrium calculations and phase stability problems involve multiple components, multiple phases and popular thermodynamic models whereas parameter estimation problems considered involve up to 34 variables.

New Benchmark Problems Similar to Phase Equilibrium Calculations: The challenging feature of phase equilibrium calculations by free energy minimization is

the presence of comparable minima (i.e., function values at a local and global minima are close to each other) in these problems. Although there are many benchmark problems in the literature with characteristics such as flat objective function and huge number of local minima, none of them represents the comparable minima as in phase equilibrium calculations. On the other hand, testing of methods with phase equilibrium problems require chemical engineering background and careful implementation. Hence, motivated from the unique characteristic of phase equilibrium calculations, two new benchmark problems having comparable minima are proposed and employed.

(ii) Developing a Reliable and Efficient Stochastic Global Optimization Technique

Stochastic global optimization methods play an important role in practical global optimization (Zabinsky, 1998) as they are usually quite simple to implement and use, and they do not require transformation of the original problem. Among the many, the popular stochastic methods are: GA, SA, DE and TS. Experience with them shows that each of them has a unique feature of escaping from a local minimum and/or computational efficiency. The main feature of both GA and DE is the ability to escape from a local minimum using crossover and mutation operations especially for highly non-convex and non-differentiable functions. Unlike GA and DE, SA starts from a single point, and is analogous to the physical process known as annealing in which a material changes its state to reach the lowest-energy state. The main feature of SA is its up-hill moves by accepting some inferior points also during the search, in order to escape from the local minima. The main feature of TS is avoiding re-visits to

the same place during the search, thus providing good computational efficiency. All the aforementioned methods have been successfully applied to many applications in engineering and other fields (e.g., Merz et al., 1999; Teh and Rangaiah, 2003; Katare et al., 2004; Faber et al., 2005; and Mayer et al., 2005).Though GA, DE, SA and TS have their own merits, they sometimes get trapped into local minima.

To overcome the drawbacks of each method, several studies have considered integration of stochastic methods such as GA, SA and TS particularly for combinatorial optimization (Kim et al., 1997; and Mantawy et al., 1999). In this study, a careful and comprehensive attempt has been made to identify and combine the strong features of DE, TS and SA, to develop a new method with good reliability as in DE along with good computational efficiency of TS. DE has been chosen instead of GA because it is easy to implement and found to be computationally efficient compared to GA (Karaboga and Cetinkaya, 2004). The concept of TS is implemented using tabu list which keeps track of the previously visited points and eliminates re-visits during the search. An attempt has also been made to combine the concept of SA; however, the performance of the algorithm is not improved. Two versions (one with tabu list in the evaluation step and another with tabu list in the generation step of DE) of the new integrated method are thoroughly tested over many multi-modal test functions, challenging phase equilibrium calculations and parameter estimation problems in differential and algebraic systems. The better version with tabu list in the generation step of DE is then evaluated for many non-linear programming (NLP) and mixed-integer non-linear programming (MINLP) problems often encountered in chemical engineering practice.

A Transformation to Enhance the Reliability of Stochastic Global Optimization Methods: Though stochastic methods have been successfully used for many applications, their reliability may be affected at times due to the highly non-linear and non-convex nature of the objective function. In order to enhance the reliability of these methods, a new transformation of the original objective function is proposed in this study. It transforms the original objective function in such a way that the current local minimum becomes the global maximum while other better local minima are not affected. The effectiveness of the transformation is studied with DE on several test functions; however, the transformation can be used along with any stochastic global optimization methods.

1.5 Outline of the Thesis

There are 7 chapters in this thesis. Following this introduction, Chapter 2 describes the implementation and evaluation of RTA for benchmark functions and chemical engineering applications. Chapter 3 describes the evaluation and comparison of popular stochastic methods, DE and TS for benchmark, phase equilibrium and phase stability problems. Chapter 4 introduces the proposed stochastic method, DE with tabu list (DETL) and its application to a wide range of test functions, phase equilibrium calculations and parameter estimation problems in differential and algebraic systems. The application of DETL to several NLPs and MINLPs is presented in Chapter 5. Chapter 6 describes the proposed transformation to enhance the reliability of stochastic global optimization methods. Finally, conclusions of this study and future works are outlined in Chapter 7.

CHAPTER 2

IMPLEMENTATION AND EVALUATION OF RANDOM TUNNELING ALGORITHM*

Random tunneling algorithm (RTA) is a global optimization algorithm based on the concepts of terminal repeller and unconstrained subenergy tunneling (TRUST) algorithm. Even though RTA and TRUST seem to be efficient and attractive for benchmark problems involving 2 to 8 variables, they have not been evaluated for benchmark problems involving 10 to 20 variables and for chemical engineering applications. In this study, RTA has been implemented with some changes in the global phase and used it to solve the benchmark problems with 2 to 20 variables and a few to hundreds of local minima. The results show that RTA is comparable to or better than many other methods reported in the literature. The potential of RTA for chemical engineering applications, namely, phase equilibrium calculations using Gibbs free energy minimization and parameter estimation in models, is then studied. Phase equilibrium problems considered include vapor-liquid, liquid-liquid and vapor-liquid-liquid examples involving several components and popular thermodynamic models and parameter estimation problems involve up to 34 variables. RTA successfully located global minimum for most of the examples but the reliability of the method is low for problems having a local minimum comparable to the global minimum.

* This chapter is based on the paper – Mekapati Srinivas and Rangaiah, G. P. Implementation and evaluation of random tunneling algorithm for chemical engineering applications, Computers and Chemical Engineering, 30, pp. 1400-1415. 2006.

2.1 Introduction

Global optimization is playing an important role in the chemical engineering applications, particularly in the design of chemical process systems (Floudas, 2000a). In general, global optimization methods can be broadly classified into two categories: deterministic and stochastic (Pardalos et al., 2000). Many global optimization methods, each having its own special features suitable to a particular problem have been developed, and some of the recent unconstrained minimization methods are reviewed below.

Barhen (1997) proposed a deterministic global optimization algorithm known as terminal repeller and unconstrained subenergy tunneling (TRUST). It builds on two main concepts: subenergy tunneling and non-Lipschitzian terminal repellers. The former transforms the objective function in such a way that points whose value is greater than the current local minimum are approximately zero. The transformation preserves all properties relevant for optimization. The repeller makes the system to escape from the current local minimum and explores a new valley with better function value. The algorithm has been applied to benchmark problems up to 3 variables and on a seismology problem. The results of Barhen (1997) show that TRUST is faster than all other methods reported in the literature. RoyChowdhury et al. (2000) proposed hybridization of gradient descent algorithm with dynamic tunneling (GRDT) method for global optimization. This method consists of two phases: iterative phase and initialization phase. The former phase implements steepest descent technique to locate a local minimum whereas the latter uses dynamic tunneling concept to find a better point for next local descent. The algorithm has been tested for

10 benchmark functions involving up to 6 variables. The results show that performance of GRDT is comparable with other methods in the literature.

Oblow (2001) proposed Stochastic Pijavskij Tunneling (SPT) algorithm for global optimization. The method is based on TRUST method and combines a series of local descents with stochastic searches. SPT uses a rejection-based stochastic procedure to locate new local minima and a fixed Lipschitz like constant is used to reject unpromising regions in tunneling. The algorithm can be easily implemented in low dimensional case and is less effective in high dimensional case without further heuristics. Several approximations for parameters of the algorithm are proposed by Oblow (2001) to improve the method. The algorithm has been used to solve a seismology problem and the results show that SPT is competitive with other methods in the literature. Cai and Shao (2002) proposed a fast annealing evolutionary algorithm (FAEA) which combines the concept of population in genetic algorithms (GA) with annealing procedure in simulated annealing (SA). A fast annealing technique is used in FAEA and a diversification strategy is applied during every annealing step in order to avoid entrapment in local minima. The algorithm has been applied to bench mark problems and Lennard-Jones molecular clusters up to 13 atoms. The results show that reliability and efficiency of FAEA is high when compared with modified genetic algorithm and SA.

Toh (2002) proposed a global optimization method using monotonic transformation (MTT). The transformation magnifies the relative ordering of global and local minima, and the regions that contain global minima are identified using level sets. Then the original unconstrained problem is reformulated as a constrained

one and is solved using penalty function method. The method has been applied to training an artificial neural network problem and the results show that it is reliable in locating the global minimum. Groenwold and Snyman (2002) proposed two global optimization algorithms to minimize an unconstrained objective function through the modeling of dynamic search trajectories. The first one is Snyman-Fatti (SF) algorithm which models the trajectory of a motion of a particle of unit mass in a n-dimensional conservative force field. The second one is modified bouncing ball (MBB) algorithm that models the trajectory of a projectile in a conservative gravitational field. The above two methods are modified to increase the likelihood of convergence to a lower local minimum. The performance of MBB is effective for Rastrigin and Shubert functions whereas SF algorithm is effective for Shekel family of functions.

Jiang et al. (2002) proposed a random tunneling algorithm (RTA) which is also based on TRUST method. It is a two-phase optimization method in which a global phase is carried out by random tunneling and a local phase by gradient optimization with BFGS (Broyden, Fletcher, Goldfarb and Shanno) method. In the global phase, the system is perturbed randomly from the last local minimum and a set of simultaneous differential equations are solved to find a better point to begin the descent again. The method is used to evaluate the Lennard-Jones molecular clusters problem and the results show that RTA located global minimum of Lennard-Jones clusters of up to 20 atoms successfully.

The computational efficiency (in terms of number of function and gradient evaluations) of the above global optimization methods for several test functions as reported by researchers in the literature is summarized in Table 2.1. The numbers in

the Table 2.1 represent the function and gradient evaluations required to locate the global minimum of the corresponding test functions. Even though TRUST and SPT seem to be faster, they did not include the gradient calls and are not tested for functions having more than 3 variables. From Table 2.1, it is clear that RTA is efficient among all other methods. Motivated from the above comparison (Table 2.1), RTA has been chosen for thoroughly evaluating its performance for benchmark and challenging chemical engineering applications.

RTA is mainly based on the concepts of subenergy transformation and terminal repeller in the terminal repeller and unconstrained subenergy tunneling (TRUST) 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 former phase, a minimization algorithm such as gradient descent or Newton's method is used to minimize the given objective function f(x) to locate a local minimum x^*. In the tunneling 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 for the next cycle, and the process is repeated sequentially until a stopping criterion such as failure to find a zero within the prescribed CPU time is met. The last local minimum thus found is declared to be the global minimum.

Table 2.1: Comparison of computational efficiency of different global optimization methods

Function*	BR	CA	GP	RA_2	RA_5	SH	H_3	H_6	GW_2	GW_8	Reference
FAEA	4850	---	5125	5964	---	6700	5280	18, 090	---	---	Cai and Shao, 2002.
MTT	57	44	---	363	---	239	213	787	---	---	Toh, 2002.
SF	3922	426	2092	4799	---	8491	933	1025	---	---	Groenwold and Snyman, 2002.
MBB	286	213	592	1873	---	1057	973	499	---	---	Groenwold and Snyman, 2002.
TRUST #	55	31	103	59	---	72	55				Barhen, 1997
GRDT	466	290	5175	822	---	502	2811	963	---	---	RoyChowdhury, 2000.
SPT #	67	26	123	140	---	150	75	---	---	---	Oblow, 2001.
RTA	23	135	113	383	687	202	60	196	281	465	Jiang et al., 2002.

Note: The number in each cell represents the function and gradient calls required to locate the global minimum of the test function by an algorithm, and blank (---) represents no results are available in the literature. * BR - Branin, CA - 6 hump camelback, GP - Goldstein and Price, RA_2 and RA_5 - Rastrigin 2 and 5 variables, SH - Shubert, H_3 and H_6 - Hartman 3 and 6 variables, GW_2 and GW_8 - Griewank 2 and 8 variables. # Gradient calls are not included in this method.

Different from the above approach, TRUST (Cetin et al. 1993) is a deterministic algorithm which has been applied successfully in exploratory seismology. The method introduces two concepts, namely, subenergy tunneling and non-Lipschitzian terminal repellers, which in turn make the search to escape from the last local minimum. The subenergy tunneling transforms the objective function f(x) as

$$E_{sub}(x, x^*) = \log\left(\frac{1}{1+\exp(-(\hat{f}(x)+a))}\right) \tag{2.1}$$

where a is a positive constant, $\hat{f}(x) = f(x) - f(x^*)$ and x^* is the current local minimum. The transformation converts all the points whose value is greater than or equal to the current local minimum to approximately equal to zero and preserves all other points whose value is less than the current local minimum (Figure 2.1). In other words, $E_{sub}(x, x^*)$ has the same critical points and the same relative ordering of the local and global minima as f(x). In Figure 2.1, the dashed line shows the transformation of the objective function f(x) according to equation 2.1. The terminal repeller term is a penalty function that converts the current local minimum to a local maximum. The transformation involved is

$$E_{rep}(x, x^*) = (-3/4)\rho(x - x^*)^{4/3} H(\hat{f}(x)) \tag{2.2}$$

where ρ is a positive constant and H is the heaviside function, which is equal to 1 for positive values of the argument ($\hat{f}(x)$) and is 0 otherwise. Here, x^* is treated as an equilibrium point. The repeller term repels the dynamic system ($\frac{dx}{dt} = g(x) = \rho(x - x^*)^{1/3} H(\hat{f}(x))$) from the equilibrium point, x*, by perturbing to x' which violates the Lipschitz condition ($\left|\frac{\partial g(x^*)}{\partial x}\right| < +\infty$). Thus the system escapes from

the current local minimum in finite time. Finally, the virtual objective function combines both subenergy tunneling and terminal repeller terms as follows.

$$E(x,x^*)=E_{sub}(x,x^*)+E_{rep}(x,x^*) \tag{2.3}$$

The dynamic system that is to be solved is the gradient of $E(x,x^*)$ which is given by

$$\dot{x}_i=-\left(\frac{\partial f(x)}{\partial x_i}\right)\left(\frac{1}{1+\exp(\hat{f}(x)+a)}\right)+\rho(x-x^*)^{1/3}H(\hat{f}(x)); \quad i=1,2,...,N \tag{2.4}$$

where N is the number of variables in the given problem.

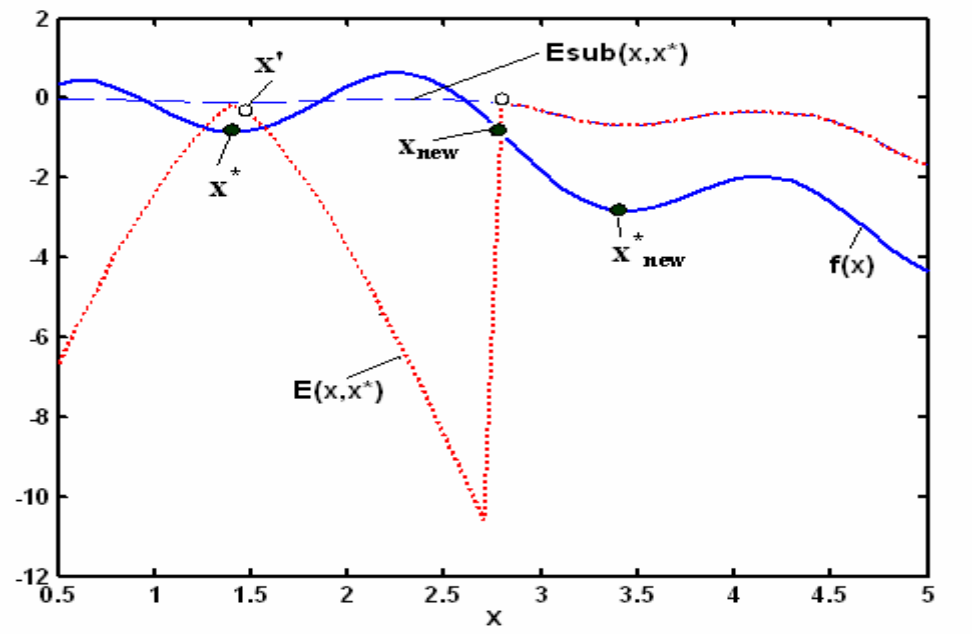


Figure 2.1: Schematic diagram of TRUST showing the original function, f(x), subenergy transformation, E_{sub} (x, x^*) in equation 2.1 and virtual objective function, E (x, x^*) in equation 2.3

The tunneling phase of the algorithm starts by perturbing the system from the current local minimum x^* to x' (as shown in Figure 2.1). The perturbation is carried out by adding a small positive value d to the local minimum ($x' = x^*+d$). Then the repeller term makes the dynamic system flow down over the virtual surface escaping

from x^* to find a better point x_{new} (Figure 2.1) where the argument of the Heaviside function is negative. Then the local minimization is started from x_{new} to find the next local minimum x_{new}^*. The algorithm starts with one of the boundary points as a starting point and terminates when the dynamic system flows out of the other boundary. Thus TRUST sweeps the whole domain in one dimensional case and finds the global minimum.

The implementation of TRUST (Barhen et al. 1997) in multi-dimensional case differs from one dimensional case by solving gradient of subenergy tunneling function (equation 1) in the local minimization phase and performing one-dimensional (1D) search in the tunneling phase along each dimension. For the later, one has to integrate the following differential equation in the tunneling phase

$$\dot{x}_i = \rho(x_i - x_i^{\mu^*})^{1/3} H[f(\hat{x}_i) - f(x^{\mu^*})]; \quad i = 1, 2, \ldots, N \tag{2.5}$$

where the last local minimum is $x_i^{\mu^*}$ and $\hat{x}_i$ represents that all components of x except i are fixed. The algorithm terminates when no better point $((f(\hat{x}_i) - f(x^{\mu^*})) < 0)$ is found in the tunneling phase and declares the last local minimum as the global minimum. A direct extension of 1D search to multi-dimensional optimization does not guarantee that the global minimum will always be found (Cetin et al., 1993; Barhen et al., 1997). Diller and Verlinde (1999) have studied TRUST algorithm for the purpose of molecular docking along with other algorithms and reported that the performance of TRUST was not up to expectation.

Even though TRUST and RTA seem to be efficient and attractive for benchmark problems involving 2 to 8 variables (Barhen et al., 1997; Jiang et al., 2002), they have not been evaluated for benchmark problems involving 10 to 20

variables (Table 2.1) and for chemical engineering applications. In this work, RTA is implemented with some changes in the global phase and without introducing the population, and then tested for many benchmark problems and for selected chemical engineering applications such as phase equilibrium calculations and parameter estimation in models. The changes in global phase include the random perturbation and implementing uniform grid search instead of solving a system of differential equations. The population is not introduced because application problems that are solved consist of fewer local minima (less than the minima in benchmark problems) and Jiang et al. (2002) too did not use population for solving benchmark problems.

2.2 Implementation of Tunneling

RTA has been implemented in a slightly different way here for achieving computational efficiency. The first change in the global phase which is also known as tunneling phase is in the random perturbation from the current local minimum x^*. The random perturbation of Jiang et al. (2002) is given by

$$x_i' = x_i^* + S_i \lambda\, r(x_i^u - x_i^l) \qquad \forall i \in [1, 2, ..., N] \tag{2.6}$$

where x_i^u and x_i^l are the upper and lower bounds of the i^{th} element of the state vector x, λ represents the magnitude of perturbation which can have a value between 0 and 1, r is a random number in the interval 0 and 1, and S_i is a sign variable determining the direction of tunneling with -1 or 1. The above perturbation may generate a point that is out of the feasible region and requires boundary violation check every time the system is perturbed. In order to overcome this problem, random perturbation in this work has been implemented in the following way

$$x_i' = x_i^* + \lambda\, r(ulb_i - x_i^*) \qquad \forall i \in [1, 2, ..., N] \tag{2.7}$$

where ulb_i can be chosen randomly either upper or lower bound of i^{th} element of the x vector. The above perturbation indirectly eliminates S_i in equation 2.6.

The second change is in the type of search implemented in the random tunneling instead of solving the system of differential equations (equation 2.4) from the perturbed point x' as in RTA. Here, uniform grid search has been implemented because the dynamic system which is to be solved approximates to either gradient descent phase or tunneling phase (Cetin et al., 1993) based on the value of $\hat{f}(x)$ as discussed below.

i) When $f(x) \geq f(x^*)$ i.e., $\hat{f}(x) \geq 0$, the exponential term in equation 2.4 rapidly tends to zero (i.e., $\partial E_{sub}(x, x^*)/\partial x \cong 0$). In other words, the subenergy function is nearly flat and approximately zero in magnitude in the neighborhood of x^*. As the magnitude of subenergy function is negligible compared to that of the repeller term, the dynamic system behaves approximately as

$$\dot{x}_i \cong \rho(x - x^*)^{1/3}; \qquad i = 1, 2, ..., N \tag{2.8}$$

ii) When $f(x) < f(x^*)$ i.e., $\hat{f}(x) < 0$, the gradient multiplier term $(1/[1 + \exp(\hat{f}(x) + a)])$ (equation 2.4) is approximately unity, the repeller term is equal to zero and the dynamic system behaves as

$$\dot{x}_i \cong -\left(\frac{\partial f(x)}{\partial x}\right) \tag{2.9}$$

The above analysis shows that when $\hat{f}(x) \geq 0$ (tunneling phase) the system is not using any function information in order to generate the next search point. If any better point is explored (i.e., $\hat{f}(x) < 0$), then RTA stops the tunneling phase and provides

this better point to the local minimization phase. It is also unclear how the dynamic system (equation 2.4) with n first order differential equations has been solved simultaneously in RTA using random step sizes for each dimension (Jiang et al., 2002). Our numerical experience with TRUST of Cetin et al. (1993) for benchmark problems showed that selection of time step in solving the differential equations is important and no guidelines are available for choosing it. In addition, solution of differential equations (equation 2.4) needs gradient evaluation at each step which involves significant computational effort if the gradient is calculated numerically. Based on the previous analysis and due to the unavailability of the TRUST and RTA programs, a simple uniform grid search has been employed in the tunneling phase.

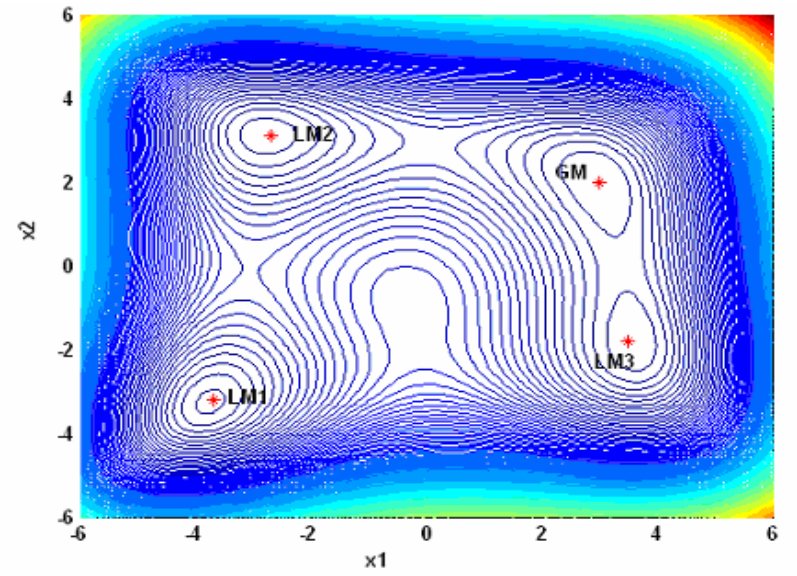
The third change is the insertion of one-dimensional (1D) search along each coordinate axis from the perturbed point (x') in addition to the search towards the random perturbed point from the last local minimum (x^*). All the searches have been performed using uniform grid search. The main reason for adding 1D search is that the random search (i.e., search towards the perturbed point) was unable to solve Rastrigin functions in which the distribution of local minima is along the co-ordinate axes. The direction reversals in RTA are taken care by perturbing the system again from the last local minimum.

The above tunneling technique can be explained from the contour diagrams shown in Figure 2.2a and b. The contour diagram of modified Himmelblau function (mHB) (Deb, 2002) in Figure 2.2a shows that there are three local minima for this function, and the global minimum (GM) is at $x = (3, 2)$ with a function value of zero. The local minima LM1, LM2 and LM3 are at x = (-3.763429, -3.266052), (-2.787063,

3.128204) and (3.581492, -1.820800) with function values 7.367345, 3.487127 and 1.504353 respectively. The mHB function also has one maximum point and two saddle points which are not of interest here.

For example, consider the second local minimum (LM2) as the current local minimum (x^*). The contour plot (Figure 2.2b) shows the better regions that are to be explored. These regions have function values that are lower than $f(x^*)$. Now the tunneling phase starts by perturbing the system from x^* to x' as shown in Figure 2b. Random tunneling (RT) is performed from x^* towards x' and it continues until it finds a better point (i.e. any point in the better regions) or hits the boundary. In this case RT was unable to find a better point and 1D search is started from the perturbed point, x'. As the choice of dimension to be explored first in 1D search can be random, first dimension (along x_1 in Figure 2.2b) is considered as the starting direction in this study. This 1D search successfully identified a point (b in Figure 2.2b) in the better region, thus escaping from the last local minimum. Tunneling phase returns this better point as a new initial guess to the local minimization phase to find an improved point in its neighborhood. As can be seen from Figure 2.2b, both random tunneling and 1D search may fail to explore better regions. In such cases, tunneling phase returns perturbed point (x') as a new initial guess to the local minimization phase and tunneling will be repeated by perturbing the system from the best minimum among the local minima found.

(a)



(b)

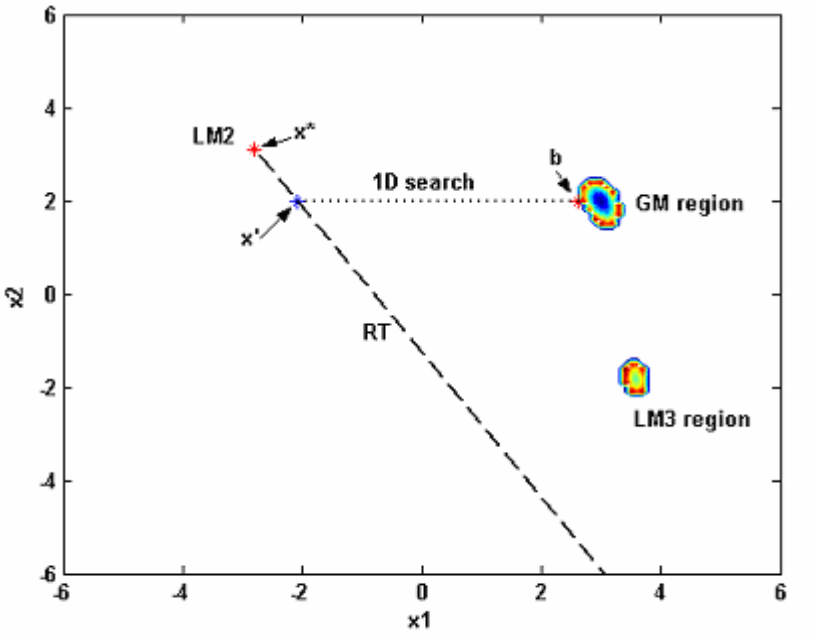


Figure 2.2: (a) Contour diagram and (b) Schematic of tunneling for modified Himmelblau function

2.3 Description of the Algorithm

The main stages of the algorithm are local minimization phase and tunneling phase. The algorithm (Figure 2.3) starts with the selection of values of parameters, namely, magnitude of perturbation (λ), step size (ss) for uniform grid search, maximum number of perturbations during a particular tunneling phase (ps) and the maximum number of tunneling phases (tp). Then a point x is generated randomly in the feasible region and is optimized using a fast convergent quasi-Newton technique to obtain the current local minimum (x^*) in the neighborhood of x. Tunneling phase is started from this local minimum which comprises of three steps. The first step is random perturbation (equation 2.7) from the current local minimum and the second step involves tunneling from the perturbed point using uniform grid search until it hits the boundary. The third step consists of 1D tunneling from the perturbed point along each coordinate axis as follows. Suppose, for example, search is along the first coordinate axis. Then all remaining elements of x (i.e., x_i for i = 2, 3, …, N) are kept at the perturbed point x' and search is performed using specified step size towards the upper bound of the first coordinate until it hits the boundary. After hitting the upper boundary, direction of the search is reversed towards the lower boundary starting from x' and search continues until it hits the boundary. The same process is repeated for all elements of x.

The three steps of the tunneling phase are repeated until the number of perturbations reaches the maximum number (ps) or a better point ($f(x) < f(x^*)$) is found. If any such point is found, then the tunneling phase gets terminated and sends this better point as the new initial guess to the next local minimization phase else the

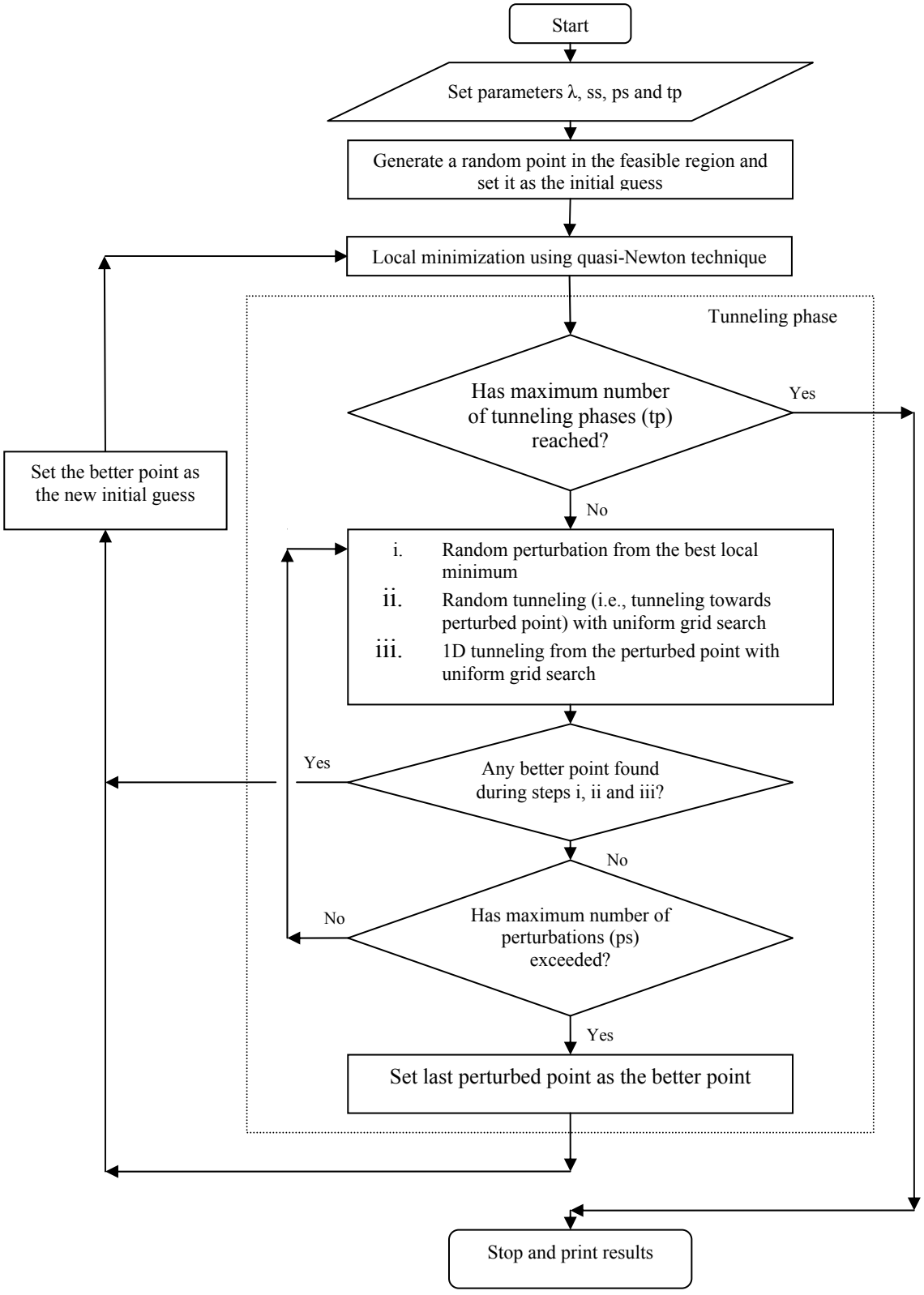


Figure 2.3: Flowchart of RTA implemented in this study

last perturbed point will be the new initial guess. In the local minimization step, 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 (tp). The algorithm then terminates declaring the last/best local minimum as the global minimum.

2.4 Benchmark Problems

The RTA in Figure 2.3 is programmed in FORTRAN with the IMSL subroutine: DBCONF for quasi-Newton method in the local minimization phase. This subroutine calculates the gradient numerically and approximates the Hessian matrix with BFGS update. The RTA is tested for many benchmark problems (Jiang et al., 2002; Deb, 2002; Teh and Rangaiah, 2002 and Trefethen 2002.) with number of variables in the range 2 to 20 and a few to hundreds of local minima. For consistency, all decision variables in each function are normalized to the range [0, 1] using their bounds, in the program. A brief description of the functions is given below and the global minima are in Table 2.2

1. Branin function (BR):

$$f(x) = [x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6]^2 + 10(1 - 1/8\pi)\cos x_1 + 10 \qquad (2.10)$$

where search domain is $-5 \le x_1 \le 10,\ 0 \le x_2 \le 15.$

2 Camelback function (CA):

$$f(x) = [4 - 2.1x_1^2 + (x_1^4/3)]x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \qquad (2.11)$$

where the search domain is $-5 \le x_1, x_2 \le 5.$

3. Goldstein and Price function (GP):

$$f(x) = [1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$$
$$\times[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)] \quad (2.12)$$

where the search domain is $-2 \le x_1, x_2 \le 2$.

4. Rastrigin (RA2, RA5, RA10, RA15 and RA20) function:

$$f(x) = 10n + \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i)) \quad (2.13)$$

where the search domain is $-5.12 \le x_i \le 5.12$ and n = 2, 5, 10, 15 or 20.

5. Shubert function (SH):

$$f(x) = \left\{\sum_{j=1}^{5} j\cos[(j+1)x_1 + j]\right\}\left\{\sum_{j=1}^{5} j\cos[(j+1)x_2 + j]\right\} \quad (2.14)$$

where the search domain is $-10 \le x_1, x_2 \le 10$.

6. Modified Himmelblau function (mHB):

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + 0.1[(x_1-3)^2 + (x_2-2)^2] \quad (2.15)$$

where the search domain is $-6 \le x_1, x_2 \le 6$.

7. Hartmann (H3 and H6) function:

$$f(x) = -\sum_{j=1}^{4} c_j \exp\left[-\sum_{i=1}^{N} a_{ji}(x_i - p_{ji})^2\right] \quad (2.16)$$

where c, a and p are constants whose values are given in Torn and Zilinskas (1989) and the search domain is $0 \le x_i \le 1$.

8. Griewank (GW2 and GW8) function:

$$f(x) = \sum_{i=1}^{N}\frac{x_i^2}{d} - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (2.17)$$

where d = 200 and 600 for N = 2 and 8 respectively and the search domain is $-600 \le x_i \le 600$.

9. Rosenbrock (ROS10, ROS15 and ROS20) function:

$$f(x) = \sum_{i=1}^{N-1}\left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right] \quad (2.18)$$

where the search domain is $-5 \le x_i \le 10$ and n = 10, 15 or 20.

10. Zakharov (Z10, Z15 and Z20) function:

$$f(x) = \left(\sum_{i=1}^{N} x_i^2\right) + \left(\sum_{i=1}^{N} 0.5ix_i^2\right)^2 + \left(\sum_{i=1}^{N} 0.5ix_i^2\right)^4 \quad (2.19)$$

where the search domain is $-5 \le x_i \le 10$ and n = 10, 15 or 20.

11. Trefethen problem:

$$f(x,y) = \exp(\sin(50x)) + \sin(60e^y) + \sin(70\sin(x)) + \sin(\sin(80y)) - \sin(10(x+y)) + \frac{1}{4}(x^2 + y^2) \quad \ldots (2.20)$$

where the search domain is $-1 \le x, y \le 1$.

The efficiency of RTA for the above benchmark functions is evaluated based on number of function and gradient calls (NFG) required to obtain the global minimum. The termination criterion is achievement of the global minimum with an accuracy of 10^{-6} in the function value (absolute difference). All these are chosen for comparing performance of RTA in this study with the performance results of global optimization methods in the literature.

Table 2.2: Global minimum for benchmark problems

Function	Dimension (N)	Global minimum	Remarks
BR	2	0.397887 at x = {9.42478, 2.475}	Three global minima
CA	2	-1.031628 at x = {0.089842,-0.712656}	Six local minima
GP	2	3 at x = {0, -1}	Four local minima
RA	2, 5,10,15,20	0 at x = {0, ..., 0}	More than 50 local minima
mHB	2	0 at x = {3, 2}	Four local minima
SH	2	-186.7309 at x= {0.0217, -0.9527}	18 global minima; 760 local minima
H3	3	-3.86278 at x = {0.11464, 0.555649, 0.852547}	Four local minima
H6	6	-3.3223 at x= {0.201, 0.150, 0.477, 0.275, 0.311, 0.657}	Four local minima
GW	2, 8	0 at x = {0, …, 0}	Several hundreds of local minima
ROS	10, 15, 20	0 at x = {1, …..,1}	Several local minima
Z	10, 15, 20	0 at x = {0,…..,0}	Several local minima
Trefethen	2	-3.306868 at x = {-0.024403} and y = {0.210612}	Several hundreds of local minima

2.5 Phase Equilibrium Problems

One important application of global optimization is phase equilibrium calculations which play a significant role in the design, simulation and optimization of chemical processes. Development of robust and efficient methods for the calculation of phase equilibrium has long been a challenge and still it is. Basically, methods for

multiphase equilibrium calculations can be categorized into two types. The first type is the simultaneous equation-solving approach and the second one is Gibbs free energy minimization approach. In the former approach, a set of non-linear equations arising from mass balances and phase equilibrium relations are solved whereas Gibbs free energy function is minimized in the second approach. The equality of chemical potential criterion, which is only the necessary condition for free energy minimization, is used in the equation-solving approach which makes the method inadequate for finding the global minimum. A significant effort has been expended in equation-solving approach (e.g., Ohanomah and Thompson, 1984a, b and c; Joulia et al., 1986; Bullard and Beigler, 1993; and Teh and Rangaiah, 2002). The modern trend is the treatment of phase equilibrium problems by direct minimization of Gibbs free energy. The difficulty of phase equilibrium calculations arises due to the highly non-linear and non-convex form of the objective function, which gives no guarantee that the global minimum will be found by local optimization methods. Thus, a global optimization method is needed for reliable multiphase equilibrium calculations.

Gibbs free energy minimization for phase equilibrium calculations is first introduced by White et al. (1958). Since then several researchers have used different optimization techniques for phase equilibrium problems and a comprehensive review of these techniques can be found in Teh and Rangaiah (2003). Recently, Nichita et al. (2002a and b) applied the two-phase tunneling method of Levy and Montalvo (1985) for phase equilibrium calculations. In this study, the calculation of multiphase equilibria is organized in a stepwise manner which combines phase stability analysis by minimization of the tangent plane distance function with phase splitting calculations. The results show that the tunneling method is efficient and reliable for

solving multiphase equilibria and the stability problems. However, tunneling method was not used for free energy minimization in a single step, which is more challenging.

Teh and Rangaiah (2002) compared selected methods of equation solving approach and free energy minimization approach for phase equilibrium calculations. Typical examples covering vapor-liquid equilibrium (VLE), liquid-liquid equilibrium (LLE) and vapor-liquid-liquid equilibrium (VLLE) problems with popular thermodynamic models are studied in this paper. The results show that genetic algorithm (GA) followed by modified simplex method of Nelder and Mead (NM) is more reliable and desirable than equation solving methods when there are multiple minima especially for VLLE calculations. Teh and Rangaiah (2003) studied a version of Taboo search (TS) namely, enhanced continuous TS (ECTS) for phase equilibrium calculations via Gibbs free energy minimization. The results show that both TS and GA have high reliability in locating the global minimum, and that TS requires less number of function evaluations than GA.

The expression for total free energy, G can be simplified for different situations by eliminating constant terms for a particular system (Teh and Rangaiah, 2002). If vapor and liquid phases are described by different thermodynamic models, then the dimensionless free energy for a non-reacting system is given by

$$\frac{G_I}{RT} = \sum_{\substack{k\in np \\ (liquidphaseonly)}} \sum_{i=1}^{nc} n_i^{L(k)} \left[\ln\left(x_i^{L(k)} \gamma_i^{L(k)} P_i^{sat} \right) \right] + \sum_{i=1}^{nc} n_i^{V} \left[\ln\left(y_i \hat{\varphi}_i^{V} P \right) \right] \qquad (2.21)$$

where the superscript, $L(k)$ and V refer to liquid phase k and vapor phase respectively. The first term in the above equation refers to only liquid phases

assuming activity coefficient models. For equation of state models describing all liquid and vapor phases, the above equation can be written as

$$\frac{G_{II}}{RT} = \sum_{\substack{k \in np \\ (liquid phase only)}} \sum_{i=1}^{nc} n_i^{L(k)} \left[\ln\left(x_i^{L(k)} \hat{\varphi}_i^{L(k)} \right) \right] + \sum_{i=1}^{nc} n_i^{V} \left[\ln\left(y_i \hat{\varphi}_i^{V} \right) \right] \tag{2.22}$$

If only liquid phases exist, then equation 2.21 can be simplified by ignoring P_i^{sat} (which does not change for a particular example) and the second term for vapor phase can be eliminated. Then the equation becomes

$$\frac{G_{III}}{RT} = \sum_{k=1}^{np} \sum_{i=1}^{nc} n_i^{L(k)} \left[\ln\left(x_i^{L(k)} \gamma_i^{L(k)} \right) \right] \tag{2.23}$$

In the equations 2.21 – 2.23, decision variables are number of moles of component i in phase k (n_i^k), np is the number of phases at equilibrium, nc is the total number of components in the system, γ_i^k and $\hat{\varphi}_i^k$ are the activity and partial fugacity coefficients of component i in phase k respectively. P_i^{sat} and y_i are the saturated vapor pressure and mole fraction in vapor phase of component i. More information about the formulation can be found in Rangaiah (2001).

At physical equilibrium (i.e., without any chemical reaction), moles of each component should be conserved and the number of moles of each component (n_i^k) should be non-negative. Hence, free energy minimization should satisfy the following constraints and bounds.

$$\sum_{k=1}^{np} n_i^k = z_i F \qquad i = 1,2,...,nc \tag{2.24}$$

$$0 \le n_i^k \le z_i F \qquad i = 1,2,...,nc \tag{2.25}$$

The above constrained minimization problem can be transformed into an unconstrained minimization problem by introducing the variables β_i^k (for i = 1,2,…,

np-1; k = 1,2,…, np) instead of mole numbers n_i^k (for i = 1,2,…, nc; k = 1,2,…, np) as the decision variables. The new variables, β_i^k are bounded between 0 and 1 and are related to the mole numbers n_i^k by

$$n_i^1 = \beta_i^1 z_i F \qquad i = 1, 2, ..., nc \tag{2.26}$$

$$n_i^k = \beta_i^k (z_i F - \sum_{j=1}^{k-1} n_i^j) \qquad i = 1, 2, ..., nc;\ k = 2, ..., np-1 \tag{2.27}$$

$$n_i^{np} = (z_i F - \sum_{j=1}^{np-1} n_i^j) \qquad i = 1, 2, ..., nc \tag{2.28}$$

The equality constraints are eliminated by the introduction of equations 2.26 – 2.28 thus reducing the number of decision variables from nc×np to nc×(np-1). The bounds on the decision variables are set as:

$$1 \times 10^{-15} \le \beta_i^k \le 1.0 \qquad i = 1, 2, ..., nc \tag{2.29}$$

The lower bound of the variables is taken as 1×10^{-15} instead of 0 to avoid the numerical difficulties associated with the Gibbs free energy function when the number of moles of a component in a phase is equal to zero.

Phase equilibrium examples considered in this study include VLE, LLE and VLLE examples involving multiple components (2 to 10 components) and popular thermodynamic models. The feed composition, operating conditions and the thermodynamic models used for each example are given in Tables 2.3a-2.3c. Global minimum and local minima for all these examples are given in Teh and Rangaiah (2003). Along with local minima, there are trivial solutions for several examples at which equilibrium composition equals to the feed composition (Teh and Rangaiah, 2002).

Table 2.3a: Selected examples for vapor-liquid equilibrium (VLE)

No.	Mixture	Feed	Pressure & Temperature	Model	Reference
1	Methane (1), propane (2)	F = 1.0 mole; z_i = {0.68, 0.32}	100 bar, 277.6 K	SRK	Hua et al. (1998)
2	Carbon dioxide (1), methane (2)	F = 1.0 mole z_i = {0.20, 0.80}	60.8 bar, 220 K	PR	Hua et al. (1998)
3	Hydrogen sulfide (1), methane (2)	F = 1.0 mole z_i = {0.0187, 0.9813}	40.53 bar, 190K	SRK	Hua et al. (1998)
4	Nitrogen (1), methane (2), ethane (3)	F = 1.0 mole z_i = {0.15, 0.30, 0.55}	76 bar, 270 K	PR	Hua et al. (1998)
5	Methane (1), carbon dioxide (2), Hydrogen sulfide (3)	F = 1.0 mole z_i = {0.4989, 0.0988,0.4023}	48.6 bar, 227.55 K	PR	Sun & Seider (1995)
6a	Benzene (1), acetonitrile (2), water (3)	F = 1.0036 mole z_i = {0.34359, 0.30293, 0.34718}	0.1 atm, 300 K	Ideal (V) NRTL (L)	Castillo & Grossman (1981)
6b	Benzene (1), acetonitrile (2), water (3)	F = 1.0036 mole z_i = {0.34359, 0.30293, 0.34718}	0.1 atm, 300 K	Ideal (V) UNIFAC (L)	---
7	Nitrogen (1), argon (2), oxygen (3)	F = 9.38529 moles z_i = {0.78112, 0.00930, 0.20958}	607.95kPa, 100.79 K	PR	PRO/II (1993)
8	Methane (1), ethane (2), propane (3), i-butane (4), n-butane (5), i-pentane (6), n-pentane (7), n-hexane (8), n-pentadecane (9)	F = 0.96890 moles z_i = {0.61400, 0.10259, 0.04985, 0.00898, 0.02116, 0.00722, 0.01187, 0.01435, 0.16998}	19.84 atm, 314 K	SRK	Castillo & Grossman (1981)
9	Mixture of 10 hydrocarbons (for more details, see Teh and Rangaiah, 2002)	F =35.108 moles	3998.98 kPa, 287.48 K	PR	Hyprotech (1998)

Table 2.3b: Selected examples for liquid-liquid equilibrium (LLE)

No.	Mixture	Feed	Pressure & Temperature	Model	Reference
10a	n-butyl-acetate (1), water (2)	F = 1.0 mole z_i = {0.50, 0.50}	1.0 atm, 298 K	NRTL	Heidemann & Mandhane (1973)
10b	n-butyl-acetate (1), water (2)	F = 1.0 mole z_i = {0.50, 0.50}	1.0 atm, 298 K	UNIFAC	McDonald & Floudas (1997)
11	Toluene (1), water (2)	F = 1.0 mole z_i = {0.50, 0.50}	1.0 atm, 298 K	NRTL	Castillo & Grossman (1981)
12	Furfural (1), 2,2,4-trimethyl pentane (2), Cyclohexane (3)	F = 1.0 mole z_i = {0.10, 0.10, 0.80}	1.0 atm, 298 K	UNIQUAC	Prausnitz et al. (1980)
13	Toluene (1), water (2), aniline (3)	F = 0.9987 mole z_i = {0.29989, 0.20006,0.50005}	1.0 atm, 298 K	NRTL	Castillo & Grossman (1981)

Table 2.3c: Selected examples for vapor-liquid-liquid equilibrium (VLLE)

No.	Mixture	Feed	Pressure & Temperature	Model	Reference
14	Benzene (1), acetonitrile (2), water (3)	F = 1.0 036 moles z_i = {0.34359, 0.30923, 0.34718}	0.769 atm, 333 K	Ideal (V) NRTL (L)	Castillo & Grossman (1981)
15	Methanol (1), methyl acetate (2), water (3)	F = 1.0 mole z_i = {0.15, 0.45, 0.40}	0.965 atm, 325 K	Ideal (V) UNIFAC (L)	McDonald & Floudas (1997)
16	Ethanol (1), benzene (2), water (3)	F = 1.0 mole z_i = {0.20, 0.35, 0.45}	1.0 atm, 338 K	Ideal (V) UNIFAC (L)	Prokopakis & Seider (1983)

2.6 Parameter Estimation Problems

Parameter estimation problems are important in developing good mathematical models that are used in the design, control and optimization of chemical processes. Often, non-linear models are necessary for simulating chemical processes. There are many methods such as method of least squares and maximum likelihood criterion for estimating the parameters in non-linear models (Bard, 1974). Due to the highly non-linear nature of the models, the optimization problem becomes non-convex that results in local minima. Hence, standard methods using local minimization techniques to solve these problems can not provide the best set of values for model parameters (Gau and Stadtherr, 2000). Thus global optimization is needed for parameter estimation problems. Two parameter estimation problems: modeling of VLE (Stadtherr, 2002) of a binary system: benzene (1) - hexaflourobenzene (2), and Kowalik problem (Moore et al., 1992), both having multiple minima, are considered in this study.

An error-in-variables formulation (Schwetlick and Tiller, 1985; Esposito and Floudas, 1998; and Stadtherr, 2002) which accounts for errors in all measured variables rather than in the dependent variables only, was used for the VLE parameter estimation problem. Wilson equation is used to describe the non-ideality of liquid in VLE. The objective function is

$$\varphi(\theta, \tilde{x}_1, \tilde{T}) = \sum_{i=1}^{m} \left[\frac{(\tilde{x}_{1,i} - x_{1,i})^2}{\sigma_{x1}^2} + \frac{(\tilde{T}_i - T_i)^2}{\sigma_T^2} + \frac{(\tilde{y}_{1,i}(\theta, \tilde{x}_{1,i}, \tilde{T}_i) - y_{1,i})^2}{\sigma_{y1}^2} + \frac{(\tilde{P}_i(\theta, \tilde{x}_{1,i}, \tilde{T}_i) - P_i)^2}{\sigma_P^2} \right] \quad \ldots(2.30)$$

where the decision variables are energy parameters (θ_1 and θ_2), mole fractions of component 1 in the liquid phase ($\tilde{\mathbf{x}}_1$) and system temperature ($\tilde{\mathbf{T}}$) at different

measurements (m). $x_{1,i}, y_{1,i}, P_i$ and T_i are the measured data from the experiments (Stadtherr, 2002). $\sigma_{x1}, \sigma_{y1}, \sigma_P$ and σ_T are the standard deviations associated with the measurement of the respective variables in the subscript. $\tilde{y}_1$ and $\tilde{P}$ can be calculated from

$$\tilde{y}_1 = \frac{\gamma_1 \tilde{x}_1 p_1^0(\tilde{T})}{\gamma_1 \tilde{x}_1 p_1^0(\tilde{T}) + \gamma_2 (1 - x_2) p_2^0(\tilde{T})} \tag{2.31}$$

$$\tilde{P} = \gamma_1 \tilde{x}_1 p_1^0(\tilde{T}) + \gamma_2 (1 - \tilde{x}_1) p_2^0(\tilde{T}) \tag{2.32}$$

where $p_1^0(\tilde{T})$ and $p_2^0(\tilde{T})$ are pure component vapor pressures given by Antonie equation, and γ_1 and γ_2 are the activity coefficients given by Wilson equation. More information about the problem can be found in Stadtherr (2002).

The first two terms on the right hand side of equation 2.30 are quadratic and convex in nature but the remaining two terms are highly nonlinear and non-convex with respect to decision variables. The variables are θ_1, θ_2, $\tilde{x}_{1,i}$ and $\tilde{T}_i$ $(i = 1,2,...,m)$ thus resulting in 2m+2 variables. Here, $m = 16$ and the total number of variables is 34. The bounds over variables are chosen using plus and minus three standard deviations of measurement error (i.e., $\tilde{x}_{1,i} \in [x_{1,i} - 3\sigma_x, x_{1,i} + 3\sigma_x]$ and $\tilde{T}_i \in [T_i - 3\sigma_T, T_i + 3\sigma_T]$). The bounds over θ $(\theta \in [-1000, 2000])$ are taken in order to cover both local minima but is narrower compared to the original interval used by Stadtherr (2002). The function values at local and global minimum for 34 variable problem are respectively 161.3 and 19.999 (Stadtherr, 2002). The problem is also solved by taking temperature as constant in which case the number of variables is 18. For the reduced problem, the local and global minimum value is found to be 237.7569

Table 2.4: Global and local minima for VLE parameter estimation problem with 18 variables

Number	Global minimum	Local minimum
i	$\mathbf{x_{1,i}}$	$\mathbf{x_{1,i}}$
1	0.086732	0.085284
2	0.131225	0.129663
3	0.161842	0.160430
4	0.170457	0.169120
5	0.218721	0.218025
6	0.261657	0.261784
7	0.356726	0.359261
8	0.365194	0.367967
9	0.404097	0.408022
10	0.542825	0.550108
11	0.638787	0.646827
12	0.704562	0.711688
13	0.830610	0.832218
14	0.857743	0.857756
15	0.911106	0.908110
16	0.959139	0.955368
	θ	**θ**
θ_1	-435.184	272.370
θ_2	1052.632	-273.696

Note: Function values at local and global minima are 237.7569 and 26.7209 respectively.

and 26.7209 respectively. These stationary points are confirmed to be minima by calculating gradient and Eigen values of the Hessian matrix. The global and local minimum for 34 and 18 variables examples are given in Stadtherr (2002) and in Table 2.4 respectively.

The objective function for Kowalik problem (Moore et al., 1992) is given by

$$f(x_1, x_2, x_3, x_4) = \sum_{i=1}^{11} \left(a_i - x_1 \left(\frac{b_i^2 + b_i x_2}{b_i^2 + b_i x_3 + x_4} \right) \right)^2 \tag{2.33}$$

where x_1, x_2, x_3 and x_4 are the parameters to be optimized, and a and b are constants (Moore et al., 1992). The values of local minima and global minimum are available in Esposito and Floudas (1998).

2.7 Results and Discussion

2.7.1 Parameter Tuning

Benchmark problems CA, GP, SH, H_3, H_6, GW, R_{15} and R_{20} are used for tuning the parameters of RTA, which are magnitude of perturbation (λ), step size (ss) and maximum number of perturbations (ps). The parameter, maximum number of tunneling phases (tp), is not included in the tuning because achieving the global minimum (with an accuracy of 10^{-6} in function value) is used as the termination criterion for comparison with other methods in the literature. The parameters are tuned to find the global minimum with good success rate and less number of function and gradient evaluations (NFG). Tuning is carried out by varying one parameter at a time while keeping all other parameters at their nominal values. The nominal values

are chosen as $\lambda = 0.5$, ss = 0.1 and ps = 1 based on the preliminary experience with RTA by solving some benchmark problems. The optimal parameter values obtained are $\lambda = 1$, ss = 0.1 and ps = 1 except for Rastrigin (RA_5, RA_{10}, RA_{15} and RA_{20}) functions for which $\lambda = 0.001$ and ss = 0.01. The change in the optimal values of λ and ss for Rastrigin functions is due to the distribution of local minima along the coordinate axes which makes it difficult for tunneling phase to explore better regions especially in high dimensional problems. The use of 1D search along with small magnitude of perturbation (λ) provides an advantage in exploring these better regions. With a small λ, the perturbed point will be near the local minimum and 1D search successfully identifies the better region even if random tunneling fails.

The parameters for phase equilibrium problems are tuned by selecting examples 10a, 14 and 15 which are found to be difficult by Teh and Rangaiah (2002). The tuning is carried out by varying one variable while keeping others fixed at their nominal values. The nominal value for tp is chosen as 20 based on the preliminary experience with benchmark problems. The optimal parameter values obtained for phase equilibrium problems are $\lambda = 1$, ss = 0.02, ps = 1 and tp = 15. The optimal step size (ss) obtained is smaller compared to benchmark problems (optimum ss = 0.1), which may be due to the presence of a few but comparable minima in phase equilibrium problems. As the minima are comparable, better regions will be very small. To explore these regions in the tunneling phase, a smaller step size in the uniform grid search is needed.

For the parameter estimation problems studied, the optimal parameter values (λ, ss and ps) obtained are the same as those for benchmark problems. The maximum

number of tunneling phases (tp) obtained is 5 for both VLE parameter estimation problems and Kowalik problem. Even though the number of variables is high in VLE parameter estimation problems when compared to the phase equilibrium problems, the step size is high and the maximum number of tunneling phases is low. This may be due to the presence of local and global minima with quite different function values contrary to the comparable minima in phase equilibrium problems. As the local and global minima have quite different function values, it is expected that better region with respect to local minimum will be large enough such that even a large step size (ss) and a few number of tunneling phases are sufficient to explore them.

For comparison purposes, the parameter estimation problems have also been solved by genetic algorithm followed by quasi-Newton minimization (GA-QN) and tabu search followed by quasi-Newton minimization (TS-QN). More information about the implementation of GA-QN and TS-QN can be found in Rangaiah (2001) and Teh and Rangaiah (2003) respectively. The parameters of these algorithms are tuned using both VLE modeling problems and Kowalik's problem. The optimal parameter values available in the literature (Teh and Rangaiah (2003) for TS-QN and Rangaiah (2001) for GA-QN) are taken as the nominal values and tuning is carried out for each parameter while keeping the rest fixed at their nominal values. The optimal parameters obtained for TS-QN are: N_t and N_p (tabu and promising list size) = 10; ε_t and ε_p (tabu and promising radius) = 0.05; N_{ip} (initial population size) = 15N where N is the dimension of the problem; h_n (length of the hyper-rectangle) = 0.75; and $Iter_{max}$ (maximum number of iterations) = 50N; and for GA-QN, they are: NP (population size) = 15N; P_{cross} (probability of crossover) = 0.8; P_{mut} (probability of mutation) = 0.6 and Gen_{max} (maximum number of generations) = 35N.

2.7.2 Performance of RTA for Benchmark Problems

The benchmark problems are solved 100 times, each time by generating different initial estimate randomly in the feasible region. The achievement of global minimum with desired accuracy (i.e., absolute difference in the function value is 10^{-6}) is considered as the termination criterion for comparison with other methods and the gradient of the function is calculated numerically. All the results are summarized in Table 2.5a and b, and are compared in terms of NFG required to locate the global minimum, with TRUST, gradient descent with dynamic tunneling (GRDT), fast annealing evolutionary algorithm (FAEA), tunneling method and RTA (Jiang et al., 2002) in the literature. The blank in this table indicates no results are available in the literature for the respective functions. The results of RTA (this work), RTA (Jiang et al., 2002) and tunneling method include gradient calls whereas the results of FAEA (does not require gradient), reported results of TRUST did not include gradient calls and no information is available for GRDT in RoyChowdhury et al. (2000). The results in Table 2.5a represent the average over 100 trials for RTA (this work), RTA (Jiang et al., 2002) and FAEA. The results of TRUST and tunneling method are the average of 2^n (where n is the number of variables) trials and average over 4 trials respectively, and no information is available for GRDT. The results show that RTA (this work) is slightly inefficient compared to TRUST but better than GRDT, FAEA and tunneling method. The computational efficiency of RTA (this work) is low compared to TRUST probably because the later did not include the gradient calls (which could be more than one-half of the number of function calls) and also the results are the average of maximum 8 trials. Results for TRUST and RTA (Jiang et al., 2002) could not be validated by us as their programs and the parameters employed are not available.

Table 2.5a: Average NFG for benchmark problems (< 10 variables)

Function	TRUST[#]	GRDT	FAEA	Tunneling method	RTA (Jiang et al., 2002)	RTA (this work)
BR	55	466	394	---	23	24
CA	31	290	303	13,891	135	76
SH	72	502	446	1,644	202	301
GP	103	5,175	490	---	113	130
mHB	---	---	---	---	---	209
RA_2	59	822	544	---	383	656
RA_5	---	---	2,762	---	687	2,092
H_3	---	2,811	488	---	60	63
H_6	---	963	2,229	---	196	146
GW_2	---	---	7,804	---	281	1,306
GW_8	---	---	---	---	465	2,381
Trefethen	---	---	---	---	---	21,081

#Gradient calls are not included in this method. Results for TRUST, GRDT, FAEA and Tunneling method are taken from Barhen et al. (1997), RoyChowdhury et al. (2000), Cai and Shao (2002) and Levy and Montalvo (1985) respectively.

Table 2.5b: Average NFG for benchmark problems (≥ 10 variables)

Function	RTA (this work)
ROS_{10}	399
RA_{10}	9,604
Z_{10}	238
ROS_{15}	606
RA_{15}	23,511
Z_{15}	437
ROS_{20}	646
RA_{20}	41,726
Z_{20}	673

The NFG for the RTA (this work) is comparable with RTA (Jiang et al., 2002) for most of the two dimensional functions except for Rastrigin and Griewank functions. This is due to the presence of many local minima and their distribution in these problems. As the number of local minima is high, number of tunneling phases will be more, which includes 1D search along all coordinate axes contributing to more NFG. Even though coordinate search in the tunneling phase contributes to more NFG, for certain functions like Rastrigin, convergence to the global minimum could not be achieved using random tunneling alone. This may be due to the distribution of local minima along coordinate axes in Rastrigin functions, which made us to include 1D search in the tunneling phase. Even though Shubert function has 760 local minima, NFG is less because it does not have comparable minima (i.e., function values at local and global minima are close to one another) like in Rastrigin function; also, it has many global minima (18) at different locations and finding any one of them is sufficient. This causes the gradient based minimization technique (quasi-Newton method) to locate the global minimum easily thus requiring less NFG. In case of Rastrigin function, local minimization traps in one of the local minima and the tunneling phase faces difficulty in exploring better regions thus requiring larger NFG. The NFG for Trefethen problem is high because of the associated huge number of local minima.

The results for the benchmark problems with greater than or equal to 10 variables are given in Table 2.5b. Results are given only for RTA (this work) as no information is available in the literature on TRUST, GRDT, FAEA, and RTA (Jiang et al., 2002) for these functions. All the results are average over 100 trials. As the number of variables is increasing NFG is increasing for these functions especially for

Rastrigin functions. This is due to the increase in complexity with more number of variables. The NFG for Rosenbrock and Zakharov functions are lower even for 20 variables when compared to Rastrigin functions, probably due to lower number of local minima when compared to Rastrigin functions.

2.7.3 Performance of RTA for Phase Equilibrium Calculations

Each one of these examples is solved 25 times, each time starting from a different point randomly chosen in the entire feasible region. All the examples are solved with RTA using two types of termination criterion: one with convergence to the global minimum (with an accuracy of 10^{-6} in function value) referred as RTA-GM and another with general stopping criterion (i.e., the algorithm terminates when the maximum number of tunneling phases (tp), is reached) referred as RTA-GS. In RTA-GM, the algorithm terminates whenever it finds the global minimum or the number of tunneling phases reaches 600. The results are given in Tables 2.6a and 2.6b which include both objective function calls and the function calls required for the gradient (NFE), which is calculated numerically. The average NFE is calculated based on the successful runs only. For all problems, the number of phases at equilibrium is chosen correctly and the present study is limited to calculation of phase compositions and number of moles of each phase at equilibrium. Results by RTA are compared with those of TS-QN and GA-QN, which are taken from Teh and Rangaiah (2003). The comparison is made in terms of reliability (success rate i.e., number of times the algorithm successfully identified global minimum out of 25 trials) and computational efficiency (NFE required in locating the global minimum).

Two-phase equilibrium (VLE and LLE) calculations

The results for two-phase equilibrium examples are given in Table 2.6a and the success rate is 100% for all the methods unless otherwise stated. RTA-GM located the global minimum for all the examples with 100% success rate except for example 9. This is because of comparable minima in this example. The reliability and computational efficiency of RTA-GM are better when compared with RTA-GS. NFE of RTA-GM is 3 to 33 times less than RTA-GS. This is because in case of general stopping criterion, the algorithm has to run for the maximum number of tunneling phases (tp) even though it finds the global minimum in just a few number of tunneling phases. To improve the efficiency, RTA-GS is tried with tp = 10 also. As the maximum number of tunneling phases decrease (from 15 to 10), success rate and NFE also decrease. This is because, as tp decreases, search points for all the tunneling phases and number of local minimization phases decrease resulting in low success rate and low NFE. Hence, tp = 15 is needed for better reliability.

The success rate of RTA-GS (with tp = 15) is 100% for all the examples except for example 12 (96% success rate). This is due to the presence of comparable minima (function values at global minimum and at trivial solution are -0.360353 and -0.354340 respectively) which poses difficulty in identifying better regions in the tunneling phase. For example 9, all methods failed to locate the global minimum due to the very close objective function value (-161.5364) at trivial solution to the function value at global minimum (-161.5416). The reliability of RTA-GS is comparable to both TS-QN and GA-QN; although its computational efficiency is

Table 2.6a: Average NFE for VLE and LLE examples

Number	RTA-GM	RTA-GS		TS-QN	GA-QN
		tp = 10	tp = 15		
1	206	2,034	3,345	1,412	20,017
2	496	1,674[a]	2,513	1,349	20,018
3	390	1,801[b]	2,806	1,187	20,024
4	1,071	2,807[c]	4,068	1,777	20,238
5	130	2,882	4,393	1,511	20,027
6a	275	3,919	5,907	1,616	20,089
6b	270	3,878	5,800	1,648	20,088
7	786	3,124[b]	4,605	1,894	20,054
8	1,118	14,555	21,924	10,040	20,515
9	Trivial	Trivial	Trivial	Trivial	Trivial
10a	469	1,834	2,744	1,425	20,018
10b	99	2,163	3,249	1,369	20,024
11	104	2,229	3,346	1,367	20,026
12	1,212	2,668[c]	4,052[a]	1,571	20,028[d]
13	610	3,427	5,166	1,719	20,069

[a] Success rate = 96%; [b] Success rate = 92%; [c] Success rate = 84%; [d] Success rate = 76%.

lower than TS-QN, it is better than GA-QN for two-phase equilibrium examples tested. NFE of RTA-GS is 4 to 7 times less than GA-QN but is 2 to 4 times more than TS-QN. NFE of RTA-GS for example 8 is more than that with GA-QN, and is mainly due to the use of general stopping criterion in this example. Even though RTA-GS has located the global minimum in just one tunneling phase in example 8, the algorithm continues for maximum number of tunneling phases (tp) which increases the NFE.

Three-phase equilibrium (VLLE) problems

The results for three phase equilibrium (VLLE) problems are given in Table 2.6b. RTA-GM located the global minimum for all the examples with 100% success rate. The reliability of RTA-GM is better than RTA-GS but computational efficiency is less compared to RTA-GS. NFE by RTA-GM are 1.2 times less, 1.7 and 5.04 times more than RTA-GS for examples 14, 15 and 16 respectively. The main reason for more NFE is the presence of comparable minima in these examples (Teh and Rangaiah, 2003). Example 14 has a constrained minimum with function value -1.407855 compared to the global minimum with function value -1.408518. For examples 15 and 16, function values at local and global minima are -1.334404 and -1.334461, and are -1.233294 and -1.235896 respectively. As the minima are comparable, better region as shown in Figure 2.2b becomes smaller and smaller making it extremely difficult for tunneling phase to explore. This increases the number of tunneling phases (around 50 for example 14 to 550 for example 16) to explore better regions thus contributing to more NFE.

RTA-GS is evaluated with both tp = 15 (optimal maximum number of tunneling phases) and 50 to observe its effect on success rate and NFE. As the maximum number of tunneling phases increase (from 15 to 50), both success rate and NFE increase. This is due to the intensive search resulting from more number of tunneling phases. The reliability of RTA-GS in locating the global minimum for these examples is low when compared to TS-QN and GA-QN even though computational efficiency of RTA-GS is comparable to TS-QN and is better than GA-QN. This is due to the comparable minima in these examples.

The results of RTA are compared with Simulated Annealing (SA) and tunneling method for some examples available in literature. The results of SA taken from Rangaiah (2001) are average over 100 trials but no such information is provided for results of tunneling method in Nichita et al. (2002). The stopping criterion used for SA is similar to that of RTA-GS (i.e., maximum number of iterations) but is different for tunneling method. The success rate of SA for examples 8 and 10a is 100%, and is low (52%) for example 15 due to comparable minima in that example. The NFE of SA for these examples (8, 10a and 15) is 134767, 34543 and 99737 respectively, and is high compared to that of RTA. This is because of the different escaping mechanisms from local minima in the corresponding methods. SA climbs the hills to explore potential (global) regions whereas RTA tunnels the hills thus providing good computational efficiency compared to SA.

Table 2.6b: Average number of function evaluations and success rate: VLLE examples

No.	RTA-GM	RTA-GS				TS-QN		GA-QN	
		tp = 15		tp = 50					
	NFE	Success rate	NFE	Success rate	NFE	Success rate	NFE	Success rate	NFE
14	7,711[#]	56	9,442	92	31,512	100	5,615	100	20,212
15	15,526[#]	52	8,696	64	28,547	100	5,627	84	20,254
16	36,856[#]	12	7,308	36	24,375	80	5,479	100	20,262

[#] Success rate is 100%.

Although the stopping criterion was different for RTA and tunneling method, comparison is made for completeness. The NFE took by tunneling method for examples 2, 3 and 4 are 739, 645 and 4831 respectively. NFE of tunneling method is low compared to RTA for examples 2 and 3, and probably because of the way these examples were solved by the corresponding methods. Nichita et al. (2002a) used a two-step procedure: solving the corresponding phase stability problem (minimization of tangent plane distance function (TPDF) by tunneling method) followed by a flash calculation (by Newton's method) using the solutions obtained from TPDF minimization whereas RTA has been applied directly to the minimization of Gibbs free energy function. Generally, the non-linearity and the number of variables of phase stability problems are less compared to those of corresponding Gibbs free energy functions. Apart from the work of Nichita et al. (2002a), there have been a few successful applications of tunneling method of Levy and Montalvo (1985) in the literature. These applications include Barron et al. (1996), Kanzow (2000), Merlitz and Wenzel (2002) and Gomez et al. (2003) but it was not clear whether these problems have comparable minima as in phase equilibrium problems**.**

2.7.4 N-dimensional Test Function

To provide a benchmark problem with a global minimum very comparable to the next higher minimum (similar to phase equilibrium problems), a modified n-dimensional test function is developed in this work and is given by

$$f(x) = (1/2)\sum_{i=1}^{N}(x_i^4 - 16x_i^2 + 5x_i) - \alpha\sum_{i=1}^{N}(x_i + 2.90353)^2 \qquad (2.34)$$

where search domain is $-5 \leq x_i \leq 5$ and the function has 2^n minima including the global minimum which is at $x_i = -2.90353$ for i = 1, 2, …, n. This function is developed from the n-dimensional test function given in Cetin et al. (1993). As α value changes (0, 0.3 and 0.4304) in equation 2.34, the minima become comparable and are given in Table 2.7 for 4-variables case.

Table 2.7: Minima for the modified n-dimensional test function (4variables)

Number	α = 0	α = 0.3	α = 0.4304
1	-114.254506	-143.564453	-156.653489
2	-114.254506	-143.564453	-156.653489
3	-114.254506	-143.564453	-156.653489
4	-114.254506	-143.564453	-156.653489
5	-128.391225	-147.931190	-156.657214
6	-128.391225	-147.931190	-156.657214
7	-128.391225	-147.931190	-156.657214
8	-128.391225	-147.931190	-156.657214
9	-128.391225	-147.931190	-156.657214
10	-128.391225	-147.931190	-156.657214
11	-142.527944	-152.297926	-156.660938
12	-142.527944	-152.297926	-156.660938
13	-142.527944	-152.297926	-156.660938
14	-142.527944	-152.297926	-156.660938
15	-100.117787	-139.197716	-156.649765
16	-156.664663	-156.664663	-156.664663

At $\alpha = 0.4304$, the minima are comparable; the fractional change in the function value from the global minimum to the nearest local minimum is around 0.005% to 0.001% compared to around 0.2 to 0.003% in phase equilibrium problems. The contour plots of the original and modified N-dimensional test functions are shown in Figures 2.4 and 2.5. For plotting, bounds over the variables are tightened to $-4 \leq x_i \leq 4$ for clarity while retaining all the minima. The contours of both the test functions are similar except change in the function values. For the one dimensional case, the function values at local minimum (LM in Figure 4) are -25.029446 and -39.162441 for the original test function and modified one respectively. The global minimum (GM in Figure 2.4) is at function value -39.166165 for both the test functions. For the two dimensional case, the function values at the local minima (LM1, LM2 and LM3 in Figure 5) are -50.058893, -64.195612, and -64.195612 for the original test function, and are -78.324882, -78.328607, and -78.328607 for the modified one. The global minimum (GM in Figure 2.5) is at function value -78.332331 for both the test functions. Besides these minima, these functions also have one maximum and 4 saddle points (Figure 2.5), which are not of interest here.

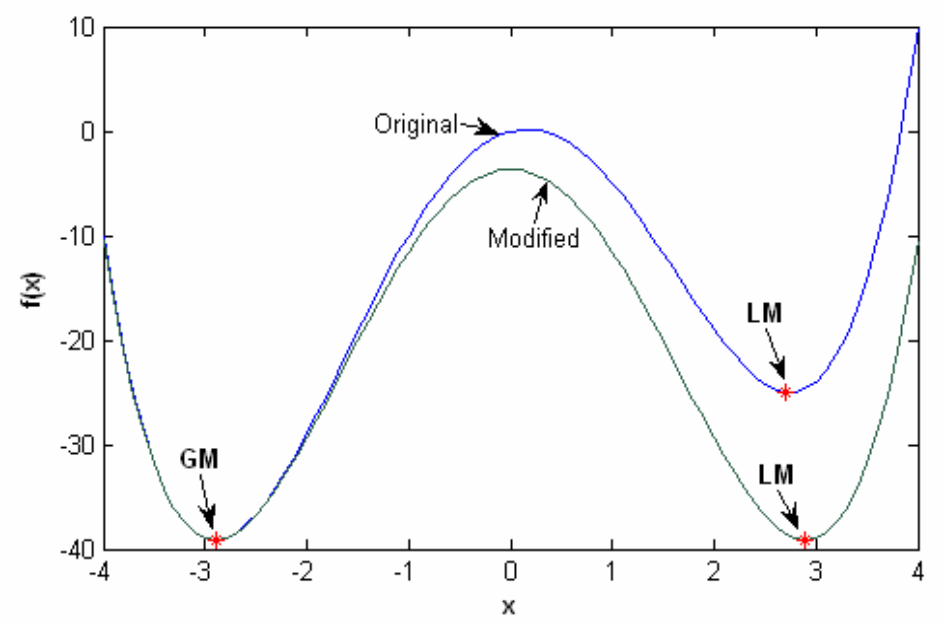
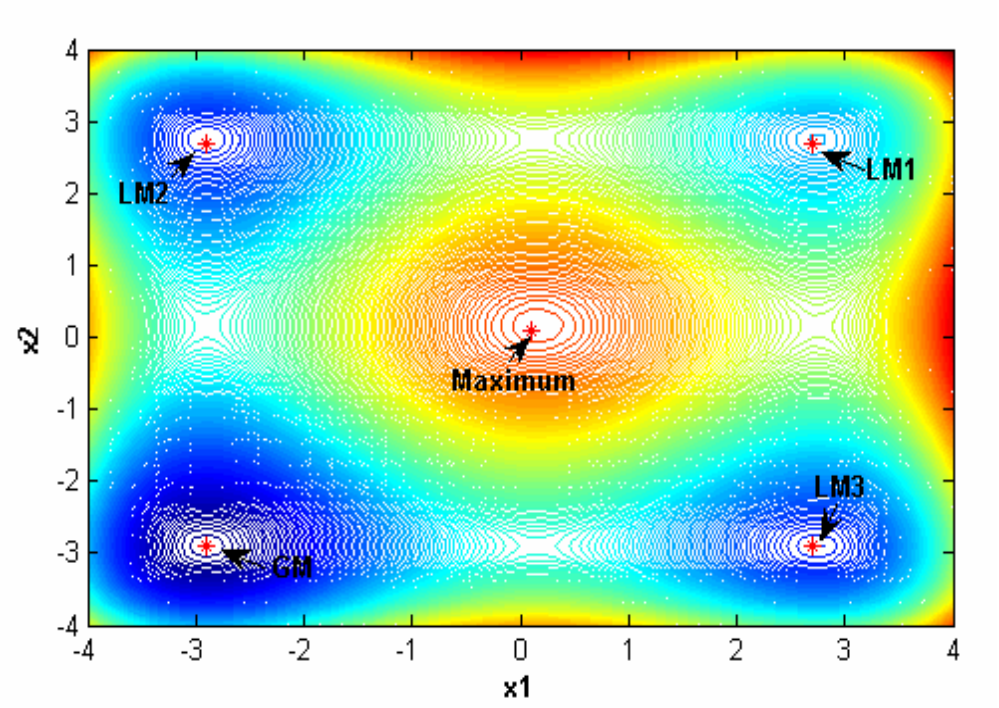


Figure 2.4: Original and modified N-dimensional test functions in one dimensional case

(a)



(b)

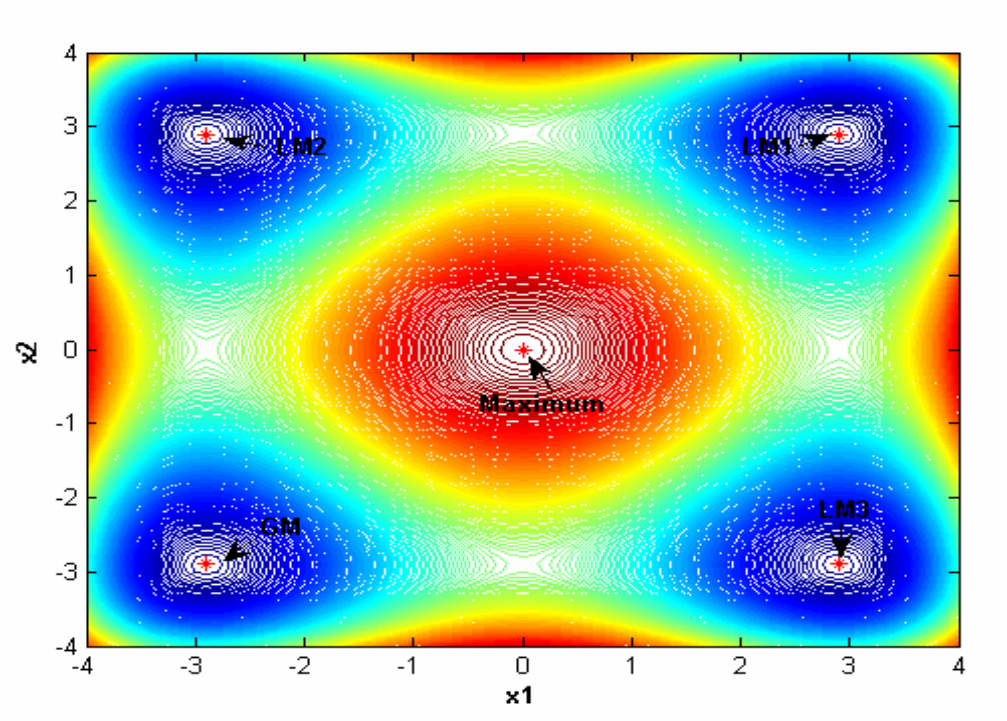


Figure 2.5: Contour plots of (a) Original and (b) Modified test functions in two dimensional case

Table 2.8: Average NFE and success rate of RTA (with general stopping criterion): modified N-dimensional test function

Variables	α = 0			α = 0.3			α = 0.4304		
	Difference*	Success rate	NFE	Difference*	Success rate	NFE	Difference*	Success rate	NFE
2	18.0	100	2,650	5.57	100	2,688	4.75×10^{-3}	99	2,839
4	9.02	91	5,496	2.79	81	5,634	2.37×10^{-3}	68	5,833
6	6.02	46	8,784	1.86	33	9,042	1.58×10^{-3}	22	9,209
8	4.51	16	12,590	1.39	10	12,922	1.19×10^{-3}	8	13,012
10	3.61	5	16,727	1.11	4	17,146	9.51×10^{-4}	1	17,354

* Difference represents the percentage change in function value from the global minimum to the nearest local minimum.

RTA with general stopping criterion is then tested on the modified n-dimensional test function with the optimal parameter values obtained for the phase equilibrium problems. The function is solved 1000 times, each time with a randomly generated initial estimate, for more precise results. The results are given in Table 2.8 in terms of success rate and NFE required in locating the global minimum. Average NFE is calculated based on the successful runs only. As the minima become comparable (i.e., $\alpha = 0$ to 0.4304), success rate decreases. This is because better regions become smaller and tunneling phase faces difficulty in exploring them. As the number of variables increases, the complexity increases resulting in low success rate for problems having more than 4 variables. The success rate is 100% in case of 2 variables (same as in phase equilibrium examples 1, 2, 3, 10a and 10b) but decreases as the number of variables increases (as in examples 14, 15 and 16).

2.7.5 Performance of RTA for Parameter Estimation Problems

Both VLE modeling problems and Kowalik's problem are solved 25 times, each time with a randomly generated different initial estimate. All the examples are solved with both RTA-GM, RTA-GS**,** GA-QN and TS-QN**.** The results are given in Table 2.9 and the success rate of all the methods is 100% unless otherwise stated. NFE for these problems include function calls for both objective function and gradient evaluations, and are the average over successful runs only**.** The corresponding CPU times for each problem are given in brackets. Note that the computer system employed in this study is Pentium 4 (CPU 2.8 GHz, 512MB 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 243.

Table 2.9: Average NFE and CPU time for parameter estimation problems

Examples	RTA-GM	RTA-GS		TS-QN	GA-QN
		tp = 3	tp = 5		
VLE – 18 variables	78,113 (3.07)	189,155 (7.78)	305,864 (12.38)	20,349 (1.57)	174,473 (9.51)
VLE – 34 variables	348,696 (13.97)	723,171[a] (31.50)	1,150,505 (47.89)	40,789 (3.79)	617,628 (47.08)
Kowalik problem	474 (0.002)	1,166 (0.004)	1,834 (0.007)	1,501 (0.012)	6,177 (0.066)

Note: [a] Success rate is 92%.

All methods have successfully solved the parameter estimation problems with 100% success rate. The reliability of RTA-GM and RTA-GS for VLE problems is high even though they involve 18 to 34 variables. This is because function values at the local and global minima are quite different which allows the tunneling phase to explore them successfully. The results show that TS-QN is superior compared to both RTA and GA-QN in terms of computational efficiency. The NFE of RTA-GM is less than that of GA-QN, and that of RTA-GS is higher compared to GA-QN. The CPU time taken by RTA-GS (tp = 3) for VLE problems is less compared to GA-QN even though NFE of the former is more than that of the latter. This could be because of additional steps required for crossover and mutation in GA-QN. In a similar way, the time taken by RTA-GM for VLE -18 variables problem is less than that of TS-QN for VLE – 34 variables problem even though the latter took less NFE compared to that of former. This may be because of extra time taken for comparison of each search point with those of tabu and promising lists in TS-QN. As expected, NFE increases with number of variables from 18 to 34 for all the methods. Results for Kowalik's problem show that the reliability of all methods including RTA is very good with 100% success rate. NFE for solving Kowalik's problem by interval analysis is 545,900

(Moore et al., 1992), which is very high compared to that of RTA, TS and GA methods. VLE modeling problems and Kowalik problem are also evaluated by RTA with tp = 3 along with the optimal maximum number of tunneling phases (i.e., tp = 5) to observe its effect on success rate and NFE required to locate the global minimum. The effect is as expected – success rate and NFE increase with tp.

2.8 Summary

RTA was implemented and evaluated for benchmark, phase equilibrium and parameter estimation problems. The tunneling phase of RTA was implemented with uniform grid search and 1D search along the coordinate axes. The results for benchmark problems involving 2 to 20 variables with a few to hundreds of local minima show that the computational efficiency of RTA is comparable to or better than other methods in the literature. They also show that number of function calls is high for some functions especially for Rastrigin function due to the presence of more local minima and their distribution along the coordinate axes in those functions. RTA was then tested on two and three-phase equilibrium problems with several components and different thermodynamic models, and using two types of termination criterion: convergence to global minimum (RTA-GM) and general stopping criterion (RTA-GS). The results show that reliability and computational efficiency of RTA-GM are better than RTA-GS. The reliability of RTA-GS is comparable to TS-QN and GA-QN, and its computational efficiency is lower than TS-QN but is better than GA-QN for VLE and LLE examples. The reliability of RTA-GS for VLLE examples is lower than both TS-QN and GA-QN even though its computational efficiency is comparable to TS-QN and is better than GA-QN. This is due to the presence of comparable minima in phase equilibrium problems which make it challenging for the

tunneling phase to explore new regions with lower function value. A benchmark problem having comparable minima as in phase equilibrium problems, was proposed and used for testing RTA. VLE parameter estimation problems having 18 and 34 variables, and minima with different function values, have also been studied. Both RTA-GM and RTA-GS located the global minimum for these examples successfully even though number of variables is high compared to phase equilibrium problems. The four parameter Kowalik problem has also been solved by RTA with 100% success rate. Even though RTA solved many phase equilibrium examples and parameter estimation problems successfully, the reliability of the algorithm needs to be improved especially for VLLE examples and example 9, where all methods (TS-QN, GA-QN and RTA) failed to locate the global minimum. This study shows that RTA is attractive for application problems which do not have comparable global and local minima.

CHAPTER 3

EVALUATION OF DIFFERENTIAL EVOLUION AND TABU SEARCH*

Phase equilibrium calculations (PEC) and phase stability (PS) problems play a crucial role in the simulation, design and optimization of separation processes such as distillation and extraction. The former involve the global minimization of Gibbs free energy function whereas the latter requires the global minimization of tangent plane distance function (TPDF). In this work, two promising global optimization techniques: differential evolution (DE) and tabu search (TS) have been evaluated and compared, for the first time, for benchmark, PEC and PS problems. A local optimization technique is used at the end of both TS and DE to improve the accuracy of the final solution. Benchmark problems involve 2 to 20 variables with a few to hundreds of local minima whereas PEC and PS problems consist of multiple components with comparable minima. PEC involves both vapor-liquid, liquid-liquid and vapor-liquid-liquid equilibria with popular thermodynamic models. The results show that DE is more reliable but computationally less efficient compared to TS for benchmark, PEC and PS problems tested.

* This chapter is based on the paper – Mekapati Srinivas and Rangaiah, G. P. A study of differential evolution and tabu search for benchmark, phase equilibrium and phase stability problems, Computers and Chemical Engineering, 31, pp. 760-772. 2007a.

3.1 Introduction

Phase equilibrium calculations (PEC) and phase stability (PS) problems have to be solved a very large number of times in the design and analysis of chemical processes. For a system with specified components, composition, temperature and pressure, PEC involves the calculation of number of moles of each phase and its composition at equilibrium whereas PS analysis determines the stability of the system. There are mainly two different approaches for PEC: equation solving approach and Gibbs free energy minimization approach. The former involves solving a system of non-linear equations resulting from mass balances and equilibrium relationships whereas the latter involves the minimization of highly non-linear Gibbs free energy function. Even though equation solving approach seems to be faster and simple, the solution obtained in this method may not correspond to the true minimum of Gibbs free energy function. Also, it needs a priori knowledge of phases existing at equilibrium. Hence, Gibbs free energy minimization is the desirable approach for PEC.

The concept of Gibbs free energy minimization was first proposed by White et al. (1958), stating that a necessary condition for a given system to be at equilibrium is that the total Gibbs free energy must be at the global minimum. The objective function in this approach is highly non-linear and non-convex necessitating reliable and efficient global optimization. Several researchers have applied different global optimization methods using this approach and a review of these studies can be found in Teh and Rangaiah (2003). Recently, Nichita et al. (2002a) employed a tunneling method for PEC. In this method, the calculations are organized in a stepwise manner,

combining PS analysis by minimization of tangent plane distance function (TPDF) for phase splitting calculations. The tunneling method has two phases: a local bounded minimization using a limited memory quasi-Newton method and a tunneling phase to find a better initial estimate for the next phase of local minimization. The results show that the tunneling method is efficient and reliable for solving multiphase equilibrium problems. Iglesias-Silva et al. (2003) proposed an algebraic method that includes Gibbs free energy minimization for PEC. The method uses orthogonal derivatives, the tangent plane condition and mass balances to reduce the Gibbs minimization procedure to a task of finding the solution of a system of non-linear equations. The results show that the method has good convergence rate. Burgos-Solórzano et al. (2004) solved the problem by combining reliable deterministic techniques such as interval Newton technique with local optimization procedure. The deterministic techniques validate the results obtained from the local optimization technique and provide corrective feedback until the right result is found. The results show that the procedure is good for high pressure chemical and multiphase equilibrium using cubic equation of state models. In these cited works, there is no rigorous comparison of the proposed methods with others in the literature.

Most often PEC needs a priori information about the number of phases existing at equilibrium. Some of the methods (e.g., Gautam and Seider, 1979) explore all possible number of phases in a systematic manner. Initially the method assumes a small number of phases and checks for the stability of phases. If they are stable the method retains the phases else a new phase will be added. A PS problem can be formulated as either minimization problem or an equivalent non-linear equation solving problem. However, the conventional solution techniques are initialization

dependent and may converge to a local or trivial solution based on the initial guess. Baker et al. (1982) proposed the tangent plane criterion often used for PS analysis. The criterion states that a hypothetical phase is thermodynamically stable provided the tangent plane generated at the given composition lies below the molar Gibbs free energy surface for all the compositions. The problem can be formulated as the minimization of TPDF, which is a non-linear and non-convex objective function requiring global optimization. A review of several works using the tangent plane criterion for PS problems can be found in Rangaiah (2001).

Tessier et al. (2000) implemented interval Newton technique for PS analysis. The examples are modeled by non-random two liquid (NRTL) and universal quasi-chemical (UNIQUAC) thermodynamic models. They also proposed two enhancements for interval Newton method. The results indicate that the computational efficiency of the enhanced methods is better compared to the original one. Nichita et al. (2002b) used a global optimization method namely, tunneling method for PS analysis. The problem has been formulated both in conventional approach (i.e., composition space) and in reduced variable approach. The results show that the method is reliable in solving the PS problems. Balogh et al. (2003) used a modified TPDF such that the zeros of the objective function become its minima, since it is advantageous to search for minima with known zero minimum value. They employed a method namely, stochastic sampling and clustering to locate the minima of the modified TPDF. The results show that the method is able to solve small to moderate size problems in an efficient and reliable way.

However, most of the methods employed for PEC and PS problems are local in nature and relatively few stochastic global optimization techniques have been explored for these problems. Stochastic methods are usually quite simple to implement and use, and they do not require transformation of the original problem. Furthermore, these techniques can locate the vicinity of global solutions with relative efficiency compared to deterministic techniques (Moles et al., 2003). Among the many, differential evolution (DE) (Storn and Price, 1997) and tabu search (TS) (Chelouah and Siarry, 2000) are some of the most promising methods reported in the literature. Even though they have been tested for several applications in chemical engineering and other fields (e.g., Lin and Miller, 2004a and b; Mayer et al., 2005; and Bingul, 2004), they have not been applied to PEC and PS problems except Teh and Rangaiah (2003) who studied PEC problems by TS. Also, DE and TS have not been comprehensively compared for benchmark problems. Hence, in this work, both DE and TS are first evaluated and compared for benchmark problems with 2 to 20 variables but involving a few to hundreds of local minima. The methods are then tested for PEC and PS problems involving multiple components, multiple phases and popular thermodynamic models. The evaluation includes both reliability and computational efficiency using practical stopping criteria.

3.1.1 Differential Evolution

DE (Storn and Price, 1997) is a population based direct search method. The algorithm implemented in this study (Figure 3.1) starts with specifying the parameters, namely, amplification factor (A), crossover constant (CR), type of strategy, population size (NP), maximum number of successive iterations (Sc_{max})

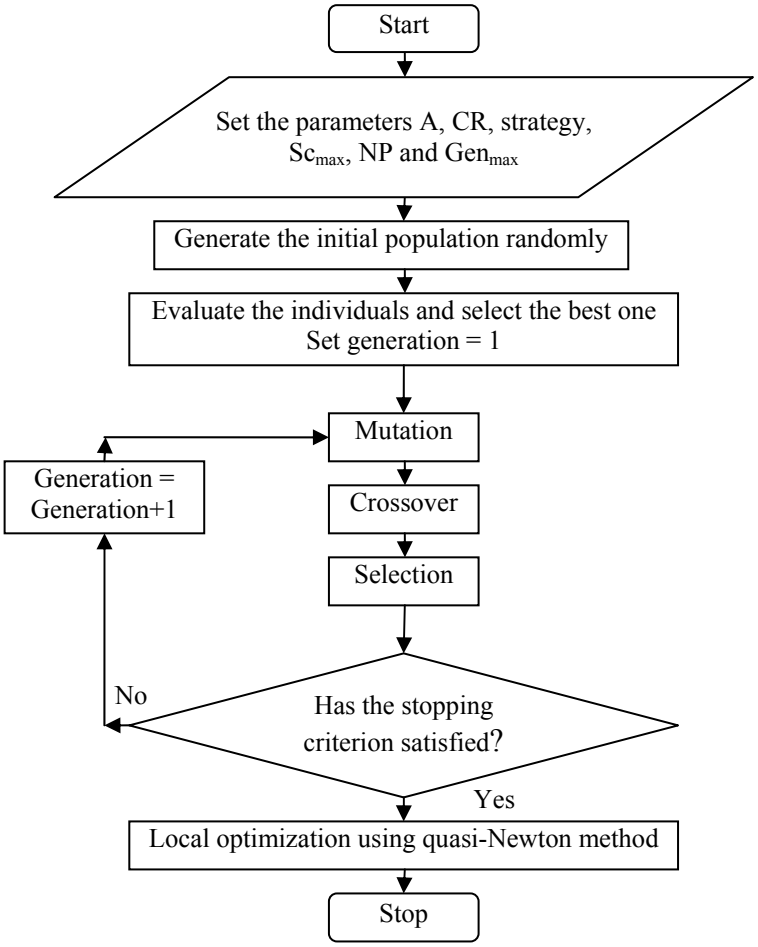


Figure 3.1: Flow chart of DE-QN

without improvement in the best function value and maximum number of generations (Gen_{max}). The initial population is randomly generated using the uniformly distributed random numbers to cover the entire solution space. The individuals are checked for the boundary violation to see if any individual is generated in the infeasible region; the infeasible points are replaced by generating new individuals. The objective function values of all the individuals are calculated and the best point is determined. Then the three main steps: mutation, crossover and selection on the population, are carried out. Mutation and crossover operations are performed to diversify the search thus escaping from the local minima. The mutant vector is generated for each randomly chosen target vector $X_{i, G}$ by

$$V_{i, G+1} = X_{r1, G} + A\,(X_{r2, G} - X_{r3, G}); \qquad i = 1, 2, 3, \ldots, NP. \tag{3.1}$$

where r_1, r_2 and r_3 belongs to the set {1, 2, 3, …, NP} and $X_{r1, G}$, $X_{r2, G}$ and $X_{r3, G}$ represents the three random individuals chosen in the current generation, G, to produce the mutant vector for the next generation, $V_{i, G+1}$. The random numbers r_1, r_2 and r_3 should be different from the running index, i, and hence NP should be ≥ 4 to allow mutation. A is a real value between 0 and 2 which controls the amplification of the differential variation between the two random vectors.

In the crossover step, the trial vector, $U_{i, G+1}$ is produced by copying some elements of the mutant vector, $V_{i, G+1}$ to the target vector, $X_{i, G}$ with probability equal to CR. As illustrated in Figure 3.2, a random number (ran) is generated for each element of the target vector. If ran $\leq$ CR, the element of mutant vector is copied else the target vector element is copied. After mutation and cross over operations, the trial vector competes with the target vector for selection into the next generation. A greedy criterion based on objective function value is used to screen the trial vector. If the trial

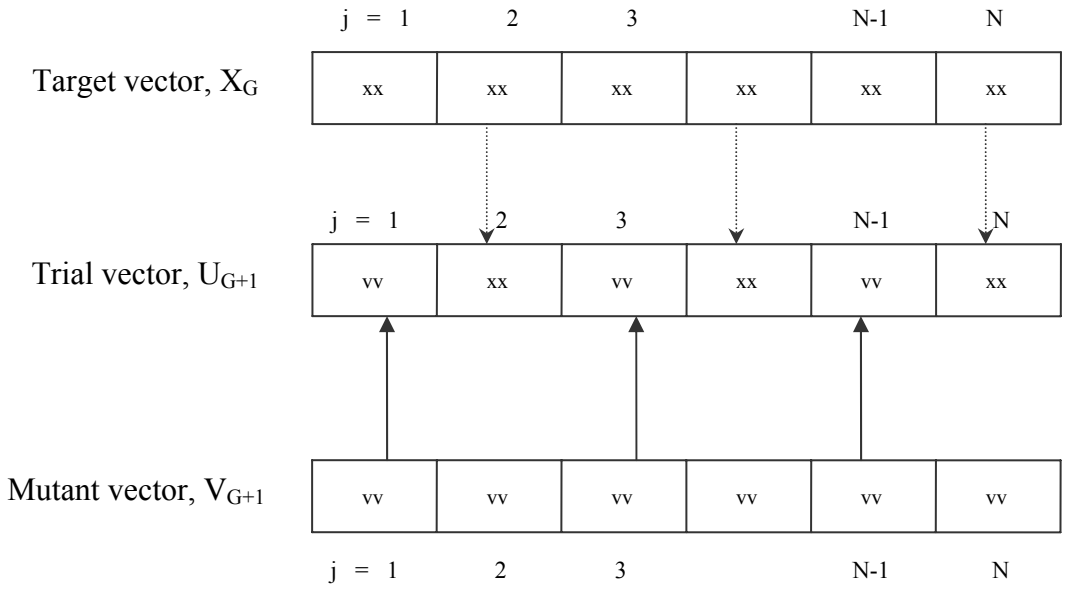


Figure 3.2: Schematic diagram of crossover operation; for continuous lines, ran ≤ CR, and for dotted lines, ran > CR

vector has a better value compared to the target vector, it replaces the target vector in the population thus allowing the best solution into further generations. The process of mutation, crossover and selection is repeated until a termination criterion such as maximum number of generations is satisfied. The algorithm then terminates providing the best point that has been explored over all the generations. The best point is further refined using a fast convergent quasi-Newton method to achieve the best minimum which is declared to be the global minimum.

3.1.2 Tabu Search

TS, first developed by Glover (1989, 1990), has been widely used for combinatorial optimization (Youssef et al., 2001) but its use is very limited in continuous optimization (Hu, 1992; Chelouah and Siarry, 2000; Teh and Rangaiah, 2003; Lin and Miller, 2004a and b). TS is a meta heuristic that guides the heuristics to escape from the local minima. The main concepts of TS include diversification and

identifying the most promising region. The diversification step performs an exhaustive search in the entire solution space by generating solutions that are not seen before. To implement this, TS maintains both tabu list (consisting of unpromising points) and promising list to avoid repeated visits to the same place in the search region, which in turn improves the computational efficiency. After a specified maximum number of iterations, in-depth search known as intensification is performed from the most promising point.

The TS algorithm (Figure 3.3) starts with the selection of values for the parameters: tabu list size (N_t), promising list size (N_p), tabu and promising radii (ε_t and ε_p), length of the hyper-rectangle (h_n), initial population size (NP_{init}), number of neighbors (N_{neigh}), maximum number of successive iterations (Sc_{max}) without improvement in the best function value and maximum number of iterations ($Iter_{max}$). The algorithm then randomly generates a population of specified size and evaluates the objective function value at each individual. The best point is filled into the promising list and the remaining will be sent to the tabu list. The best point found is selected as the current centroid (s) of the hyper-rectangle, which is used to generate neighbors to explore for better points in the neighborhood. The generation of neighbors can be executed in many ways, i.e., either by using hyper-circles or hyper-rectangles etc. In this study, hyper-rectangles have been used to generate the neighbors. A detailed explanation about the generation of neighbors using hyper-rectangles is available in Teh and Rangaiah (2003). The neighbors are then compared with the points in tabu and promising lists, and only those points away from the latter are evaluated. The rejection of the neighbors which are nearer to the points in tabu and promising lists improves the computational efficiency of TS avoiding repeated

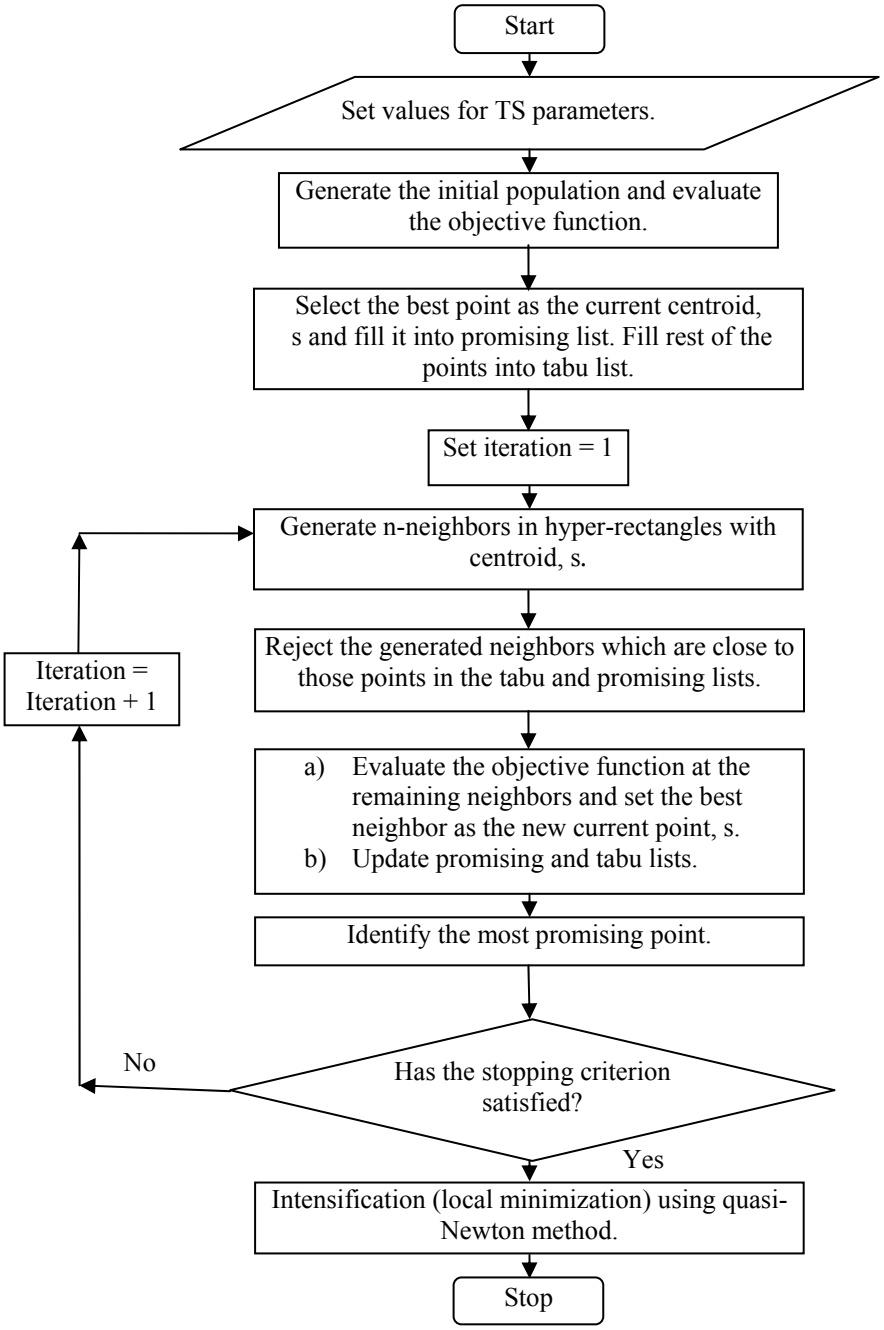


Figure 3.3: Flow chart of TS-QN

visits to the same place during the search. The algorithm selects the best point found in the current iteration as the centroid of the hyper-rectangle to generate neighbors for the next iteration. The best point in the current iteration is accepted even if it is worse than that of the previous iterations to avoid entrapment in the local minima. The process of generating neighbors is repeated and the tabu and promising lists are updated in each iteration. Once the tabu/promising list is filled, the next tabu/promising point will be placed in the first position of tabu/promising list and subsequent positions are occupied by the remaining points. Thus both tabu and promising lists are updated dynamically during the search to keep the latest point(s) in the list by replacing the earliest-entered point(s). After a specified number of iterations, most promising area is identified and is further investigated by intensification step. Generally, a local optimization technique is used in this step; a fast convergent quasi-Newton technique is used in this study. The algorithm then terminates by declaring the final solution as the global minimum.

3.2 Implementation of DE and TS

A Matlab code for DE is taken from the website http://www.icsi.berkeley.edu/~storn/code.html, and a boundary violation check is implemented in the code. For the local minimization step, an in-built subroutine from the Matlab optimization tool box namely, FMINCON is used. The objective function for DE code is written in FORTAN and simple gateway functions are used to call it from the Matlab. This is adopted as all our programs for PEC and PS are in FORTRAN. For TS, the FORTRAN code developed by Teh and Rangaiah (2003) is used; it uses the IMSL subroutine namely, DBCONF for the local minimization step.

Both FMINCON and DBCONF employ the fast convergent quasi-Newton method with BFGS update for the Hessian matrix. For the first time, a local optimization technique is used with DE in this study, and a similar work is done for TS by Teh and Rangaiah (2003). The minimization technique at the end of these methods is executed to find the final solution accurately and efficiently.

3.3 Benchmark Problems

Several benchmark problems having 2 to 20 variables and a few to several hundreds of local minima are used to evaluate both DE followed by quasi-Newton method (DE-QN) and TS followed by quasi-Newton method (TS-QN). A brief description of the functions and the global minima are given in Table 3.1.

Table 3.1: Details of the benchmark problems

Function	Number of variables (N)	Global minimum	Remarks
Goldstein and Price function (GP_2)	2	3 at x = {0, -1}	Four local minima
Easom function (ES_2)	2	-1 at x = $\{\pi, \pi\}$	Several local minima
Shubert function (SH_2)	2	-186.7309 at x = {-0.8427, -0.1889}	18 global minima; 760 local minima
Hartmann function (H_3)	3	-3.86278 at x = {0.114614, 0.555649, 0.852547}	Four local minima
Rosenbrock function (ROS_N)	2,5,10 and 20	0 at x = {1, …, 1}	Several local minima
Zakharov function (ZAK_N)	2,5,10 and 20	0 at x = {0, …, 0}	Several local minima

Two types of stopping criteria are used in this study. They are maximum number of iterations/generations ($Iter_{max}$ in TS-QN and Gen_{max} in DE-QN) (referred

as stopping criterion 1 (SC1)) and maximum number of iterations/generations or maximum number of successive iterations/generations (Sc_{max}) without improvement in the best function value (referred as stopping criterion 2 (SC2)). Several published studies (e.g., Cai and Shao, 2002) employed convergence to the global minimum as a stopping criterion. On the contrary, we used SC1 and SC2 because, in reality, global minimum of application problems is unknown *a priori*. The performance of the two methods is evaluated based on both reliability (measured in terms of how many times the algorithm located the global minimum out of 100 trials, referred as success rate (SR)) and computational efficiency (measured in terms of average number of function evaluations (NFE) in all the 100 trials). The gradient is calculated numerically and the NFE includes the function calls for evaluating both the objective function and the numerical gradient for the quasi-Newton method.

3.3.1 Parameter Tuning

Test functions GP_2, ES_2, SH_2, ROS_5, ROS_{10} and ROS_{20} have been selected to tune the parameters of TS-QN and DE-QN to find the global minimum with good reliability and computational efficiency. The nominal parameter values chosen for TS-QN are N_t and N_p =10; ε_t and ε_p = 0.01; h_n = 0.5; NP_{init} = 20N, where N is the dimension of the problem; N_{neigh} = 2N (subject to a minimum of 10 and a maximum of 30); Sc_{max} = 5N and $Iter_{max}$ = 50N, and for DE-QN, A = 0.4; CR = 0.1; NP = 50; Sc_{max} = 5N and Gen_{max} = 50. The nominal values for TS-QN and DE-QN are chosen based on the optimum values available in Chelouah and Siarry (2000), and preliminary numerical experience with some of the benchmark problems respectively. The tuning is performed by varying one parameter at a time while the rest are fixed at

their nominal/recent optimum values. The optimal parameters obtained for TS-QN and DE-QN are given in Table 3.2. The optimal parameters found for TS-QN are the same as its nominal parameters. This may be because the nominal parameters are chosen based on the optimal settings given in Chelouah and Siarry (2000). The optimal NP obtained for DE-QN is independent of N because the effect of N is accounted via Gen_{max} and the size range (up to 20 variables) of the problems considered in this study.

Table 3.2: Optimal parameter values for TS-QN and DE-QN

Parameters	Benchmark problems	Phase equilibrium calculations	Phase stability problems
TS-QN			
N_t and N_p	10	10	10
ε_t and ε_p	0.01	0.02	0.02
NP_{init}	20N	20N	20N
N_{neigh}*	2N	2N	2N
h_n	0.5	0.5	0.5
$Iter_{max}$	50N	100N	min {50N, 100}
Sc_{max}	6N	2N	2N
DE-QN			
A	0.5	0.3	0.3
CR	0.5	0.9	0.9
NP	50	min {50N, 200}	min {50N, 100}
Gen_{max}	20N	75	50
Sc_{max}	10N	12N	6N

* N_{neigh} is restricted to a minimum of 10 and a maximum of 30 for benchmark and PEC problems, and to 20 to 30 for PS problems to have good reliability and computational efficiency.

3.3.2 Results and Discussion

The results for solving the benchmark problems by TS-QN and DE-QN are given in the Tables 3.3a and 3.3b. Each benchmark problem is solved 100 times, each time by generating a random initial estimate. The results are compared in terms of SR and NFE, which is the average of all 100 trials. It is evident from Table 3.3a that the reliability of DE-QN is better compared to TS-QN, for both SC1 and SC2. This is perhaps due to the different escaping mechanisms associated with these two methods. DE performs mutation and crossover over a set of individuals (i.e., population), whereas TS accepts the best point in each iteration as the new centroid of the hyper-rectangle for generating neighbors even though it is worse than the previous best points in order to escape from the local minima. The reliability of TS-QN is less for ES_2 function because the function is flat everywhere in the feasible region except near the center (global minimum region). As the function is flat, all the neighbors generated in TS-QN will have the same value trapping the search in that region, whereas DE-QN explored the global minimum region by generating different new individuals by the process of mutation and crossover. The reliability of both TS-QN and DE-QN for Shubert function is high even though it has 760 local minima. This may be because locating one of the several global minima (around 18) in this example is sufficient to achieve the best function value. The reliability of TS-QN is less for Rosenbrock functions because of narrow global minimum region in these functions.

Even though reliability of DE-QN is more than TS-QN, its computational efficiency is less compared to TS-QN (Table 3.3a). NFE for DE-QN is 1.05 (ZAK_{20}) to 3.11 times (H_3) more than that for TS-QN using SC1, and is 1.05 (ZAK_{20}) to 7.95

Table 3.3a: Success rate (SR) and Number of Function Evaluations (NFE) for solving benchmark problems by DE-QN and TS-QN

Function	TS-QN				DE-QN			
	SC1		SC2		SC1		SC2	
	SR	NFE	SR	NFE	SR	NFE	SR	NFE
GP_2	100	918	99	301	100	2,026	100	1,998
ES_2	90	1,040	85	433	100	2,072	76	1,747
SH_2	100	1,033	92	355	99	3,051	100	1,790
ROS_2	100	1,059	100	475	100	2,065	100	2,065
ZAK_2	100	1,009	100	343	100	2,077	100	2,048
H_3	100	987	100	386	100	3,071	100	3,071
ROS_5	78	2,799	79	2,081	100	5,177	100	5,171
ZAK_5	100	2,629	100	1,294	100	5,093	100	5,093
ROS_{10}	78	8,578	78	8,541	100	10,213	100	10,210
ZAK_{10}	100	8,491	100	8,473	100	10,143	100	10,125
ROS_{20}	75	22,074	75	22,074	98	20,394	95	20,338
ZAK_{20}	100	19,157	100	19,157	100	20,238	100	20,234

Table 3.3b: SR and NFE using DE-QN and TS-QN with same N_{neigh}/NP (= 20) and $Iter_{max}/Gen_{max}$ (=50N)

Function	TS-QN		DE-QN	
	SR	NFE	SR	NFE
GP_2	100	1,384	100	2,036
ES_2	83	1,683	100	2,027
SH_2	100	1,621	100	2,056
ROS_2	100	1,603	100	2,124
ZAK_2	100	1,542	100	2,027
H_3	100	2,021	100	3,039
ROS_5	76	4,002	100	5,021
ZAK_5	100	4,142	100	5,061
ROS_{10}	87	8,395	99	10,185
ZAK_{10}	100	8,268	100	10,092
ROS_{20}	81	18,499	96	20,308
ZAK_{20}	100	15,603	100	20,020

times (H_3) more than that of TS-QN using SC2. This could be because of avoiding repeated visits to the same place in TS by keeping track (i.e., by maintaining tabu and promising lists) of the previous search points which in turn improves its computational efficiency. NFE for both TS-QN and DE-QN increases with the number of variables due the increase in the size of the solution space which makes both the algorithms to generate more points.

DE-QN and TS-QN have also been evaluated using SC2. The computational efficiency and reliability (Table 3.3a) are better and comparable using SC2 compared to that of SC1 for both DE-QN and TS-QN. NFE of DE-QN using SC1 is 1.01 (ZAK_2) to 1.7 times (H_3) more compared to SC2; similarly, NFE of TS-QN is 1.34 (ROS_5) to 3.05 times (GP_2) more compared to SC2 for TS-QN. This is because the algorithms will terminate if the best function value does not change successively after the specified Sc_{max} iterations/generations resulting in good computational efficiency with SC2.

The performance of TS-QN and DE-QN is also compared by keeping the similar parameters (i.e., N_{neigh} and $Iter_{maxt}$ in TS-QN; NP and Gen_{max} in DE-QN) to the same value. The values chosen are: N_{neigh} and NP = 20; and $Iter_{max}$ and Gen_{max} = 50N, and the remaining (algorithm-specific) parameters are chosen from Table 3.2. The algorithms are compared using SC1 so that another parameter Sc_{max} need not be included. From the results given in Table 3.3b, it is clear that the reliability (i.e., SR) of DE-QN is better compared to that of TS-QN, whereas the computational efficiency (NFE) of the latter is better than that of DE-QN for all the functions tested. Thus, the relative performance of TS-QN and DE-QN with similar parameter values is the same

as that with optimal parameters. Tables 3a and 3b show that SR and NFE for TS-QN with similar parameter values are inferior to those with optimal parameters for smaller problems. On the other hand, SR and NFE for DE-QN with similar parameter values (Table 3.3b) are comparable to those with optimal parameters (Table 3.3a). These findings highlight the challenges in tuning parameters and the possibility of different sets of optimal parameters. In this study, parameters are tuned carefully and systematically to ensure a fair comparison of the methods.

3.4 Phase Equilibrium Problems

The details of mathematical formulation for these problems are discussed under section 2.5 (Chapter 2). Phase equilibrium examples considered in this study include vapor-liquid equilibrium (VLE), liquid-liquid equilibrium (LLE) and vapor-liquid-liquid equilibrium (VLLE) examples involving multiple components (2 to 8 components) and popular thermodynamic models. The feed composition, operating conditions and the thermodynamic models used for each example, and global and local minima for all these examples are given in Teh and Rangaiah (2003). In this Chapter (and in subsequent Chapters also), only the difficult examples (i.e., examples with multiple minima) are chosen from Teh and Rangaiah (2003) for evaluation of both DE and TS since these examples pose real challenge to the global optimization algorithms. Along with local minima, there are trivial solutions for several examples at which equilibrium composition equals to the feed composition (Teh and Rangaiah, 2002).

3.4.1 Parameter Tuning

Examples 5, 9 and 10 are selected for tuning the parameters of TS-QN and DE-QN, which are shown to be difficult in Teh and Rangaiah (2002). The parameters are tuned one at a time keeping others fixed at their nominal/recent optimum values. The optimal parameters obtained for TS-QN and DE-QN are given in Table 3.1. The difference in the values of the parameters (A, CR and Gen_{max} for DE-QN, and ε_t, ε_p and $Iter_{max}$ for TS) compared to those for benchmark problems is due to the presence of a few but comparable minima (i.e., function values at last local minimum and the global minimum are close to each other) in the PEC examples.

3.4.2 Results and Discussion

All the examples are solved 100 times each, starting from a different point randomly chosen each time in the feasible region. In this and subsequent Chapters, all the application problems are solved 100 times each for consistency and better evaluation, compared to 25 times in the earlier Chapter. Both DE-QN and TS-QN are evaluated based on two types of stopping criteria: SC1 and SC2. The results are shown in Table 3.4a and are given in terms of reliability (i.e., SR) and computational efficiency (i.e., NFE), and are average over all the 100 trials. The study here is limited to the calculation of equilibrium compositions and the number of phases existing at equilibrium is assumed to be known *a priori*.

Table 3.4a shows that DE-QN is able to solve all but one example tested with 100% success rate with both SC1 and SC2. The reliability of TS-QN is comparable to

DE-QN for all the VLE and LLE examples except for example 8. This is due to the presence of comparable minima (function values at the global minimum and at the trivial solution are -0.360353 and -0.354340 respectively) in this example. The reliability of TS-QN is low for example 10 (VLLE) also because of its comparable minima (function value at the local and global minima are -1.235896 and -1.233294 respectively). The SR of TS-QN for this example is lower compared to example 8 because of the very close function values at the local and global minima and more number of variables (6 for example 10 and 3 for 8). As the function value at local minimum is close to that of the global minimum, the algorithm traps into local minimum either because of the premature convergence or due to the difficulty in escaping from the local minimum because of the narrow better regions (i.e., regions where the function value is less than that of the local minimum) resulting low SR.

For example 4, both DE-QN and TS-QN failed to locate the global minimum because of the comparable minima (function value at the local and global minima are -161.5364 and -161.5416) and the increase in dimensionality (10 variables). Overall, reliability of DE-QN is better compared to TS-QN indicating that the escaping mechanism of DE is better than that of TS-QN using SC1. The NFE of DE-QN is 2.69 (example 9) to 7.24 (example 8) times more than that of TS-QN. Even though the reliability of DE-QN is more, its computational efficiency is inferior to TS-QN. This is because of avoiding revisits to the same place during the search in TS-QN. For all examples, NFE increases with number of variables (example 2 with 2 variables to example 10 with 6 variables) for both TS-QN and DE-QN due to the associated large solution space.

Table 3.4a: SR and NFE for solving phase equilibrium problems by DE-QN and TS-QN

Example number*	TS-QN				DE-QN			
	SC1		SC2		SC1		SC2	
	SR	NFE	SR	NFE	SR	NFE	SR	NFE
Vapor-liquid equilibrium problems								
1(2)	99	1,348	99	1,348	100	7,608	88	6,345
2(6a)	96	1,618	96	1,618	100	11,421	100	11,421
3(6b)	96	1,639	96	1,639	100	11,424	100	11,424
4(9)	---	---	---	---	---	---	---	---
Liquid-liquid equilibrium problems								
5(11a)	98	1,432	98	1,432	100	7,600	100	6,788
6(11b)	100	1,359	100	1,359	100	7,625	100	7,076
7(12)	100	1,367	100	1,367	100	7,621	100	7,510
8(13)	94	1,575	94	1,575	100	11,417	99	11,359
Vapor-liquid-liquid equilibrium problems								
9(17)	100	5,648	100	5,648	100	15,215	100	15,215
10(18)	81	5,486	81	5,486	100	15,226	100	15,226

* The number in the brackets refers to the example number in Teh and Rangaiah (2003). This study considers only the more difficult examples from the previous work.

Table 3.4b: Comparison of CPU times for DE-QN and TS-QN with those in the literature

Example number and details	Computation time in seconds			Reference
	TS-QN	DE-QN	Reported (Estimate)*	
1 (VLE, PR)	0.022	0.122	0.643 (0.232)	Hua et al. (1998)
5 (LLE, NRTL)	0.018	0.090	0.234 (0.026)	McDonald and Floudas (1997)
6 (LLE, UNIFAC)	0.024	0.156	0.370 (0.042)	
9 (VLLE, UNIFAC)	0.201	0.201	8.800 (1.001)	

* The number in the brackets is the estimated CPU time, based on MFlops using LINPACK benchmark problem, for the computer (Pentium 4, CPU 2.8 GHz and 512 MB RAM) used in the present study.

As shown in Table 3.4a, the reliability and computational efficiency of DE-QN using SC2 is comparable and slightly better compared to SC1. This is because the algorithm terminates if the best function value does not change even after Sc_{max} number of generations in SC2. For TS-QN, there is no improvement in the computational efficiency using SC2 because the maximum number of iterations ($Iter_{max}$) is reached before the specified number of Sc_{max} iterations, causing the algorithm to terminate. Note that both the methods can be used with different thermodynamic models (such as activity coefficient and equation of state models) for describing the physical equilibrium.

The CPU times of DE-QN and TS-QN are also compared with interval Newton and branch and bound methods for a few examples available in Hua et al. (1998) and McDonald and Floudas (1997) respectively. The computer system used in the present study is Pentium 4 (CPU 2.8 GHz, 512MB RAM). The average (over 10 trials) MFlops (million floating point operations per second) of this computer for the LINPACK benchmark program (available at http://www.netlib.org) for a matrix of order 500 is 211. The computer systems used in Hua et al. (1998) and McDonald and Floudas (1997) are Sun Ultra 1/170 workstation and HP9000/730 respectively, and the corresponding MFlops for the LINPACK benchmark program for a matrix of order 100 are 24 and 76-75 respectively (Teh and Rangaiah, 2003). Therefore, the computer system used in this study is about 2.8 and 8.8 times faster than Sun Ultra 1/170 and HP9000/730 respectively. The order of the matrix in the present LINPACK benchmark program (at http://www.netlib.org) is higher (500), and is used for the computer in this study compared to the earlier (100) used for computers of Hua et al. (1998) and McDonald and Floudas (1997), to cope with current high-speed

computers. Although the difference in the matrix orders may have some effect on MFlops, this is not considered here. The CPU times (in seconds) for DE-QN, TS-QN, interval Newton, and branch and bound methods are compared in Table 3.4b; the numbers in the brackets represent the corresponding CPU time (estimated using the factors based on MFlops) for the computer system used in this study. CPU time taken by TS-QN is less compared to all other methods because of its lesser NFE; CPU time of DE-QN is less than that of interval Newton method for example 1. Though CPU time of branch and bound method is less than that of DE-QN for examples 5 and 6, it is higher for example 9, probably due to the number of variables in them (number of variables for examples 5, 6 and 9 is 2, 2 and 6 respectively). The number of nodes in branch and bound method increases exponentially with number of variables, resulting in more computational time.

3.5 Phase Stability Problems

For a given temperature (T) , pressure (P) and composition $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_{nc})$, the molar Gibbs free energy, g of the system is given as the summation of the product of mole fraction and partial molar Gibbs free energy, $\overline{G_i}$ for all components (Rangaiah, 2001):

$$g = \sum_{i=1}^{nc} x_i \overline{G}_i \tag{3.2}$$

The tangent plane, t at a specified composition $\mathbf{x}^* = \{x_1^*, x_2^*, x_3^*, \ldots, x_{nc}^*\}$ is given as:

$$t = \sum_{i=1}^{nc} x_i \overline{G}_i^* \tag{3.3}$$

where superscript * represents evaluation at composition x*. The TPDF can be expressed as:

$$H = g - t = \sum_{i=1}^{nc} x_i (\overline{G}_i - \overline{G}_i^*) \tag{3.4}$$

Depending on the expressions of $\overline{G}_i$ and $\overline{G}_i^*$, different forms of H exist. If the non-ideality of the phase is described by fugacity approach, then the dimensionless H can be expressed as:

$$H = \sum_{i=1}^{nc} x_i \left[\ln(\phi_i x_i) - \ln(\phi_i^* x_i^*) \right] \tag{3.5}$$

where ϕ_i represents the fugacity coefficient of the component i in the given phase. If the excess Gibbs free energy approach is used to represent the non-ideality, then the dimensionless H can be expressed as:

$$H = \sum_{i=1}^{nc} x_i \left[\ln(x_i) - \ln(x_i^* \gamma_{iL}^*) \right] \tag{3.6}$$

where γ_{iL} represents the activity coefficient of component i in the liquid phase L. Depending upon the approach, the objective function is either equation 3.5 or 3.6 and the constraints are:

$$\sum_{i=1}^{nc} x_i = 1 \tag{3.7}$$

and $0 \le x_i \le 1$ (3.8)

The decision variables in PS problems are x_i for i = 1, 2, …, nc. The constrained problem can be transformed into an unconstrained problem by introducing the variables, β_i (for i = 1, 2, …, nc-1) instead of mole fractions x_i (for i = 1, 2,…, nc) similar to equations 7 to 9. To avoid the computational difficulties, the lower bounds are taken as 10^{-15} instead of 0. The examples considered (Rangaiah, 2001) include multiple components and different thermodynamic models. Several compositions are considered for each example. The number of components, feed

composition, temperature and pressure of all these examples can be found in Rangaiah (2001) except for example 5 (Table 3.5) taken from Castillo and Grossman (1981). The global minimum, comparable minimum (i.e., the local minimum with function value very close to that of the global minimum) and the composition at the comparable minimum for all these cases are given in Table 3.6. The composition at the global minimum for the first four examples can be found in Rangaiah (2001) while that for 5th example in Table 3.5.

Table 3.5: Details of the phase stability example 5 – toluene (1), water (2) and aniline (3) at 298 K and 1.0 atm (Castillo and Grossman, 1981)

Composition	Component	Feed composition, x_i^*		Global solution	
		Liquid 1	Liquid 2	Objective function value	x
1	1	0.29989	-	-0.29454012	0.000067
	2	0.20006	-		0.996865
	3	0.50005	-		0.003068
2	1	0.34673	0.00009	0.0	0.346740
	2	0.07584	0.99495		0.075840
	3	0.57742	0.00496		0.577420

3.5.1 Parameter Tuning

Compositions 2, 4 and 5 of Example 2, which are found to be difficult in the preliminary trials, are chosen to tune the parameters of TS-QN and DE-QN. For PS examples, a random way of generating neighbors from the centroid in TS-QN is also studied along with the systematic way (using hyper-rectangles) of generating neighbors. This is because, for some examples, generation of neighbors using hyper-rectangles did not give good reliability, perhaps due to the distribution of local and

global minima in these examples. As shown in Figure 3.4, the problem (2^{nd} example, 5^{th} composition) has a local minimum at β = (0.692780, 0.0129963) and the global minimum at β = (0.278990, 0.682251) with function values 1.323587×10^{-6} and 0 respectively. Initially the TS-QN using hyper-rectangles to generate neighbors found a best point ('+' point in Figure 3.4) in the local minimum region and set it as a new centroid of hyper-rectangles to generate neighbors. As the local minimum is close to the boundary, the distribution of neighbors ('o' points in Figure 3.4) is not spread to the global minimum region (i.e., one side of the hyper-rectangle becomes the lower boundary of the variables forcing many points near the boundary). To circumvent this difficulty, a random way of generating neighbors ('*' points in Figure 3.4) is implemented to explore better points in the global minimum region for these problems. A mixed (generating half of the total number of neighbors using hyper-rectangles and the rest randomly) way of generating neighbors is also studied.

For PS problems, the optimal values of parameters (Table 3.2) obtained for TS-QN are the same as those for PEC except $Iter_{max}$ = 50N. The optimal parameters of DE obtained are given in Table 3.2. The parameters $Iter_{max}$ and Gen_{max} are less compared to PEC because the number of decision variables is less compared to PEC for these problems.

Table 3.6: Global and comparable minima for PS problems

Composition	Function value	
	Global minimum	**Comparable minimum**
Example 1 (1)		
1	-0.03246624	0 at {0.5; 0.5}
2	-0.21418620	No local minimum.
3	-0.07427426	-6.06283×10^{-3} at {0.39221; 0.60778}
4	-0.00671171	0 at {0.65; 0.35}
5	-0.00070557	0 at {0.93514; 0.06486}
6	0	1.11127×10^{-3} at {0.93476; 6.5230×10^{-2}}
Example 2 (2)		
1	-0.11395074	0 at {0.4; 0.3; 0.3}
2	-0.05876117	-4.11636×10^{-6} at {0.61986; 0.00562; 0.37452}
3	-0.22827470	-2.71678×10^{-3} at {0.20145; 0.43018; 0.368351}
4	-0.02700214	-3.09637×10^{-6} at {0.03001; 0.00211; 0.96788}
5	0.0	1.323587×10^{-6} at {0.69280; 0.00399; 0.30321}
Example 3 (3)		
1	-0.00395983	1.08905×10^{-2} at {0.11520; 0.88479}
2	-0.08252179	-5.68944×10^{-2} at {0.11813; 0.88186}
3	-0.00246629	0 at {0.112; 0.888}
Example 4 (4)		
1	-1.48621570	-1.48554 at {0.94672; 4.35930e-2; 7.85484×10^{-3}; 1×10^{-15}; 1.2478×10^{-3} ; 1.96839×10^{-4}; 2.63986×10^{-4}; 1.20802×10^{-4}}
Example 5		
1	-0.29454012	0 at {0.29989; 0.20006; 0.50005}
2	0	No local minimum.

Note: The number in the brackets refers to the example number in Rangaiah (2001).

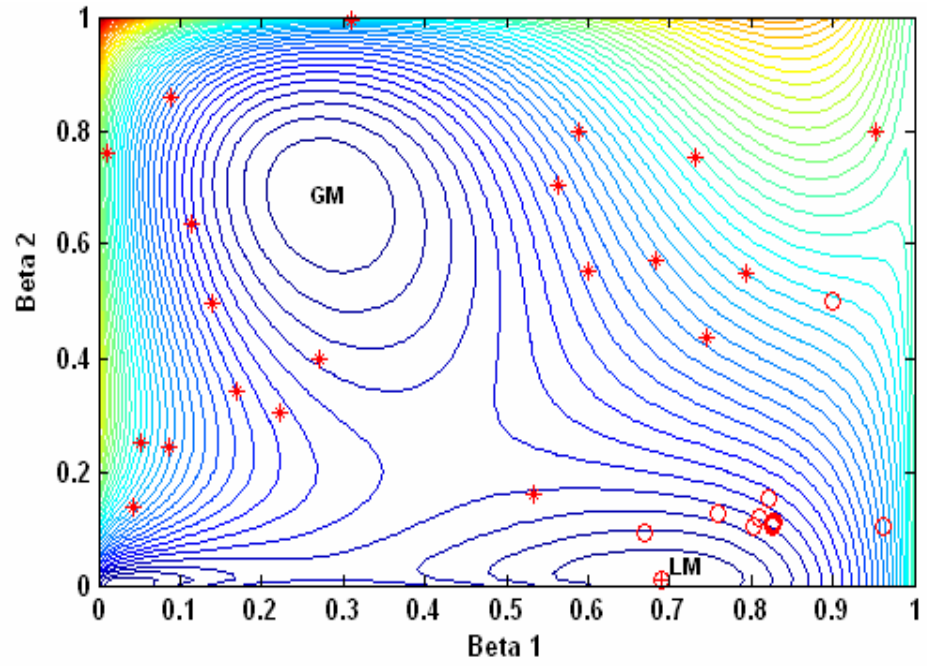


Figure 3.4: Generation of neighbors for example 2 (composition 5) using hyper-rectangles (points with 'o') and randomly (points with '*') from the best point (point with '+') at β = {0.69280; 0.01298}

3.5.2 Results and Discussion

All the PS examples are solved 100 times, each time from a different randomly chosen point in the feasible region. Initially, the examples are solved with SC1 to study the performance of TS-QN with different ways of generating neighbors and the results (averaged over all the 100 trials) of TS-QN and DE-QN are given in Table 3.7. The results for TS-QN are given for three types of generating neighbors: TS-S-QN (systematic using hyper-rectangles), TS-R-QN (randomly) and TS-M-QN (mixed).

Table 3.7: SR and NFE for solving phase stability problems by DE-QN, TS-S-QN, TS-M-QN and TS-R-QN using SC1

Composition	NFE for TS-S-QN	NFE for TS-M-QN	NFE for TS-R-QN	NFE for DE-QN
Example 1				
1	567 (99%)	360	684	2,568
2	563	330	645	2,562
3	465	361	681	2,566
4	571	332	642	2,569
5	583	324	640	2,557
6	581	325	639	2,567
Example 2				
1	1196	1890	1,989	5,112
2	1,273 (64%)	1768	2,023	5,115
3	1272	1956	2,035	5,116
4	1,240 (91%)	1810	1,987	5,111
5	1,277 (84%)	1,878 (85%)	1,984 (76%)	5,114
Example 3				
1	569	645	757	2,584
2	577	523	759	2,582
3	558	646	758	2,582
Example 4				
1	2,100	2203	2,767	5,143
Example 5				
1	1,306	1095	2,048	5,120
2	1,280	1026	1,981	5,108

Note: SR is 100% for all cases except for those given in the brackets.

The results in Table 3.7 show that both DE-QN and TS-QN have high reliability in locating the global minimum. The reliability of DE-QN is comparable to TS-M-QN and TS-R-QN, and is better than TS-S-QN. This shows that the systematic way of generating neighbors has less reliability for these problems. The SR of TS-M-QN, TS-R-QN and DE-QN are 100% for all examples except for 2nd example with 5th composition for which TS-QN with all types of generating neighbors is low compared to DE-QN. This is because of the presence of comparable minima (function value at the local and global minima are 1.32358×10^{-6} and 0 respectively) in this example. Consequently, the better region becomes narrower and narrower causing failure of

TS-QN to locate the global minimum region, whereas DE-QN is able to explore the global minimum region with its escaping mechanism (mutation and crossover).

The computational efficiency of TS-M-QN and TS-S-QN is better compared to TS-R-QN. This may be because some neighbors in the mixed type of generation may be close enough such that they are near to the points in the tabu list which in turn avoids repeated evaluations. Even though DE-QN is more reliable than TS-QN, the latter is computationally more efficient than the former. The NFE of DE-QN is around 2.3 times (4^{th} example) to 7.1 times (1st example, 1^{st} composition) more than TS-M-QN. This is due to avoiding repeated visits to the same place by keeping track of the previous points during the search.

The examples are also solved with SC2 for TS-M-QN (found to be the best among all TS-QN tried) and DE-QN. The results (Table 3.8) show that there is no improvement in the computational efficiency of TS-M-QN using SC2 compared to that of SC1, because the maximum number of iterations is reached before the specified Sc_{max} number of iterations. This also indicates that the parameter $Iter_{max}$ is fine tuned. The results (Table 3.8) show that the computational efficiency of DE-QN using SC2 is better than that of SC1, and is due to the termination of the algorithm once the specified Sc_{max} number of iterations is reached irrespective of maximum number of iterations. NFE of DE-QN with SC1 is 1.1 (2^{nd} example, 2^{nd} composition) to 4.0 times (1^{st} example, 6^{th} composition) more compared to that of SC2. For example 4, the NFE of DE-QN is the same with both SC1 and SC2. This is because

Table 3.8: Comparison of SR and NFE for solving phase stability problems by DE-QN and TS-M-QN using SC1 and SC2

Composition	TS-M-QN		DE-QN	
	SC1	SC2	SC1	SC2
Example 1				
1	360	360	2,568	755
2	330	330	2,562	779
3	361	361	2,566	869
4	332	332	2,569	727
5	324	324	2,557	647
6	325	325	2,567	643
Example 2				
1	1,890	1,890	5,112	3,808(99%)
2	1,768	1,768	5,115	4,624(96%)
3	1,956	1,956	5,116	4824
4	1,810	1,810	5,111	4024
5	1,878 (85%)	1,878 (85%)	5,114	3,642(99%)
Example 3				
1	645	645	2,584	716
2	523	523	2,582	764
3	646	646	2,582	698
Example 4				
1	2,203	2,203	5,143	5,143
Example 5				
1	982	982	5,120	5,018
2	910	910	5,108	3,566

Note: SR is 100% for all the cases except for those given in the brackets.

here the optimum Sc_{max} value (i.e., $6N \cong 48$) is high because of more number of variables and the number of iterations reaches its maximum number (i.e., $Iter_{max} = 50$ for these problems) earlier than Sc_{max}, terminating the algorithm. For example 5, the NFE of DE-QN for composition 1 is higher than that of composition 2 even though optimum Sc_{max} value is the same for both of them. This may be because the 1st composition has a comparable minimum, and whereas the 2nd composition does not have any local minimum.

The effect of generation of neighbors by random and mixed way is also studied for the benchmark problems and PEC using SC1. The results averaged over

100 trials (Table 3.9a and 3.9b) show that the performance of generation of neighbors using hyper-rectangles is better compared to mixed and random way of generations even though the mixed way is computationally more efficient than the others. They also show that the reliability of TS-M-QN for Rosenbrock functions is slightly better compared to TS-S-QN. These are due to the distribution of minima, which depends on the problem type, and hence generalizing for all types of functions is difficult. The mixed way of generating neighbors seems to be suitable for problems such as PS problems where the local minimum is near the boundary and also far away from the global minimum.

Table 3.9a: Comparison of performance of TS-QN with different types of neighbor generation for benchmark problems

Function	TS-S-QN#		TS-M-QN#		TS-R-QN#	
	SR	NFE	SR	NFE	SR	NFE
GP_2	100	918	100	571	100	1,060
ES_2	90	1,040	15	580	33	1,065
SH_2	100	1,033	95	569	99	1,059
ROS_2	100	1,059	100	611	100	1,096
ZAK_2	100	1,009	100	561	100	1,050
H_3	100	987	100	548	100	994
ROS_5	78	2,799	86	1,656	84	2,901
ZAK_5	100	2,629	100	1409	100	2,649
ROS_{10}	78	8,578	80	6,326	82	11,218
ZAK_{10}	100	8,491	100	5,323	100	10,298
ROS_{20}	75	22,074	83	36,141	78	36,259
ZAK_{20}	100	19,157	100	30,668	100	30,662

TS-S-QN, TS-M-QN and TS-R-QN represent respectively systematic (using hyper-rectangles), mixed and random way of generation of neighbors.

Table 3.9b: Comparison of performance of TS-QN with different types of neighbors generation for phase equilibrium calculations

Example number	TS-S-QN#		TS-M-QN#		TS-R-QN#	
	SR	NFE	SR	NFE	SR	NFE
Vapor-liquid equilibrium						
1	99	1,348	94	1,018	100	1,947
2	96	1,618	95	1,545	95	2,029
3	96	1,639	92	1,094	90	1,985
4	---	---	---	---	---	---
Liquid-liquid equilibrium						
5	98	1,432	97	1,051	99	2,002
6	100	1,359	100	1,072	100	2,023
7	100	1,367	100	1,091	100	2,039
8	94	1,574	87	1,058	99	1,943
Vapor-liquid-liquid equilibrium						
9	100	5,648	100	5,097	100	7,639
10	81	5,486	9	3,985	24	7,601

TS-S-QN, TS-M-QN and TS-R-QN represent respectively systematic (using hyper-rectangles), mixed and random way of generation of neighbors.

3.6 Summary

Two most promising methods namely, DE and TS have been implemented along with a local minimization method (QN) at the end to refine the solution, and evaluated for benchmark, PEC and PS problems. Initially both DE-QN and TS-QN are tested on benchmark problems comprising of 2 to 20 variables and a few to hundreds of local minima. The algorithms are then evaluated and compared for PEC involving multiple components and phases with popular thermodynamic models. Both the methods successfully located the global minima with DE-QN being more reliable compared to TS-QN and the latter being computationally more efficient than the former for both benchmark and PEC problems. For example 4 of PEC, both DE-QN and TS-QN failed to locate the global minimum. The methods are then tested for PS problems involving multiple components. The generation of neighbors in TS is implemented in three different ways: using hyper-rectangles, mixed and random way of generation, to study their effectiveness. The reliability of DE-QN and TS-QN for PS problems is high and the computational efficiency of the latter is better than the former for these problems. The mixed way of generating neighbors is also studied for benchmark problems and PEC, and the results show that systematic way of generating neighbors is suitable for these problems. In summary, results of this study show that the escaping mechanism (via mutation and crossover) in DE-QN is more effective than that of TS-QN, and that TS-QN is computationally more efficient than DE-QN, perhaps due to avoiding revisits to the same place during the search.

CHAPTER 4

DIFFERENTIAL EVOLUTION WITH TABU LIST FOR GLOBAL OPTIMIZATION*

Stochastic global optimization and their applications are attracting greater attention and interest in the recent past as they provide better solutions with relatively less computational effort. Among the many popular methods, differential evolution (DE), proposed by Storn and Price (1997), is a population-based direct search algorithm for non-linear and non-differentiable functions, and has found numerous applications due its simplicity, easy to use and faster convergence. In this work, we attempted to improve the computational efficiency of DE by implementing the concept (i.e., avoiding re-visits during the search) of tabu search (TS) using the tabu list in the generation step of DE; it also provides diversity among the members of the population. DE with tabu list (DETL) is initially tested on several benchmark problems involving a few to thousands of local minima and 2 to 20 variables. It is then tested on challenging phase equilibrium calculations followed by parameter estimation problems in dynamic systems known to have multiple minima. The results show that the performance of DETL is better compared to DE and TS for benchmark, phase equilibrium and parameter estimation problems.

* This chapter is based on the paper – Mekapati Srinivas and Rangaiah, G. P. Differential evolution with tabu list for global optimization and its application to phase equilibrium and parameter estimation problems, Industrial and Engineering Chemistry Research, 46, pp. 3410-3421, 2007b.

4.1 Introduction

Global optimization is an active research area due to its ability to furnish the best possible solutions for highly non-linear and non-convex objective functions. In general, global optimization techniques can be classified into two categories: deterministic and stochastic methods (Pardalos et al., 2000). The former methods provide guaranteed global optimum using certain properties of the objective function such as the continuity, whereas the latter do not require such properties of the objective function and yet can provide global solutions. Stochastic methods are usually quite simple to implement and use, and they do not require transformation of the original problem. Furthermore, these techniques can locate the vicinity of the global solutions with relative efficiency compared to deterministic techniques (Moles et al., 2003). Among the many stochastic methods reported in the literature, genetic algorithm (GA) and differential evolution (DE) are a few of the popular ones. GA was initially developed by Holland (1975); since then, it has been used for many applications in diversified fields. DE, which is relatively new compared to GA, was first proposed by Storn and Price (1997). The working principle of DE is the same as that of GA but it is simple, easy to use and its the convergence rate is shown to be faster compared to GA (Karaboga and Cetinkaya, 2004).

Since 1997, several modifications have been proposed to DE to improve its performance further. Lee et al. (1999) proposed two modifications to the original DE, and used them to solve the dynamic optimization problems. The first modification includes the use of local search in deciding the optimum mutation parameter and the second one systematically reduces the search space by introducing heuristic

constraints. The results show that the convergence of the modified DE is better compared to the original one. Chiou and Wang (1999) proposed hybrid differential evolution (HDE) for static and dynamic optimization problems. HDE consists of two additional steps compared to original DE, namely, acceleration and migration phases to improve the convergence rate without decreasing the diversity among the individuals. Hendtlass (2001) proposed swarm differential evolution, which combines both particle swarm optimization and differential evolution algorithms. The performance of swarm DE was shown to be better compared to both DE and PSO alone for benchmark test functions.

Chiou et al. (2004) used the concept of ant colony search for the proper mutation parameter in HDE, and applied it to solve large capacitor placement problems. The results show that the proposed method performs better than HDE and Simulated Annealing (SA). Liu and Lampinen (2005) proposed fuzzy adaptive differential evolution, which uses fuzzy logic controllers to adapt the search parameters for the mutation and crossover operations. The results of the proposed algorithm for standard test functions show that it is better than the original DE. Bergey and Ragsdale (2005) proposed a modification to DE such that it utilizes selection pressure to develop off-springs that are more fit to survive than those generated from purely random operators. The proposed modification improves the computational efficiency of the original DE. Teo (2006) proposed dynamic self adaptive populations in DE. Experimental results with De-Jong's functions showed that the proposed algorithm with self-adaptive populations produced highly competitive results compared to the conventional DE. Bhat et al. (2006) proposed an improved DE (IDE) for parameter estimation in bio-filter modeling; it uses simplex-

based method to search for better solutions once the globally optimal region is identified. Their results show that IDE is twice as fast as the original DE. Babu and Angira (2006) proposed modified DE (MDE) for optimization of non-linear chemical process problems. MDE maintains only one population compared to two in the original DE. In MDE, the population is updated as and when a better member is generated instead of waiting until generating all members for the new population. The results show that the convergence rate of MDE is faster compared to the original DE.

In this work DE is modified with the concept of Tabu Search (TS) (i.e., avoiding revisits to the same place during the search) to improve the computational efficiency; it also improves the diversity among members of the population and eventually contributes to the computational efficiency. Re-visits during the search in DE are avoided by using tabu list (which keeps track of previous search points that are already evaluated) and hence the proposed method is named as Differential Evolution with Tabu List (DETL). An attempt has also been made to combine the concept of SA into DETL, however, the performance of DETL is not improved by including SA concept and hence only the special features of DE and TS are included in DETL. Another reason for not including SA concept is to minimize the number of parameters in DETL as the SA concept requires three additional parameters. Initially, the concept of TS is used in the *evaluation* step of DE and is applied to several benchmark problems and application problems such as phase equilibrium calculations (Mekapati and Rangaiah, 2006a; this paper is included in Appendix A for ready reference). Subsequently, to improve the performance further, the concept of TS is implemented in the *generation* step of DE. The resulting DETL is then tested on a wide variety of test functions which involve 2 to 20 variables and a few to thousands

of local minima. The method is then tested on challenging phase equilibrium calculations (Teh and Rangaiah, 2003) followed by parameter estimation problems in dynamic systems (Esposito and Floudas, 2000). The results are compared to those of TS, DE and MDE algorithms.

4.2 Differential Evolution with Tabu List (DETL)

In general, the basic DE consists of three steps: generation, evaluation and selection. The generation step involves producing the off-springs (new individuals) by mutation and crossover operations whereas the evaluation step calculates the fitness (objective function) value of each member of the new population. The selection step allows only those individuals of the current generation that have better fitness value, to proceed to the next generation. The process of generation, evaluation and selection steps are repeated until either the best fitness value (global solution) is found or up to the specified maximum number of generations.

The concept of TS is implemented in the generation step of DE (i.e., after crossover and mutation) to improve the computational efficiency and also to improve the diversity among the individuals. The members of the current generation are produced one at a time as in MDE (Babu and Angira, 2006) to have a better convergence rate and to facilitate the implementation of TS. The concept of TS is implemented in DE by using tabu list (with two additional parameters, namely, tabu radius (tr) and tabu list size (tls) which keeps track of the previous evaluated points such that re-visits during the search are avoided. After generating a new member of the current population, it is compared to the already evaluated points in the tabu list in

terms of the Euclidean distance. If the Euclidean distance is smaller than the specified value (tabu radius), which indicates that the objective function value at the new and at one of the points in tabu list are probably close to each other, the newly generated point is rejected considering that it may not give new information about the objective function except increasing number of function evaluations (NFE). The rejected point is replaced by generating a new individual until the Euclidean distance between the new point and to all points in tabu list is greater than the tabu radius. The process of generating new individuals including checking their closeness to those in the tabu list, is repeated until all members of the population are produced. The number of points in the tabu list is specified by the parameter tabu list size (tls) and how far the new points should be from those in the tabu list is decided by the parameter tabu radius (tr). Thus by implementing TS concept in the generation step of DE, unnecessary function evaluations are avoided while maintaining the diversity in the population.

The motivation and working principle of DETL are explained by considering the modified Himmelblau (mHB) (Deb, 2002) function which has 4 minima (Figure 4.1), and the global minimum is at x = {3, 2} with a function value of zero. The members of the population using the conventional DE (points marked with 'o') and with DETL (points marked with '*') after 2 generations are shown in Figure 4.1. More diversity can be observed among the members of DETL compared to DE which consequently helps DETL to explore the global region earlier compared to DE. NFE (averaged over 100 trials) required by DE to locate the global minimum for this function is 6991, whereas DETL (with tr = 1×10^{-6} and tls 30) took about 41% less NFE (4098, averaged over 100 trials). In both these cases, the stopping criterion used

is 'convergence to the global minimum (i.e., achieving the global minimum with an absolute error of 10^{-6} or less in the function value)'.

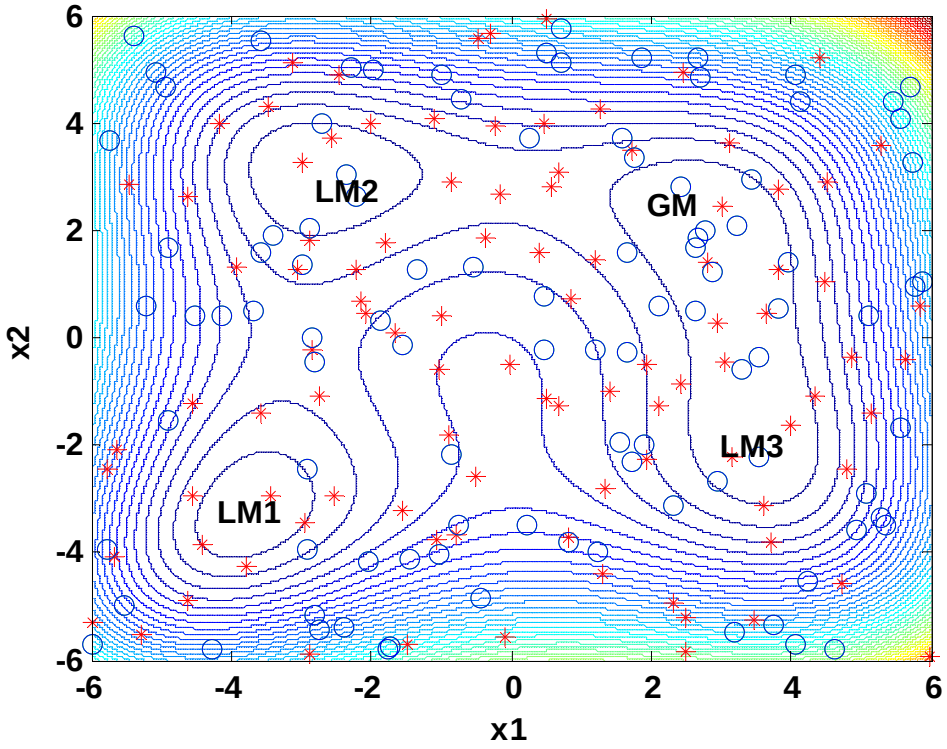


Figure 4.1: Contour diagram of modified Himmelblau (mHB) function; '○' and '✳' denotes the points generated by DE and DETL respectively. GM is the global minimum whereas LM1, LM2 and LM3 are the local minima.

A local optimization technique is included at the end of DE and DETL and it is found that NFE required by DE and DETL for mHB function is 5884 and 3771 (around 16% less for DE and 8% less for DETL compared to the NFE without local optimization) respectively. Hence, we opted to use a local optimization technique at the end of a global optimization throughout this study, to find the final solution accurately and efficiently. In this study, a gradient-based quasi-Newton method is used for local optimization; and the gradient is calculated numerically. Alternately, one can use any direct search method as in Bhat et al. (2006). However, for local

optimization, direct search methods are known to be less efficient than the gradient-based methods.

4.2.1 Description of the method

DETL begins with the selection of values for parameters: population size (NP), amplification factor (A), crossover constant (CR), tls, tr, maximum number of generations (Gen_{max}) and successive maximum number of generations (Sc_{max}) without improvement in the best function value. The algorithm (Figure 4.2) generates the initial population of size NP using uniformly distributed random numbers to cover the entire feasible region. The objective function is evaluated at each individual and the best one is captured. The evaluated individuals/points are then sent to the tabu list, which will be used to ensure that the algorithm does not search again close to these points.

The three main steps: mutation, crossover and selection of DE are performed on the population during each generation/iteration. For this, a mutant individual is generated for each randomly chosen target individual ($X_{i, J}$) in the population by

$$V_{i, J+1} = X_{r1,J} + A\,(X_{r2, J} - X_{r3, J}) \qquad i = 1, 2, ..., NP \tag{4.1}$$

where $X_{r1, J}$, $X_{r2, J}$ and $X_{r3, J}$ are the three random individuals chosen in the population of the current generation J, to produce the mutant individual ($V_{i, J+1}$) for the next generation (i.e. for J+1). The random numbers r_1, r_2 and r_3 should be different from the running index, i and hence NP should be 4 or more for mutation. The mutation parameter or amplification factor, A has a value between 0 and 2, and controls the amplification of the differential variation between the two random individuals. In the

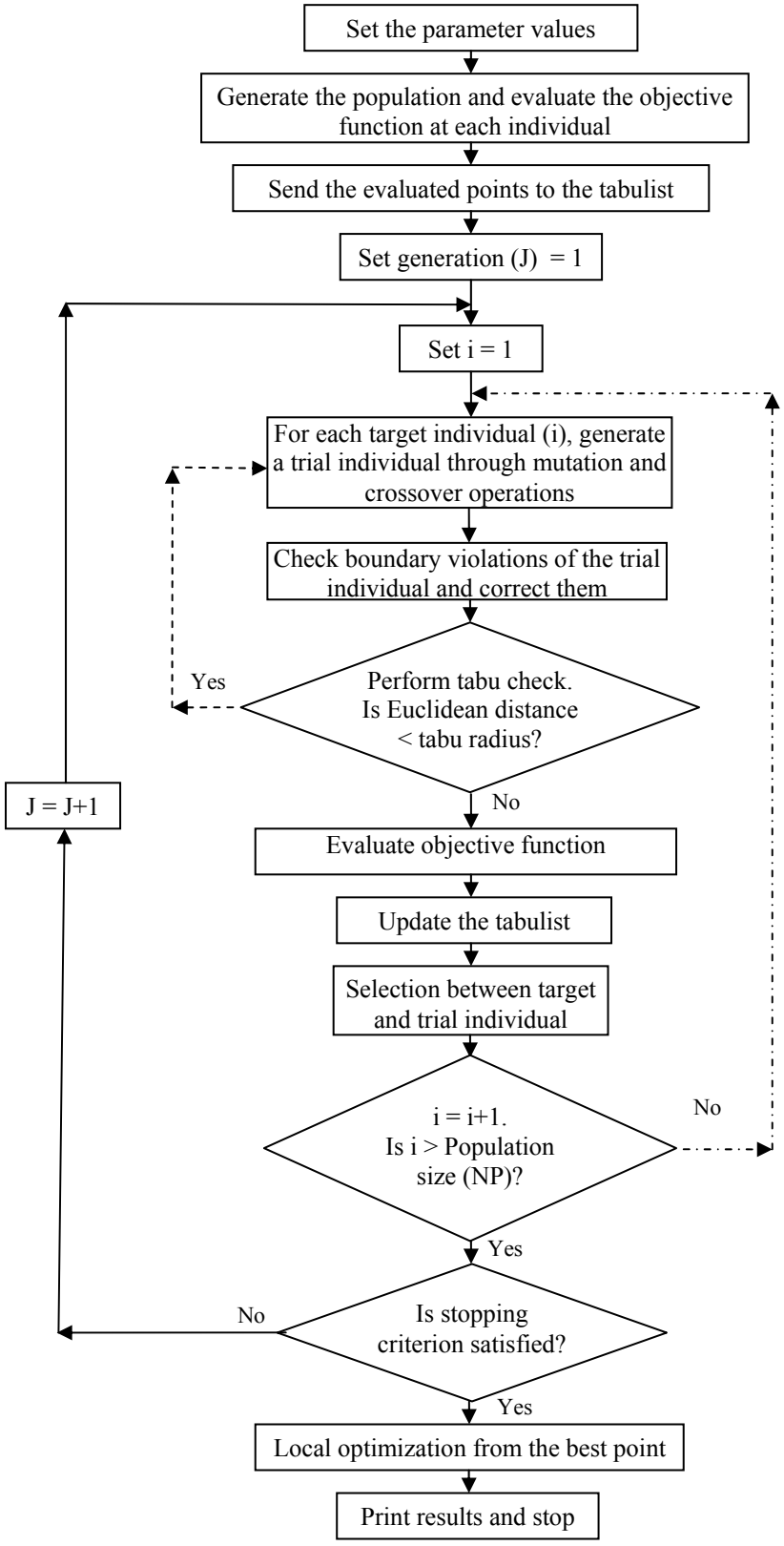


Figure 4.2: Flowchart of DETL

crossover step, a trial individual is generated by copying some elements of the mutant individual to the target individual with a probability of CR. A boundary violation check is performed to check the feasibility of the trial individual; if any bound is violated, the trial individual is either replaced by generating a new individual or forced to the nearest boundary (lower or upper). The trial individual is then compared with the points in the tabu list. If it is near to any of the points in the tabu list, the trial individual is rejected and another point is generated through mutation and crossover operation.

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 point(s) in the list by replacing the earliest entered point(s). In the selection step, a greedy criterion such as fitness (i.e., objective function) value is used to select the better one of the trial and target individuals. If the trial individual is selected, it replaces the target individual in the population immediately and may participate in the subsequent mutation and crossover operations. If the target individual is better, then it remains in the population and may participate in the subsequent mutation and crossover operations.

The process of generation, evaluation and selection is repeated NP times in each generation. The algorithm runs until the stopping criterion such as maximum number of generations (Gen_{max}) or maximum number of successive generations (Sc_{max}) without improvement in the best function value, is satisfied. The best point thus obtained over all the generations is further refined using a local optimization technique, and is declared as the global optimum.

4.3 Benchmark Problems

The performance of DETL is tested initially by employing it to solve several benchmark problems (Storn and Price, 1997; Deb, 2002; and Teh and Rangaiah, 2003) involving a few to thousands of local minima and 2 to 20 variables. These test functions are grouped into two types: moderate and difficult functions. Functions having a few to several local minima are grouped under 'moderate' whereas those with hundreds to thousands of local minima are treated as 'difficult'. This is because the chance of trapping into a local minimum is high for a function having huge number of local minima compared to the one with fewer minima. The exceptions are Shubert (SH) and modified Himmelblau (mHB) functions. SH is grouped under moderate functions even though it has hundreds of local minima due to the numerous global minima (18) it has, whereas mHB is grouped under difficult functions even though it has 4 local minima because it has small global minimum region compared to local minimum regions. Features of the moderate and difficult functions and their global minima are summarized in Table 4.1. Variables in all the test functions (and application problems described later) are normalized between 0 and 1, inside the program.

Table 4.1: Benchmark Problems

Function	Dimension and Domain	Function	Global Minimum	Remarks
Goldstein and Price (GP)[#]	2; $-2 \le x_1, x_2 \le 2$	$[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$ $\times[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2$	3 at x = {0, -1}	Four local minima
Easom (ES)[#]	2; $-100 \le x_1, x_2 \le 100$	$-\cos(x_1)\cos(x_2)\exp\left[-((x_1-\pi)^2+(x_2-\pi)^2)\right]$	-1 at x = $\{\pi,\pi\}$	Flat objective function
Shubert (SH)[#]	2; $-10 \le x_1, x_2 \le 10$	$\left\{\sum_{j=1}^{5} j\cos[(j+1)x_1+j]\right\}\left\{\sum_{j=1}^{5} j\cos[(j+1)x_2+j]\right\}$	-186.7309 at x = {0.0217, -0.9527}	760 local minima
Hartmann 3 variables (H3)[#]	3; $0 \le x_1, x_2, x_3 \le 1$	$-\sum_{j=1}^{4} c_j \exp\left[-\sum_{i=1}^{n} a_{ji}(x_i-p_{ji})^2\right]$; c, a and p are constants given in Torn and Zilinskas (1989)	-3.86278 at x = {0.11464, 0.555649, 0.852547}	4 local minima
Rosenbrock (ROS_N)[#]	2, 5, 10, 15, 20 and 30; $-5 \le x_i \le 10$	$\sum_{i=1}^{n}\left[100(x_i^2-x_{i+1})^2+(x_i-1)^2\right]$	0 at x = {1, …..,1}	Several local minima
Zakharov (ZAK_N)[#]	2, 5, 10, 15, 20 and 30; $-5 \le x_i \le 10$	$\left(\sum_{i=1}^{n} x_i^2\right)+\left(\sum_{i=1}^{n} 0.5ix_i^2\right)^2+\left(\sum_{i=1}^{n} 0.5ix_i^2\right)^4$	0 at x = {0,…..,0}	Several local minima
Modified Himmelbalu (mHB)[*]	2; $-6 \le x_i \le 6$	$(x_1^2+x_2-11)^2+(x_1+x_2^2-7)^2+0.1((x_1-3)^2+(x_2-2)^2)$	0 at x = {3, 2}	Four local minima
Rastrigin (RA_N)[*]	2, 5, 10, 15, 20 and 30; $-600 \le x_i \le 600$	$10n+\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i))$	0 at x = {0, ..., 0}	Thousands of local minima
Griewank (GW_N)[*]	10, 15, 20 and 30; $-600 \le x_i \le 600$	$\sum_{i=1}^{n} x_i^2/d-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1$	0 at x = {0, ..., 0}	Hundreds of local minima

[#] Moderate functions; * Difficult functions.

4.4 Phase Equilibrium Calculations

Phase equilibrium calculations play a significant role in the design, simulation and optimization of chemical processes. Development of robust and efficient method for phase equilibrium problems has long been a challenge and still it is. The objective is to calculate the number of moles of each component in each phase at equilibrium given the pressure, temperature, components and feed composition. In general, methods for multiphase equilibrium calculations can be categorized into two types: equation-solving approach and Gibbs free energy minimization (Teh and Rangaiah, 2002). In the former approach, a set of non-linear equations arising from mass balances and equilibrium relationships are solved whereas Gibbs free energy function is minimized in the latter approach. The equality of chemical potential criterion, which is only the necessary condition for free energy minimization, is used in equation-solving approach which makes the method inadequate to find the global optimum. Hence, several researchers have applied deterministic (e.g., McDonald and Floudas, 1995a; and Burgos-Solórzano et al., 2004) and stochastic (e.g., Rangaiah, 2001) global optimization methods to Gibbs free energy minimization; a review of these works can be found in Teh and Rangaiah (2003).

Recently, Nichita et al. (2002a) applied a tunneling method for PEC. In this method, the calculations are organized in a stepwise manner: solving PS analysis by minimization of tangent plane distance function (TPDF) using tunneling method followed by flash calculation using Newton-Raphson method. The results show that the tunneling method is reliable and efficient for multiphase equilibrium problems. However, tunneling method is not tested for the direct Gibbs free energy

minimization in a single step. Iglesias-Silva et al. (2003) proposed an algebraic method for PEC which includes free energy minimization when the number of phases is known a priori. The method uses orthogonal derivatives, the tangent plane condition and mass balances to reduce the Gibbs free energy minimization to solving a system of non-linear equations. The results show that the method has good convergence rate. Burgos-Solórzano et al. (2004) solved phase equilibrium calculations by combining reliable deterministic techniques such as interval Newton technique with local optimization procedure. Their results show that the procedure is good for high pressure chemical and multiphase equilibrium using cubic equation of state models. Mekapati and Rangaiah (2006b) evaluated random tunneling algorithm (RTA) with some modifications for phase equilibrium calculations by free energy minimization in one step. The results show that RTA successfully located the global minimum for most of the examples but the reliability of the method is low for problems having a local minimum value comparable to the global minimum value. More details of the mathematical formulation for these problems are discussed under section 2.5 (Chapter 2).

Phase equilibrium examples considered in this study include vapor-liquid equilibrium (VLE), liquid-liquid equilibrium (LLE) and vapor-liquid-liquid equilibrium (VLLE) examples involving multiple components (2 to 8 components) and popular thermodynamic models. The feed composition, operating conditions and the thermodynamic models used for each example, and global and local minima for all these examples are available in Teh and Rangaiah (2003), and hence not repeated here. Along with local minima, there are trivial solutions for several examples at

which equilibrium composition equals to the feed composition (Teh and Rangaiah, 2002).

4.5 Parameter Estimation Problems

Parameter estimation in models plays a significant role in developing better mathematical models, which are used for understanding and analyzing the physical, chemical and biological systems. Parameter estimation involves fitting the model to experimental data by minimizing the objective function such as sum of squared errors (SSE) between the experimental and predicted values. The objective function becomes non-convex if the model is highly non-linear and the use of traditional optimization techniques such as steepest descent, Newton and quasi-Newton methods fail to provide the best (global) solution. Hence, application of global optimization to estimating the parameters in dynamic systems is receiving greater attention and interest in the recent past. Esposito and Floudas (2000) applied branch and bound method for parameter estimation in differential and algebraic systems (Tjoa and Biegler, 1991) using both collocation based and integration approaches. The results showed the potential of branch and bound method for solving parameter estimation problems. Later, Katare et al. (2004) studied the same problems using GA combined with a local optimization technique (hybrid GA), and their results show that GA is computationally efficient compared to branch and bound method. In this study, DETL is evaluated for parameter estimation problems in dynamic systems studied by Esposito and Floudas (2000) and Katare et al. (2004), and its performance is compared to that of DE, TS and MDE. Example names, differential equations associated with each example and bounds on parameters are summarized in Table 4.2.

More details about the data, number of local and global minima are given in Esposito and Floudas (2000).

Table 4.2: Parameter Estimation Problems

Example Number	Application	Differential Equations	Bounds on Parameters
1	First order irreversible chain reaction	$\frac{dz}{dt} = -\theta_1 z_1$ $\frac{dz_2}{dt} = \theta_1 z_1 - \theta_2 z_2$	$0 \le \theta \le 10$
2	First order reversible reaction	$\frac{dz_1}{dt} = -\theta_1 z_1 + \theta_2 z_2$ $\frac{dz_2}{dt} = \theta_1 z_1 - (\theta_2 + \theta_3) z_2 + \theta_4 z_3$ $\frac{dz_3}{dt} = \theta_2 z_2 - \theta_4 z_3$	$0 \le \theta_1, \theta_2 \le 10$ $10 \le \theta_3, \theta_4 \le 50$
3	Catalytic cracking of gas oil	$\frac{dz_1}{dt} = -(\theta_1 + \theta_3) z_1^2$ $\frac{dz_2}{dt} = \theta_1 z_1^2 - \theta_2 z_2$	$0 \le \theta \le 20$
4	Methanol-to-hydrocarbon process	$\frac{dz_1}{dt} = -\left(2\theta_1 - \frac{\theta_1 z_2}{(\theta_2 + \theta_5) z_1 + z_2} + \theta_3 + \theta_4\right) z_1$ $\frac{dz_2}{dt} = \frac{\theta_1 z_1 (\theta_2 z_1 - z_2)}{(\theta_2 + \theta_5) z_1 + z_2} + \theta_3 z_1$ $\frac{dz_3}{dt} = \frac{\theta_1 z_1 (z_2 + \theta_5 z_1)}{(\theta_2 + \theta_5) z_1 + z_2} + \theta_4 z_1$	$0 \le \theta \le 20$
5	Lokta-Volterra problem	$\frac{dz_1}{dt} = \theta_1 z_1 (1 - z_2)$ $\frac{dz_2}{dt} = \theta_2 z_2 (z_1 - 1)$	$0.1 \le \theta \le 10$

4.6 Implementation and Evaluation of DETL

The code for DETL is written in FORTRAN and the codes for TS and DE are taken from Teh and Rangaiah (2003), and the website www.icsi.berkeley.edu/~storn/code.html respectively. The code for MDE is obtained by implementing the updation of the population after generating each individual in the DE code itself. A local optimization technique (quasi-Newton) is used at the end of all the three methods to improve the computational efficiency and accuracy of the final

solution. For the quasi-Newton method, a subprogram, DBCONF in the IMSL software is used; this subprogram uses BFGS formula to update the Hessian matrix. For one of the examples (example 2 in Table 4.2) in parameter estimation problems, a direct search method, namely, Nelder-Mead Simplex method is used for local optimization. This is because the objective function is found to be flat near the global minimum region resulting inaccurate solution with gradient-based methods combined with numerical derivatives. To solve the differential equations in parameter estimation problems, the IMSL subprogram: DIVPAG based on Gear's method, is used.

The methods are evaluated based on both reliability and computational efficiency in locating the global minimum. Reliability is measured in terms of success rate (SR) (i.e., number of times the algorithm located the global minimum to the specified accuracy out of 100 trials) and the computational efficiency is measured in terms of number of function evaluations (NFE) required to locate the global minimum. NFE is the average over only the successful runs out of 100 trials, and include both the function calls for evaluating the objective and the function calls for the numerical gradient in the local optimization. The stopping/termination criterion used in this study is the maximum number of generations (Gen_{max}) or maximum number of successive generations (Sc_{max}) without improvement in the best function value. This criterion is used instead of convergence to the global minimum as a stopping criterion because, in practical applications, global minimum is unknown '*a priori*' and also accurately. A trial/run is said to be successful only if the global optimum is obtained with an absolute error of 10^{-6} or less in the objective function value.

4.7 Parameter Tuning

The parameters of DE, TS, MDE and DETL are tuned consistently using test functions which are found to be challenging in the preliminary trials; GP, ES, ROS_5, ROS_{10} and ROS_{20} are chosen for moderate functions whereas RA_{10}, RA_{20}, GW_{10} and GW_{20} are selected for difficult functions. Examples 5, 9 and 10 are chosen for phase equilibrium calculations and Lokta-Volterra (example 5 in Table 4.2) problem is chosen for parameter estimation problems. Tuning is carried out by varying one parameter at a time while keeping the rest fixed at their nominal/recent optimum values. The nominal parameter values chosen for TS are: $N_t=N_p=10$, $\varepsilon_t=\varepsilon_p=0.01$, $h_n=0.5$, $NP_{init}=20N$ (where N is the dimension of the problem), $N_{neigh}=2N$ (subjected to a minimum of 10 and maximum of 30), $Sc_{max}=5N$ and $Iter_{max}=50$, and for DE, A=0.4, CR=0.1, NP=50, $Sc_{max}=5N$ and $Gen_{max}=50$. The same parameter values are used for MDE and DETL also; and the additional parameters in DETL are tr = 1×10^{-8} and tls = 10. The nominal values are chosen based on the optimum values available in the literature for TS (Chelouah and Siarry, 2000), and based on the preliminary numerical experience for DE and DETL.

The optimal parameter values obtained for all the methods are given in Table 4.3. The optimal parameter values obtained for DE are equally valid for MDE as mentioned in the literature (Babu and Angira, 2006). Even though the stopping criterion used for MDE in this study is different from that in Babu and Angira (2006), the fast convergence nature of MDE can be captured via the parameter Sc_{max} (i.e., as MDE converges faster, Sc_{max} reaches its optimum value earlier compared to DE thus giving lower NFE compared to DE). The optimum values for DETL are kept the same

as for DE and MDE except for additional parameters tr and tls to have a fair comparison; tr and tls are tuned separately and the optimum values are given in Table 4.3.

Table 4.3: Optimal Values of Parameters in TS, DE, MDE and DETL

Parameters	Benchmark Problems		PEC	PEP
	Moderate	Difficult		
TS				
N_t and N_p	10	10	10	10
ε_t and ε_p	0.01	0.01	0.02	0.01
NP_{init}	20N	100N	20N	20N
N_{neigh}	2N	2N	2N	2N
h_n	0.5	0.5	0.5	0.5
$Iter_{max}$	50N	100N	100N	50N
Sc_{max}	6N	50N	2N	6N
DE, MDE and DETL				
A	0.5	0.2	0.3	0.7
CR	0.5	0.5	0.9	0.5
NP	20	20	Min {40N, 120}	30
Gen_{max}	30N	60N	70	40N
Sc_{max}	7N	12N	7N	15N
tr^*	$N\times10^{-3}$	$N\times10^{-6}$	$N\times10^{-3}$	$N\times10^{-3}$
tls^*	50	50	50	50

* These are the two additions parameters in DETL and not involved in DE and MDE.

The optimum parameter values obtained for TS for moderate functions are the same as the nominal parameter values because the latter have been taken from Chelouah and Siarry (2000), which are already proven to be optimum. The optimal values of a few parameters are slightly different from one set of problems to another set due to the different characteristics of each group (i.e., moderate functions have few local minima, difficult functions have huge number of local minima, phase equilibrium calculations have comparable minima and parameter estimation problems have several minima). In this study, tuning is carried out to obtain the optimal parameter values for each set of problems rather than for each problem so that they are applicable to other problems of similar nature.

4.8 Results and Discussion

4.8.1 Benchmark problems

***Moderate functions*:** Each benchmark problem is solved 100 times, each time with a different random number seed, which will change the initial estimate(s) in the subsequent iterations, and the performance of each algorithm is compared in terms of SR and NFE. The results are averaged over 100 trials and are given in Table 4.4. It is clear from these results that SR of DE is better than that of TS except for Easom function, whereas the computational efficiency of the latter is better than that of the former. This could be probably because of the different escaping mechanisms from the local minim associated with each method. DE, MDE and DETL use crossover and mutation operations whereas TS uses the best point found in the current generation to generate neighbors for the next generation even though it is worse than those obtained in the previous iterations.

SR of TS and DE is less for Easom functions because of the flat objective function. As the function becomes flat, the neighbors generated in TS will have the same value leading to trapping of the search in that region. On the other hand, MDE and DETL escaped from the flat region with the help of mutation and crossover. Despite having the same escaping mechanism as in MDE and DETL, SR of DE is even less than that of TS for this function. This could be because of the optimum value obtained for the parameter, Sc_{max} considering several moderate functions rather than based on a single function, may not be the optimum for Easom function. For example, using maximum number of generations (Gen_{max} for DE and $Iter_{max}$ for TS)

Table 4.4: Results for Benchmark Problems – Moderate Functions

Function	DE		TS		MDE		DETL	
	SR	NFE	SR	NFE	SR	NFE	SR	NFE*
GP	100	1107	99	301	100	1068	100	740 (711+29)
ES	76	2430	85	433	97	2370	97	1615 (1586+29)
SH	96	720	92	355	98	702	100	740 (707+33)
ROS_2	100	780	100	475	100	803	100	767 (709+58)
ZAK_2	100	1218	100	343	100	1209	100	468 (450+18)
H_3	100	1846	100	386	100	1826	100	845 (799+46)
ROS_5	97	3025	79	2081	98	3102	95	2958 (2768+190)
ZAK_5	100	3054	100	1294	100	3026	100	1329 (1236+93)
ROS_{10}	97	6677	78	8541	95	6675	96	6536 (5895+641)
ZAK_{10}	100	6124	100	8473	100	6039	100	2890 (2639+251)
ROS_{20}	95	14326	75	22074	92	14383	92	13507 (11140+2367)
ZAK_{20}	100	12627	100	19157	100	12041	100	6893 (6167+726)

* The two numbers in brackets are respectively NFE required for DETL excluding local optimization and NFE for local optimization.

alone as a stopping criterion (i.e., instead of using both Gen_{max} and Sc_{max}), SR of DE increased to 100% with NFE of 3224 whereas SR of TS is still at 90% with NFE of 1040. On the other hand, both MDE and DETL achieved high SR (97%) with the same Sc_{max} value. This might be because of the early updating of population in MDE compared to DE and the diversity introduced by TS concept in DETL. The reliability of all the methods is high for Shubert function though it has huge number (760) local minima. This is possibly due to the 18 global minima it has, and reaching any one of them is sufficient to achieve the best solution. Global search by TS seems to be

affected by narrow global minimum region resulting in SR of 75-80% for Rosenbrock functions.

Even though TS has low SR, its NFE is less compared to DE especially for small variable (<10 variables) problems, probably because of avoiding re-visits during the search in TS. Though NFE of DE is higher than that of TS for most of the functions, it is less for Rosenbrock and Zakharov functions with more variables. This may be because of narrow global minimum regions associated with Rosenbrock functions, which in turn require large $Iter_{max}$ value for TS (because $Iter_{max}$ is tuned based on the challenging functions in the moderate group which include Rosenbrock functions also). Though TS located global minimum region for Zakharov functions in a few iterations, NFE is high for these functions as the algorithm terminates only after satisfying termination criterion (either $Iter_{max}$ or Sc_{max}).

The results for MDE are also given in Table 4.4. They show that the reliability of MDE is almost comparable to that of DE while NFE of the former is less than that of the latter. This is due to the updation of population members after generating each individual compared to the original DE, in which population updating is carried out only after generating all members. Also, NFE required by MDE in this study (Table 4.4) is less than those given in Babu and Angira (2006) for the same functions. This is because of the different stopping criterion used in this study (i.e., maximum number of generations or maximum number of successive generations without improvement in the objective function value) compared to the convergence to the global minimum used by Babu and Angira (2006), the use of local optimization technique at the end

and the results in this study are based on the average of 100 trials whereas results in Babu and Angira (2006) are based on average of 10 trials only.

SR of DETL is comparable to that of DE and MDE, and its NFE is less than that of both DE and MDE. For DETL, NFE without local optimization and NFE required for local optimization technique are also given in the last column of Table 4.4; these details indicate that NFE required by local optimization is around 7% in the overall NFE, and is slightly high (around 10%) for Rosenbrock functions due to the presence of narrow global minimum regions associated with them. The percentage reduction in NFE of DETL compared to DE and MDE are given in Table 4.5. Though the percentage reduction in NFE for Rosenbrock functions is only around 2 to 6%, it is high for Zakharov functions (around 45 to 50%). One exception is Shubert function for which NFE of DETL is slightly higher (as can be seen by the negative value in Table 4.5) compared to DE and MDE. Overall, an average reduction of 29% and 25% in NFE of DETL is achieved compared to DE and MDE respectively. This shows the potential of DETL especially for engineering applications where objective function evaluation requires considerable computational time.

Table 4.5: % Reduction in NFE of DETL for Benchmark Problems – Moderate Functions

Function	Compared to DE	Compared to MDE
GP	33	31
ES	34	32
SH	-3	-5
ROS2	2	4
ZAK2	62	61
H3	54	54
ROS5	2	5
ZAK5	56	56
ROS10	2	2
ZAK10	53	52
ROS20	6	6
ZAK20	45	4

$$\text{Note: \% Reduction} = \left(\frac{\text{NFE of DE or MDE} - \text{NFE of DETL}}{\text{NFE of DE or MDE}} \right) \times 100$$

Difficult functions: The performance results for difficult functions (Table 4.6) show that the reliability of DE is significantly better than that of TS; SR of TS is zero for half of these problems and low SR for the others: mHB, RA_2, GW_5, GW_{15} and GW_{20}. TS failed to locate the global minimum in almost all cases, being trapped into local minima because of huge number of local minima associated with these functions. On the contrary, DE escaped from the local minima using mutation and crossover resulting high SR (≈ 90%). The reliability of MDE and DETL is also high and comparable to that of DE. This is because escaping mechanism from the local minima is the same in all the three (DE, MDE and DETL) methods.

The NFE of MDE is slightly less than that of DE, and that of DETL is less than both DE and MDE (Table 4.6). The NFE of all methods is increasing with the number of variables due to the increase in solution space which makes all the algorithms to generate more number of points to locate the potential regions.

Table 4.6: Results for Benchmark Problems – Difficult Functions

Function	DE		TS		MDE		DETL	
	SR	NFE	SR	NFE	SR	NFE	SR	NFE*
MHB	89	1334	65	1531	88	1325	97	1607 (1581+26)
RA2	81	2312	34	1960	83	2310	100	1409 (1383+26)
RA5	100	6048	---	---	100	6049	97	3767 (3675+92)
RA10	98	12076	---	---	95	12067	99	7977 (7757+220)
RA15	95	18106	---	---	94	18092	91	13080 (12751+329)
RA20	89	24107	---	---	74	24105	83	20003 (19578+425)
GW5	90	6097	12	5068	91	6104	94	6056 (5959+97)
GW10	97	12245	---	---	95	12160	95	11875 (11617+258)
GW15	100	18036	7	41705	98	18029	100	11991 (11507+484)
GW20	99	23999	31	56101	100	23383	100	13363 (12602+761)

* The two numbers in brackets are respectively NFE required for DETL excluding local optimization and NFE for local optimization.

On average, NFE required by the local optimization in DETL (Table 4.6) is around 3% for these functions. The average percentage reduction in NFE for DETL compared to DE and MDE are given in Table 4.7. NFE of DE and MDE is less compared to that of DETL for mHB function. This may be because of the premature convergence of DE and MDE to local minima resulting less NFE along with less SR (88-89%) compared to DETL (97%). For difficult benchmark problems, the average percentage reduction in NFE by DETL is 22% and 21% compared to DE and MDE respectively.

Table 4.7: % Reduction in NFE of DETL for Benchmark Problems – Difficult Functions

Function	Compared to DE	Compared to MDE
MHB	-20	-21
RA_2	39	39
RA_5	38	38
RA_{10}	34	34
RA_{15}	28	28
RA_{20}	17	17
GW_5	1	1
GW_{10}	3	2
GW_{15}	34	33
GW_{20}	44	43

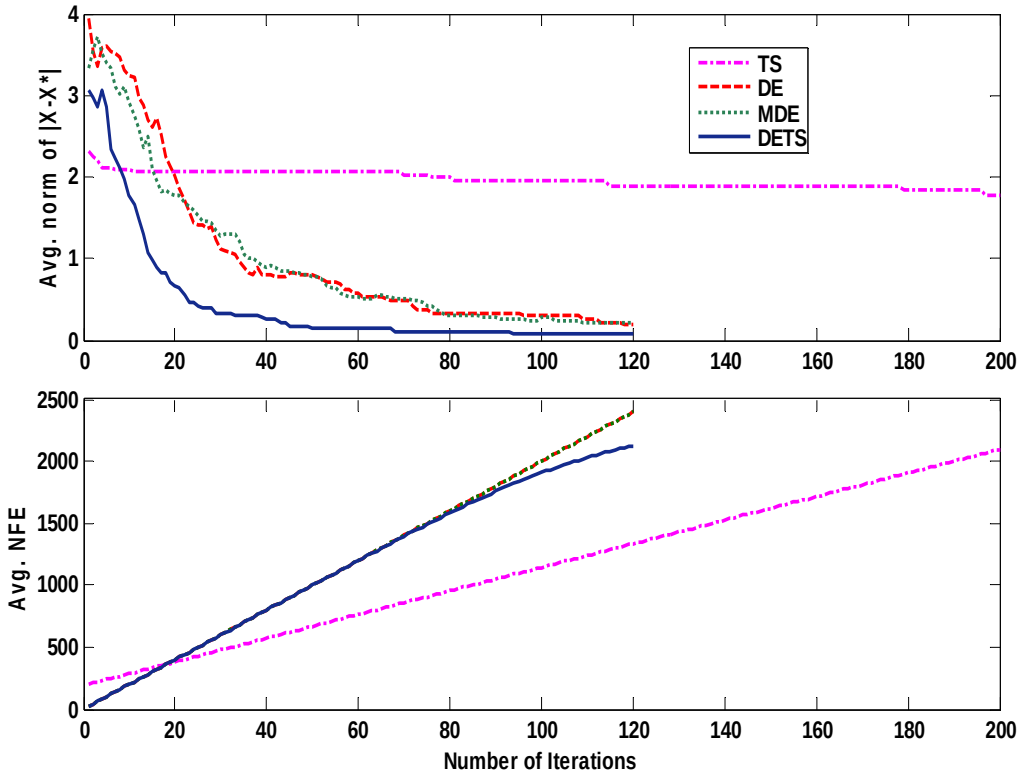
Note: $\%\ \text{Reduction} = \left(\frac{\text{NFE of DE or MDE} - \text{NFE of DETL}}{\text{NFE of DE or MDE}} \right) \times 100$

The performance of DETL is also compared to the other methods (DE, MDE and TS) by plotting convergence curves between (i) average (over 100 trials) of |X-X*| versus the number of iterations, and (ii) average NFE (over 100 trials) versus the number of iterations. Here, X is the best (global) solution obtained by the algorithm in the trial and X* is the true global optimum (Table 4.1). As shown in Figure 4.3a, the convergence of DETL is faster compared to DE, TS and MDE for mHB function. Norm of |X-X*| using DETL after 20^{th} iteration is 0.6774 whereas that of DE, MDE and TS is 2.018, 1.774 and 2.018 respectively. Even though the norm of MDE is lower than that of DE, it is higher than that of DETL. NFE required by DE, TS, MDE and DETL after 120 iterations is 2400, 1334, 2400 and 2119 respectively. This clearly shows that the diversity introduced in DE using TS concept has considerable influence on its performance. The convergence of MDE is slightly better than that of DE but is less that that of DETL. The convergence curve of TS is faster in the initial iterations (the average norm dropped to 2.078) itself but it does not improve over

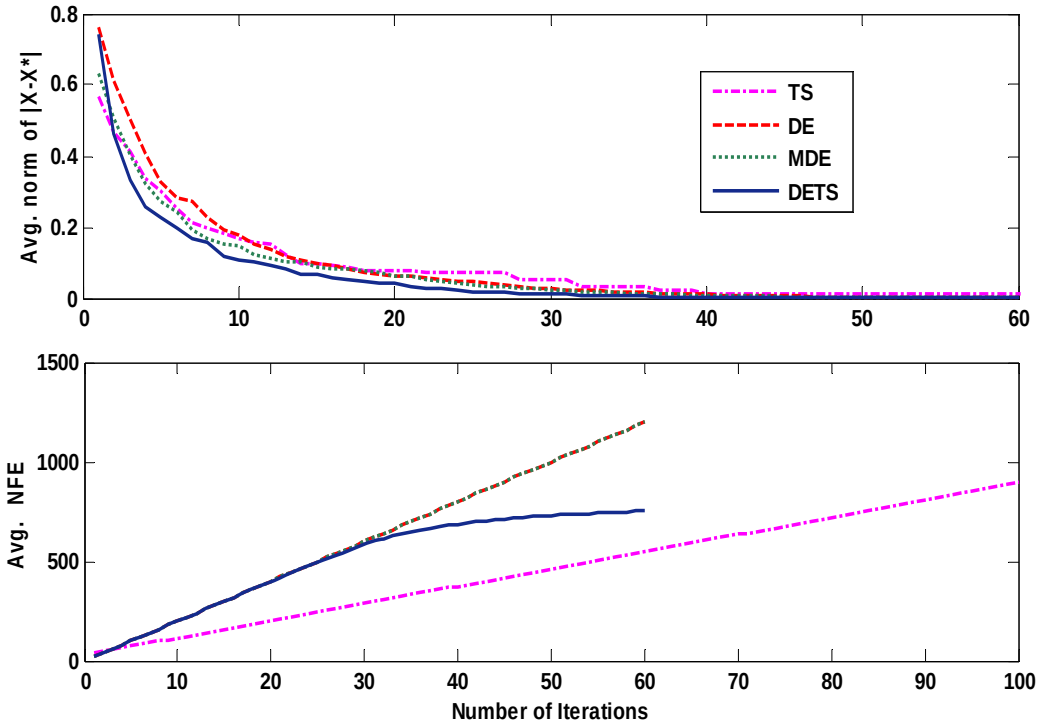
further iterations as it is stuck at the local minima. The results in Figure 4.3a to 4.3c are given for up to the maximum number of iterations, and is different for TS and other methods (Table 4.2).

NFE of DE, MDE and DETL are almost same in the initial stages whereas that of DETL is decreasing as the number of generations increase (Figure 4.3a). As the number of generations increase, members of the population come close to one another to a potential (global) region and so many newly generated members will be near to those in the tabu list and these members are not evaluated in DETL resulting in low NFE. The convergence curves for GP function are given Figure 4.3b. DETL shows faster convergence compared to TS after a few initial iterations, DE and MDE overall the iterations. Though NFE of DETL is high for this function compared to TS, it is less than that of both DE and MDE. The flat profile of Average NFE with DETL shows the function evaluations are getting reduced as the members become close enough. The convergence curves for RA_2 function given in Figure 4.3c, show that the convergence rate of DETL is faster compared to DE, MDE and TS resulting less NFE compared to all the methods (Table 4.6). Though the convergence rate of TS is faster compared to DETL in the earlier iterations, it does not improve much in further iterations and it is stuck at the local minima. These results show that the efficiency of DETL is high compared to DE, MDE and DETL especially for the functions having huge number of local minima (such as Rastrigin functions). Though the improvement in the convergence of DETL is not the same for all the functions, its performance is always better than that of DE and MDE.

(a)


Avg. norm of |X-X*|
TS
DE
MDE
DETS
Avg. NFE
Number of Iterations

(b)


Avg. norm of |X-X*|
TS
DE
MDE
DETS
Avg. NFE
Number of Iterations

(c)

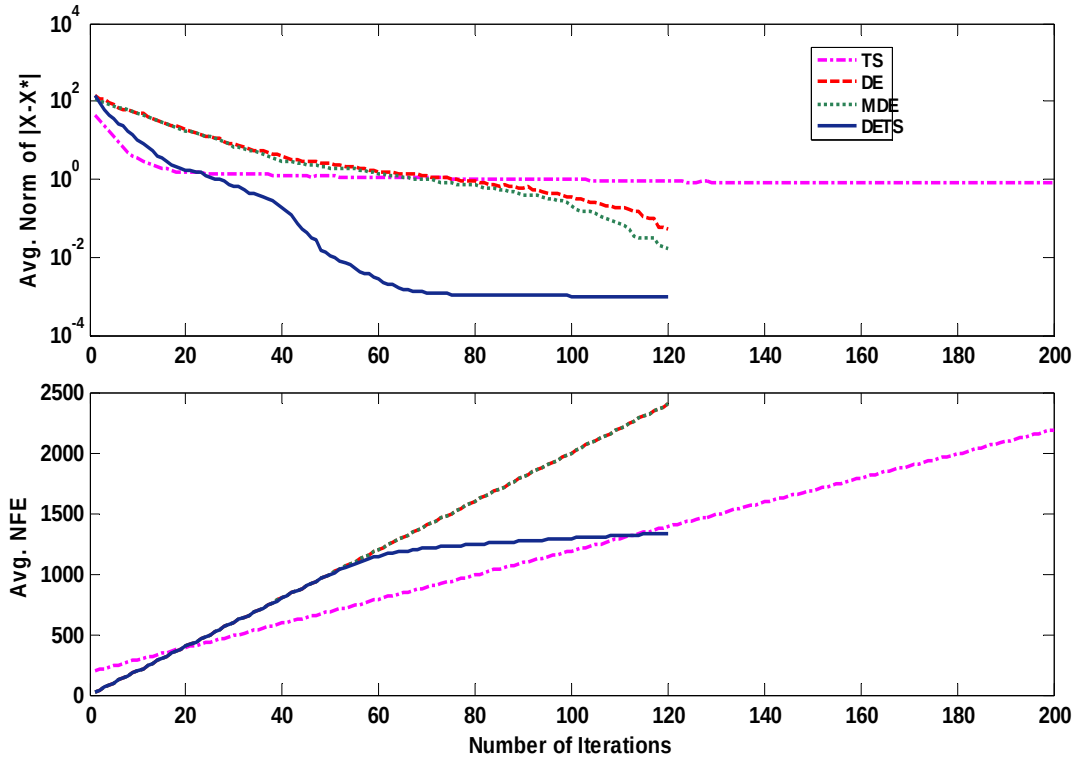


Figure 4.3: Convergence profiles for solving (a) modified Himmelblau, (b) Goldstein and Price, and (c) Rastrigin functions, by TS, DE, MDE and DETL.

4.8.2 Phase Equilibrium Calculations

All the results obtained from solving each phase equilibrium problem 100 times are summarized in Table 4.8. The reliability of DE, MDE and DETL is high, and is better compared to TS; the one exception is example 4 for which all methods fail. SR of TS is around 100% for both Vapor-liquid equilibrium (VLE) and Liquid-liquid equilibrium (LLE) examples except for example 8 due to the presence of comparable minima (function value at global minimum and trivial solution is -0.360253 and -0.354340 respectively). As the minima are comparable, better regions (i.e., regions where the function value is better than that of the last local

minimum)become narrower and narrower, and the algorithm fails to explore them resulting in low SR. For vapor-liquid-liquid equilibrium (VLLE) examples, SR of TS is low compared to DE, MDE and DETL due to the presence of comparable minima (objective function value at local and global minima are -1.334404 and -1.334461 for example 9, and are -1.233294 and -1.235896 for example 10 respectively) and more number of variables (6). Reliability of DE, MDE and DETL is close to 100% for all phase equilibrium examples which shows their superiority over TS in escaping from the local minima. In general, the characteristic of comparable minima in these examples poses a challenge to any optimization technique.

Though TS has low SR, it is computationally efficient compared to the other methods studied, probably because of avoiding re-visits during the search in TS. The SR of DE is higher compared to that of TS whereas its NFE is less than that of TS. NFE of MDE is comparable to that of DE. Despite having high SR, DETL requires less NFE compared to both DE and MDE. On average, NFE required by local optimization technique in DETL is only around 2% of the total NFE. The percentage reduction in NFE of DETL compared to both DE and MDE is given in Table 4.9. By implementing the concept of TS in DE, we are able to reduce the NFE of DETL, on average, by 39% less compared to that of DE, and by 40% less compared to that of MDE, which is a significant reduction.

Table 4.8: Results for Phase Equilibrium Calculations

Example	DE		TS		MDE		DETL	
	SR	NFE	SR	NFE	SR	NFE	SR	NFE*
Vapor-liquid equilibrium								
1 (2)	96	3426	99	1348	96	3930	98	2118 (2081+37)
2 (6a)	100	8564	96	1618	100	8564	100	3436 (3324+112)
3 (6b)	100	8571	96	1639	100	8572	100	3506 (3391+115)
4 (9)	---	---	---	---	---	---	---	---
Liquid-liquid equilibrium								
5 (11a)	96	2562	98	1432	98	2678	96	2070 (2020+50)
6 (11b)	100	5187	100	1359	100	5365	100	2187 (2152+35)
7 (12)	100	5182	100	1367	100	5152	100	2291 (2223+68)
8 (13)	98	7978	94	1575	100	7776	97	5463 (5387+76)
Vapor-liquid-liquid equilibrium								
9 (17)	99	8690	68	5648	98	8717	96	7588 (7354+234)
10 (18)	100	8689	81	5486	99	8692	100	7545 (7366+179)

* The two numbers in brackets are respectively NFE required for DETL excluding local optimization and NFE for local optimization.

Table 4.9: % Reduction in NFE of DETL for Phase Equilibrium Calculations

Example number	Compared to DE	Compared to MDE
1	38	46
2	60	60
3	59	59
4	---	---
5	19	23
6	58	59
7	56	56
8	32	30
9	13	13
10	13	13

Note: $\%\ \text{Reduction} = \left(\frac{\text{NFE of DE or MDE} - \text{NFE of DETL}}{\text{NFE of DE or MDE}} \right) \times 100$

4.8.3 Parameter Estimation Problems

The results of solving parameter estimation problems summarized in Table 4.10 show that SR of all the tested methods is 100% for all examples except for example 5. This could be because of the highly non-convex nature of the objective function in this example (see Figure 5 in Esposito and Floudas, 2000). SR of TS is less for Example 5 whereas that of DE, DETL and MDE is high. SR of MDE is slightly less than that of DE for this example. This may be because of the premature convergence due to the early updating of the population in MDE. SR of DETL is high and equal to that of DE for example 5. For example 2, we have used a direct search method for local optimization, namely, Nelder-Mead Simplex method instead of gradient-based quasi-Newton method. This is to achieve accurate final solution as this function seems to be flat near the global solution resulting slightly inaccurate solutions (absolute error of 10^{-4} instead of specified 10^{-6} or less in the objective function value) with gradient-based methods. Overall, the computational efficiency of TS is better than that of DE, MDE and DETL despite its low SR for example 5.

Table 4.10: Results for Parameter Estimation Problems

Example	DE		TS		MDE		DETL	
	SR	NFE	SR	NFE	SR	NFE	SR	NFE*
1	100	2434	100	585	100	2433	100	1098 (1061+37)
2	100	5032	100	1851	100	5017	100	5028 (4830+198)
3	100	3694	100	585	100	3749	100	3615 (3493+122)
4	100	6158	100	2702	100	6148	100	6151 (6022+129)
5	91	2023	63	699	86	1883	92	1891 (1837+54)

* The two numbers in brackets are respectively NFE required for DETL excluding local optimization and NFE for local optimization.

Though DE has good reliability, it took more NFE compared to TS. MDE has less NFE compared to DE, but has slightly low SR compared to DE for example 5 due to premature convergence. On the other hand, by implementing the concept of TS in DE, DETL achieved good reliability as in DE with less NFE compared to both DE and MDE. NFE of DETL is almost comparable to that of both DE and MDE for examples 2 and 4. NFE required for local optimization technique in DETL is around 3% (on average) for parameter estimation problems solved. The percentage reduction in NFE of DETL, on average, is 13% less compared to DE and 12% less compared to MDE for these functions (Table 4.11).

Table 4.11: % Reduction in NFE of DETL for Parameter Estimation Problems

Example number	Compared to DE	Compared to MDE
1	55	55
2	≈ 0 (-8×10^{-4})	≈ 0 (-2×10^{-3})
3	2	4
4	≈ 0 (1×10^{-3})	≈ 0 (-5×10^{-4})
5	7	≈ 0 (4×10^{-3})

Note: $\%\ \text{Reduction} = \left(\frac{\text{NFE of DE or MDE} - \text{NFE of DETL}}{\text{NFE of DE or MDE}} \right) \times 100$

4.9 Summary

A new method, namely, DETL is proposed by effectively integrating the effective features of DE and TS. It is first tested over a set of benchmark problems, divided into two categories: moderate and difficult, in terms of both reliability and computational efficiency. The results show that the performance of DETL is better for both moderate and difficult functions compared to DE, TS and MDE. The method is then evaluated for challenging phase equilibrium calculations involving multiple components and multiple phases. DETL located the global minimum successfully for these problems with high reliability (close to 100%) and with 39% less NFE compared to DE and 40% less NFE compared to MDE. It is then applied for parameter estimation in dynamic systems; the results show that the SR of DETL is almost 100% with NFE which is 26% and 13% less than that of DE and MDE respectively for these problems. Overall, the performance of DETL is better than that of DE, TS and MDE algorithms. With the significant reductions, it is very attractive for engineering applications where the objective function evaluation involves extensive calculations.

CHAPTER 5

DIFFERENTIAL EVOLUTION WITH TABU LIST FOR SOLVING NLPs AND MINLPs*

Differential evolution (DE), a population-based direct search algorithm is gaining popularity in the recent past due to its simplicity and ability to handle non-linear, non-differentiable and non-convex functions. In this study, a method, namely, differential evolution with tabu list (DETL) is presented for solving constrained optimization problems encountered in chemical engineering. It incorporates the concept of tabu search (TS) (i.e., avoiding re-visits during the search) in DE mainly to improve its computational efficiency. DETL is initially applied to many non-linear programming problems (NLPs) involving 2 to 13 variables and up to 38 constraints. It is then tested on several mixed-integer non-linear programming problems (MINLPs) encountered in chemical engineering practice. The results show that the performance of DETL is better than that of DE and the modified differential evolution (MDE) of Babu and Angira (2006) for both NLPs and MINLPs.

* This chapter is based on the paper – Mekapati Srinivas and Rangaiah, G. P. Differential evolution with tabu list for solving non-linear and mixed-integer non-linear programming problems, Industrial and Engineering Chemistry Research, 46, pp. 7126-7135. 2007c.

5.1 Introduction

Many process design, synthesis, control and scheduling problems in engineering involves formulating and solving non-linear programming (NLP) or mixed-integer non-linear programming (MINLP) problems with constraints. Examples from chemical engineering area include heat exchanger networks (Floudas et al., 1986), pooling problem (Aggarwal and Floudas, 1988), utility and refrigeration systems (Shelton and Grossmann, 1986) and evaporation systems (Hillenbrand, 1984). In general, the problem can be stated as:

$$\begin{aligned} &\text{Minimize} && f(\mathbf{x}, \mathbf{y}) \\ &\text{Subject to} && h_i(\mathbf{x}, \mathbf{y}) = 0, \quad i = 1, 2, \ldots, m1 \\ & && g_j(\mathbf{x}, \mathbf{y}) \geq 0, \quad j = 1, 2, \ldots, m2 \\ & && \mathbf{x}^l_k \leq \mathbf{x}_k \leq \mathbf{x}^u_k, \quad k = 1, 2, \ldots, N \\ & && \mathbf{y}^l_k \leq \mathbf{y}_k \leq \mathbf{y}^u_k, \quad k = 1, 2, \ldots, (p - N) \end{aligned}$$

where **x** and **y** are vectors representing continuous and discrete variables, and h and g are equality and inequality constraints respectively. Number of equality constraints, inequality constraints, continuous variables and discrete variables is respectively m1, m2, N and (p-N). The challenging characteristic of NLP/MINLP problems is the existence of non-convexities either because of the objective function and/or constraints. Use of traditional local optimization techniques to these problems lead to local solutions and hence the study of global optimization techniques to NLP/MINLP problems is of immense interest in the recent past (e.g., Adjiman et al., 1997; Angira and Babu, 2006; Danish et al., 2006).

Generally, global optimization techniques can be broadly divided into two types: deterministic and stochastic. Several deterministic algorithms have been proposed for the solution of NLP/MINLP problems in the literature (e.g., Kocis and Grossmann, 1988; Floudas et al., 1989; Ryoo and Sahinidis, 1995; and Adjiman et al., 1997). Kocis and Grossmann (1988) have solved MINLP problems using outer approximation/equality relaxation (OA/ER) algorithm. OA/ER consists of two phases such that in phase I non-convexities that cut off the global optimum are systematically identified with local and global tests. In phase II, a new master problem is solved to locate the global optimum which may have been overlooked in phase I. Floudas et al. (1989) proposed an approach to solve NLP and MINLP problems, which involves the decomposition of the variable set into two sets: complicating and non-complicating variables. The decomposition of the original problem induces special structure in the resulting sub-problems and a series of these sub-problems are solved based on the generalized Benders decomposition for the global optimum. Ryoo and Sahinidis (1995) proposed a branch-and-bound-based method for MINLP problems. It is based on the solution of a sequence of convex underestimating sub-problems generated by evolutionary subdivision of the search region. Adjiman et al. (1997) proposed two new global optimization techniques for MINLP problems involving functions that are twice-differentiable in continuous variables. Both the techniques are based on the α-branch-and-bound (αBB) global optimization algorithm (Adjiman and Floudas, 1996) for twice-differentiable NLP problems.

Most of the above deterministic methods provide mathematical guarantee to provide the global optimum while exploiting the mathematical structure of the given problem such that the original problem is decomposed into a number of sub-problems

which can be solved easily. On the other hand, stochastic methods are problem independent and converge to the global optimum with probability approaching one as their running time goes to infinity (Torn and Zilinskas, 1989). Further, they are applicable to non-convex and/or non-continuous functions.

Das et al. (1990) studied four different versions of simulated annealing (SA) for scheduling of serial, multi-product batch processes; of the four versions of SA studied, Metropolis algorithm with the Aarts and van Laarhoven annealing schedule was found to give the best results. Ku and Karimi (1991) investigated the usefulness of SA for solving batch process scheduling problems. The results show that SA is versatile and powerful for solving different forms of batch process scheduling problems. Salcedo (1992) proposed an adaptive random search method for NLP and MINLP problems. The results obtained reveal the adequacy of random search methods for non-convex NLPs and small to medium scale MINLPs in chemical engineering. Cardoso et al. (1997) proposed an SA approach for the solution of MINLP problems. The method combines the original Metropolis algorithm (Metropolis et al., 1953) with the non-linear simplex method of Nelder and Mead (1965). The proposed approach is shown to be reliable and efficient especially for larger scale and ill-conditioned problems. Jayaraman et al. (2000) applied ant colony framework for optimal design of batch plants, and has found it to be robust to locate the optimum with less than 0.04% error. Yu et al. (2000) proposed a new algorithm combining both genetic algorithm (GA) and simulated annealing (SA) to solve large scale system energy integration problems. Their results show that the new algorithm can converge faster than either SA or GA alone, and has higher probability in locating the global optimum.

Costa and Oliveira (2001) examined both GAs and evolutionary strategies (ES) for MINLP problems. Their results show that ES exhibit difficulties in highly constrained problems but, in general, they are efficient in terms of function evaluations compared to GA and SA-based M-SIMPSA of Cardoso et al. (1997). Lin and Miller (2004b) studied tabu search (TS) for the solution of NLP and MINLPs. Several constraint handling techniques and initial values for key parameters of TS are also described in this work. The results demonstrate the effectiveness of TS for chemical engineering problems. Danish et al. (2006) presented modified GA by combining various effective schemes (such as simulated binary crossover, polynomial mutation and variable elitism operators) proposed by several researchers. Modified GA is then used to solve several multi-product batch plant design problems and results reported are comparable to or better than those in the literature (Grossmann and Sargent, 1979).

Lampinen (2002) applied DE for solving non-linear constrained functions, and his results demonstrate the ability of DE for NLPs. Recently, Angira and Babu (2006) have applied a modified differential evolution (MDE) for solving seven problems in process synthesis and design. Their results show that MDE converges faster than the original DE of Storn and Price (1997). Due to the simplicity, ease of use and faster convergence properties of DE compared to GA (Karaboga and Cetinkaya, 2004), the former has been used for many applications. In this study, a hybrid method, namely, differential evolution with tabu list (DETL) is presented for solving several NLP and MINLP problems in chemical engineering practice. It incorporates the concept of TS (i.e., avoiding re-visits during the search) in DE mainly to improve its computational efficiency. DETL is inspired by our experience with both DE and TS (e.g., Srinivas

and Rangaiah, 2005 and 2007a); the results for benchmark, phase equilibrium and phase stability problems show that DE is more reliable than TS, whereas the latter is efficient than the former, primarily due to tabu list and avoiding re-visits.

5.2 Description of DETL

The motivation and working principle of DETL are discussed under section 4.2 (Chapter 4). DETL begins with the selection of values for parameters: population size (NP), amplification factor (A), crossover constant (CR), tls, tr, maximum number of generations (Gen_{max}) and maximum number of successive generations (Sc_{max}) without improvement in the best function value. The algorithm (Figure 5.1) generates the initial population of size NP using uniformly distributed random numbers to cover the entire feasible region. The objective function and the constraints are evaluated at each individual and the best one is captured. The evaluated individuals/points are then sent to the tabu list, which will be used to ensure that the algorithm does not search again close to these points.

The three main steps: mutation, crossover, and selection of DE are performed on the population during each generation/iteration. For this, a mutant individual is generated for each randomly chosen target individual (X_{kk}) in the population by

$$V_{kk,\,gen+1} = X_{r1} + A\,(X_{r2} - X_{r3}); \quad kk = 1, 2, ..., NP \tag{5.1}$$

where X_{r1}, X_{r2} and X_{r3} are the three randomly chosen individuals from the current population to produce the mutant individual ($V_{kk,\,gen+1}$). The random numbers r_1, r_2 and r_3 should be different from the running index (kk), and hence NP should be 4 or

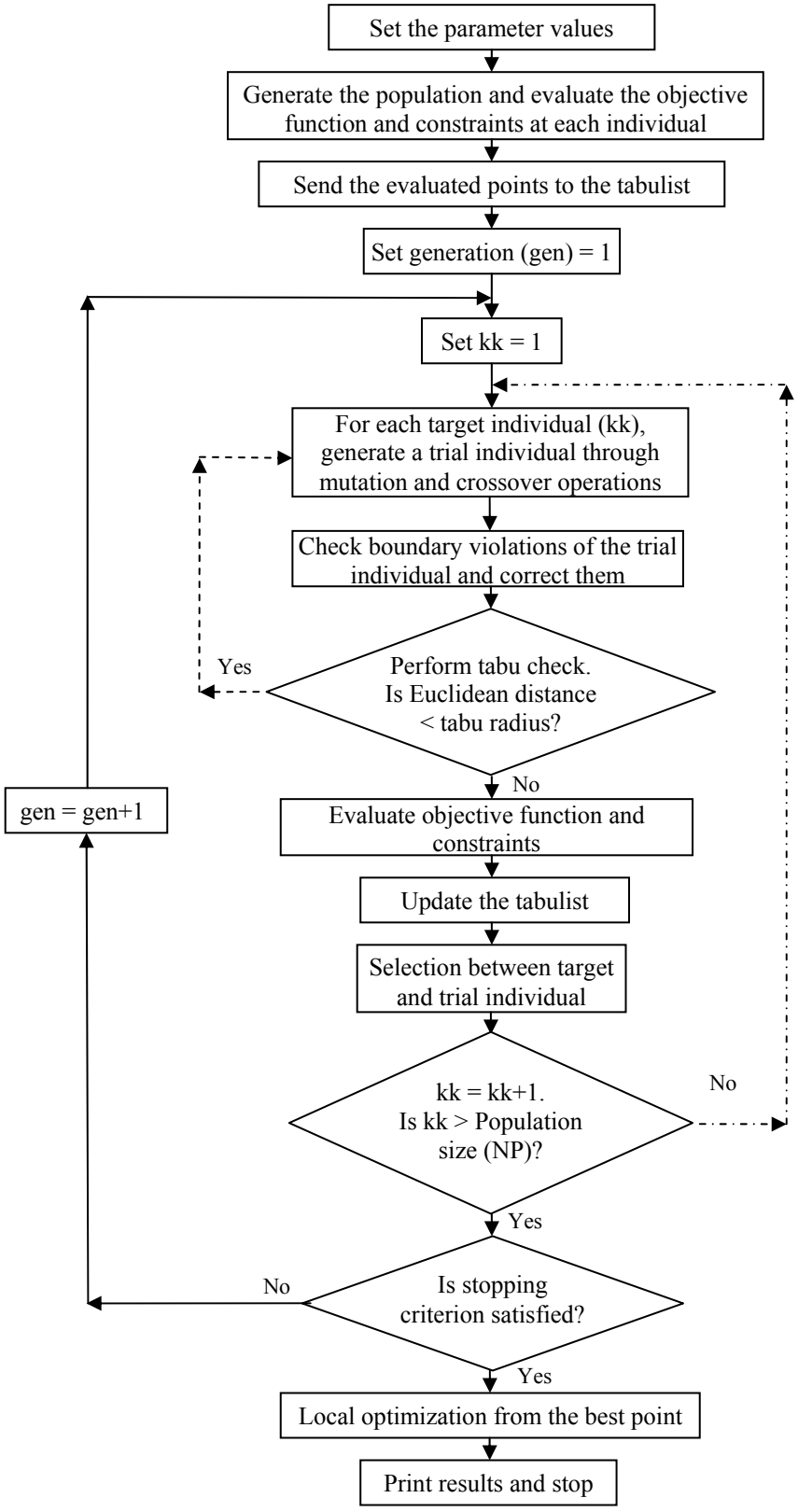


Figure 5.1: Flowchart of DETL

more for mutation. The mutation parameter or amplification factor, A has a value between 0 and 2, and controls the amplification of the differential variation between two random individuals. In the crossover step, a trial individual is generated by copying some elements of the mutant individual to the target individual with a probability of CR. A boundary violation check is performed to check the feasibility of the trial individual; if any bound is violated, the trial individual is either replaced by generating a new individual or forced to the nearest boundary (lower or upper). The trial individual is then compared with the points in the tabu list. If it is near to any of the points in the tabu list, the trial individual is rejected and another point is generated through mutation and crossover operation.

Objective function and constraints are evaluated at the trial individual only if it is away from all the points in the tabu list. After each evaluation, the evaluated point is sent to the tabu list. For example, consider a population of size 30 and tabu list of size 20. Once the first member of the population is generated (i.e., until it is not near to any of the points in the tabu list) and evaluated, it is placed in the first position of the tabu list and the subsequent points will be placed in the corresponding positions. Then, the 21^{st} evaluated point replaces the first point in the tabu list and the subsequent points occupy the corresponding positions. Thus the tabu list in DETL is updated dynamically during the search to keep the latest point(s) in the list by replacing the earliest-entered point(s). In the selection step, a greedy criterion such as fitness (i.e., objective function) value is used to select the better one of the trial and target individuals. If the trial individual is selected, it replaces the target individual in the population immediately and may participate in the subsequent mutation and

crossover operations. If the target individual is better, then it remains in the population and may participate in the subsequent mutation and crossover operations.

The process of generation, evaluation and selection is repeated NP times in each generation. The algorithm runs until the stopping criterion such as maximum number of generations (Gen_{max}) or maximum number of successive generations (Sc_{max}) without improvement in the best function value, is satisfied. The best point thus obtained over all the generations is further refined using a local optimization technique, and is declared as the global optimum.

5.3 Handling Integers and Binary Variables

Within the optimization algorithm, integers and binary variables 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 in contrast to the truncation as in Angira and Babu (2006). Truncation always takes the nearest lower integer value whereas rounding method has equal probability to choose between nearest lower and nearest upper integer values; the latter is unbiased and thus reasonable. For example, if the continuous variable has a value of 2.6 in one case and 2.4 in the second case, then truncation takes the nearest lower integer (2) in both the cases whereas rounding takes the nearest higher integer (3) in the first case and nearest lower integer (2) in the second case. Binary variables are also handled in the same way as for integers except that their bounds are restricted to 0 and 1.

5.4 Handling Constraint and Boundary Violations

In this study, all equality constraints are eliminated by using them to solve for suitable variables and substituting the resulting expressions in the objective function and/or constraints. The reformulated problem for each example is also stated in Appendix B. All inequality constraints are handled using the most popular penalty function method. Penalty function method converts the constrained problem into an unconstrained one by penalizing the infeasible solutions using penalty weights. A high value (1×10^6) 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/subtracted to the objective function, depending upon the minimization or maximization. If many constraints are violated, then each absolute violation is first multiplied with the penalty weight, and all of them are added/subtracted from the objective function value.

Boundary violations are often encountered while performing mutation operation in DE, MDE and DETL. In this study, each and every member violating variable bounds is handled by replacing them with a new member generated randomly between the lower and upper bounds of variables. This approach is referred as random generation (RG) in this work. For MINLPs, another approach, namely, forcing to bounds (FB) in which the boundary violations are forced to the nearest lower/upper boundaries, is also used. This is because many MINLP problems have global solutions at the bounds of decision variables; hence, using FB approach will improve the reliability and efficiency at times. In addition to the RG and FB approaches, a mixed approach in which approximately 50% of the boundary violations are corrected

using RG approach and the remaining with FB approach, is also used for the MINLPs tested. This is achieved by implementing RG and FB approach alternatively.

5.5 Implementation and Evaluation

The FORTRAN code of DE is taken from the website: www.icsi.berkeley.edu/~storn/code.html, and is modified for MDE (as stated in Babu and Angira (2006), to update the population once the better solution is found instead of waiting for the whole population as in DE) and then for DETL by including the tabu list and tabu check. A local optimization technique is used at the end of all the three methods to improve the computational efficiency and accuracy of the final solution. The subprogram, DNCONF in the IMSL software is used for local optimization. For MINLP examples 7 and 8, FSQP program obtained from AEM Design (www.aemdesign.com) is used for local optimization to avoid the unexpected fatal errors and consequent program termination experienced with DNCONF. Both DNCONF and FSQP are based on SQP method and use numerical (finite difference) gradient. As both these programs do not handle binary and integer variables, they are kept at the optimal solution obtained by each global algorithm (DE, MDE and DETL) and only the continuous variables are refined using the local optimization. It is reasonable to assume that the solution found by the global method does not require refining binary and integer variables.

The methods are evaluated based on both reliability and computational efficiency in locating the global optimum. Reliability is measured in terms of success rate (SR) (i.e., number of times the algorithm located the global optimum to the

specified accuracy out of 100 trials). Computational efficiency is measured in terms of number of function evaluations (NFE) required to locate the global optimum. NFE is the average over only the successful runs out of 100 trials, and includes the function calls for evaluating the objective and for the numerical gradient in the local optimization. All the constraints are also evaluated at each function evaluation. NFE is a better indicator than the computational (CPU) time since function evaluation in many applications requires extensive calculations and CPU time is dependent on the computer and implementation. The stopping/termination criterion used in this study is the maximum number of generations (Gen_{max}) or maximum number of successive generations (Sc_{max}) without improvement in the best function value. These criteria are used instead of convergence to the global minimum as a stopping criterion because, in practical applications, global minimum is unknown '*a priori*' and also accurately. A trial/run is said to be successful only if the global optimum is obtained with an absolute error of 10^{-5} or less in the objective function value.

5.6 Non-linear Programming Problems (NLPs)

The applicability of DETL for NLPs is tested by solving many test functions and engineering design problems taken from the literature (Deb, 2000; Babu and Angira, 2006; and Ryoo and Sahinidis, 1995). The examples are carefully chosen so that they have at least two of the following features: (i) multiple minima, (ii) narrow feasible region, and (iii) employed in the recent studies. Problems with no known multiple minima are specifically excluded since this study focuses on global optimization for problems with multiple minima. Number of binary/integer/continuous variables and equality/inequality constraints in each

Table 5.1: Characteristics of the NLP and MINLP problems studied

Example	Constraints (equality + inequality)	Variables (binary+ integer + real)	Reference
NLPs			
1	2 (0+2)	2 (0+0+2)	Deb (2000)
2	38 (0+38)	5 (0+0+5)	Himmelblau (1972)
3	9 (0+9)	13 (0+0+13)	Deb (2000)
4	6 (0+6)	8 (0+0+8)	Deb (2000)
5	4 (0+4)	7 (0+0+7)	Deb (2000)
6	6 (0+6)	5 (0+0+5)	Deb (2000)
7	8 (0+8)	10 (0+0+10)	Deb (2000)
8[a,*]	5 (4+1)	6 (0+0+6)	Ryoo and Sahinidis (1995)
9[a]	0	2 (0+0+2)	Umeda and Ichikawa (1971)
10	1 (0+1)	2 (0+0+2)	Sahinidis and Grossmann (1991)
11[*]	2 (2+0)	4 (0+0+4)	Ryoo and Sahinidis (1995)
12[*]	7 (5+2)	10 (0+0+10)	Visweswaran and Floudas (1990)
13[*]	2 (2+0)	3 (0+0+3)	Ryoo and Sahinidis (1995)
14	4 (0+4)	2 (0+0+2)	Ryoo and Sahinidis (1995)
15	2 (0+2)	2 (0+0+2)	Ryoo and Sahinidis (1995)
16[*]	6 (3+3)	6 (0+0+6)	Ryoo and Sahinidis (1995)
MINLPs			
1[b,c,d]	2 (0+2)	2 (1+0+1)	Ryoo and Sahinidis (1995)
2[b,c,d,e,*]	2(1+1)	3 (1+0+2)	Kocis and Grossmann (1987)
3[b,c,d]	3 (0+3)	3(1+0+2)	Floudas (1995)
4[b,c,d,*]	9 (5+4)	4(1+0+3)	Kocis and Grossmann (1989)
5[b,c,d,e]	9 (0+9)	7 (3+0+4)	Floudas (1989)
6[b,c]	3 (0+3)	5 (0+2+3)	Cardoso et al. (1997)
7[f]	13 (0+13)	10 (0+3+7)	Salcedo (1992)
8[f]	61 (0+61)	22 (0+6+16)	Salcedo (1992)

Note: Some of the examples are also given and used by ([a]) Babu and Angira (2006), ([b]) Angira and Babu (2006), ([c]) Costa and Oliveira (2001), ([d]) Cardoso et al. (1997), ([e]) Salcedo (1992) and ([f]) Grossmann and Sargent (1979)

[*] The number of variables and inequality constraints for the reformulated problem are given in Appendix B.

example is summarized in Table 5.1. The mathematical formulation, global and local optima for each of these problems are given in Appendix B. Some of the problems presented in the literature (e.g., example 2 in Deb, 2000; examples 10 and 12 in Ryoo and Sahinidis, 1995) have typographical errors; hence, the reference from which an example is obtained is also given in Table 5.1. Example 1 is the Himmelblau function

with two non-linear constraints. The feasible region is a narrow, crescent-shaped (Deb, 2000) and only 0.7% of the total search space defined by bounds, posing a significant challenge to optimization problems. Example 2 is the maximization of the net profit of a wood-pulp plant. It has many (38) constraints resulting from material and energy balances and several empirical equations. Example 3 is relatively an easy problem with the objective function and constraints being linear or quadratic. Example 4 is the design of a heat exchanger network problem. Michalewicz (1995) found this problem difficult to solve. Example 5 has feasible region of only about 0.5% of the total search space. Examples 6 and 7 are the test functions with non-linear objective function and constraints. Example 8 is a reactor network design problem. It is difficult to solve as it has a comparable local minimum (i.e., the local minimum value is very close to the global minimum value). Example 9 is the maximization of the yield of a product with respect to reaction time and temperature in a continuous stirred tank reactor. Example 10 is a test function with bilinear constraint. Examples 11 and 12 are the design of an insulated tank and pooling problems respectively. Example 13 has bilinearities whereas examples 14 and 15 are quadratically constrained problems with linear and quadratic objective functions. Example 16 is the design of a three-stage process system with recycle.

5.6.1 Parameter Tuning

The parameters of DE, MDE and DETL (i.e., CR, A, NP, Gen_{max}, Sc_{max}, tr and tls) are tuned using examples 2, 4, 6, 8 and 9 which are found to be difficult in the preliminary testing. Tuning is carried out by varying one parameter at a time with the remaining parameters fixed at their nominal or recent optimal values, in order to

achieve good reliability and also computational efficiency. The nominal parameter values are chosen based on the preliminary numerical experience, are CR = 0.5, A = 0.5, NP = 20, Gen_{max} = 50, Sc_{max} = 5N (where N is the dimension of the problem) for both DE and MDE, and tr = 1×10^{-6} and tls = 20 which are the additional parameters in DETL. The range of values used for each parameter while tuning are 0.1 to 2 for A, 0.1 to 1 for CR, 20 to 100 for NP, 50N to 500N for $Iter_{max}$, and 10N to 20N for Sc_{max}, with a minimum of 6 and a maximum of 10 points in the range. Although the parameters can be fine tuned for each example as performed in Angira and Babu (2006), a common set of optimal parameters is used for each group (NLPs and MINLPs) in this study to find optimal parameters valid for a class of problems. They will be more useful in new applications.

Table 5.2: Optimal values of parameters in DE, MDE and DETL

Parameter	NLPs	MINLPs
DE and MDE		
Amplification factor (A)	0.6	0.5
Crossover constant (CR)	0.6	0.7
Population size (NP)	20	20
Maximum number of generations (Gen_{max})	100N	40N
Max. number of successive generations without improvement in the best function value (Sc_{max})	10N	10N
DETL		
Tabu radius (tr)	$N\times10^{-3}$	$N\times10^{-2}$
Tabu list size (tls)	20	20

All the decision variables are normalized, within the optimization program, between 0 and 1 for consistency, and values reported for tabu radius (tr) in this study are for use with the normalized variables. The optimal parameter values obtained for NLPs are summarized in Table 5.2. The optimal parameters of MDE and DETL (except the additional parameters tr and tls, which are tuned separately) are kept same as those

obtained for DE for a fair comparison. Even then, if any algorithm (DE, MDE or DETL) converges faster, it will be captured by Sc_{max} termination criterion; when the algorithm converges faster, this criterion is satisfied earlier resulting in less NFE.

5.6.2 Results and Discussion

All the examples are solved 100 times, each time with a randomly generated initial estimate. These initial estimates may or may not be feasible (i.e., constraints may not be satisfied). The boundary violations are handled using RG approach for all NLPs since most of these problems have global solutions inside the bounds of the decision variables. The performance results (SR and NFE) are presented in Table 5.3. Success rate of DE, MDE and DETL is 100% for examples 1 to 7 (except for example 4) even though some of them have narrow feasible regions (examples 1 and 5) and one of them (example 2) has many (38) constraints. This shows the reliability of the escaping mechanism (crossover and mutation) in these methods to escape from the local minima. For example 4, the global optimum obtained in this study at x = (579.3006, 1359.9706, 5109.9704, 182.0177, 295.6011, 217.9823, 286.4165, 395.6011) with f = 7049.247720, is slightly better than that with f = 7049.330923 reported in Deb (2000). In a few of the trials, both MDE and DETL converged to a solution at x = (579.3199, 1355.3417, 5114.5942, 182.0188, 295.4162, 217.9811, 286.6025, 395.4162) with f = 7049.255921. This could be due to the flat objective function near the global solution thus leading the gradient-based optimization technique (DNCONF used in this study) converge somewhat prematurely in a few trials. For examples 1 to 3 and 5 to 7, NFE for DETL is less among the methods tested which clearly shows the benefit of avoiding re-visits using the tabu list in DE.

Table 5.3: SR and NFE of DE, MDE and DETL for NLPs

Example number	DE		MDE		DETL	
	SR	NFE	SR	NFE	SR	NFE
1	100	1981	100	1910	100	1132
2	100	10002	100	10043	100	6607
3	100	26074	100	26074	100	21144
4	100	12926	97	13723	95	13209
5	100	9455	100	9347	100	7282
6	100	9651	100	9487	100	7013
7	100	13892	100	13260	100	12183
8	90	1710	98	1993	98	1493
9	75	3348	72	3552	77	1998
10	100	3530	100	3823	100	1370
11	99	3692	97	3609	96	1521
12	100	8024	100	8661	100	4433
13	99	868	99	1055	100	425
14	100	2409	100	2421	100	1650
15	86	2114	77	2379	78	1689
16	100	5427	100	5481	100	2924

For example 4, NFE for DETL is fewer than that for MDE and is slightly higher compared to DE. Although MDE is expected to converge faster compared to DE (Babu and Angira, 2006), NFE for the former is slightly more or comparable to that for the latter for examples 2 and 4; however, for examples 1, 5, 6 and 7, MDE requires slightly fewer NFE compared to DE. This shows that the concept of early updating the members of a population in MDE compared to DE may not improve computational efficiency at times.

For example 8, SR of DE is slightly less (90%) than that (98%) of MDE and DETL because of the presence of comparable minima (i.e., the objective function value at the local and global minimum is -0.3881 and -0.38881 respectively, a difference of < 0.2%) in this example. Once any algorithm reaches a comparable local minimum, exploring regions where the function value is better compared to that at the

local minimum becomes very difficult resulting in low SR. For example 9, SR of all the methods is less, at around 75%. This is because of the existence of a ridge (creates numerous local solutions) as shown in Babu and Angira (2006) for this example. For examples 8 and 9, NFE for MDE is slightly more than that for DE, and that for DETL is the least. SR of DE, MDE and DETL is almost 100% for examples 10 to 16 except for example 15. Though all these methods were able to escape from two of the local minima (at f = 10.631 and at f = 10.354) in example 15, they were trapped in a constrained minimum at x = {2.60555, 0} with f = -86.422205 in several trials resulting in low SR. NFE for MDE is slightly more than that for DE whereas DETL requires the lowest NFE for examples 10 to 16.

Overall, SR of DETL is high and comparable to that of DE and MDE, and its NFE is fewer than that for both DE and MDE. The percentage reduction in NFE for DETL compared to that for DE and MDE is summarized in Table 5.4. NFE for DE is slightly less compared to MDE for many examples tested, making the average reduction in NFE for MDE compared to DE negative (-4.2%). On the other hand, the average reduction in NFE for DETL compared to DE and MDE is 33% and 35% respectively.

The comparable performance of DE and MDE found in this study is different from the finding of Babu and Angira (2006). The best results (NFE) reported for DE and MDE in the previous study are respectively 37,810 and 31,877 for example 4, 2074 and 1860 for example 8, and 7996 and 7351 for example 9; SR is either 99 or 100%. In other words, percentage reduction in NFE of MDE compared to DE is 16%, 10% and 8% for example 4, 8 and 9 respectively. The other two constrained NLP

examples tested by Babu and Angira (2006) have no known local minima, and hence are not selected in the present study.

Table 5.4: Percentage reduction in NFE for NLPs

Example number	MDE Compared to DE	DETL Compared to DE	DETL Compared to MDE
1	3.58	42.86	40.73
2	-0.41	33.94	34.21
3	0	18.91	18.91
4	-6.17	-2.19	3.75
5	1.04	22.90	22.09
6	1.69	27.33	26.08
7	4.55	12.30	8.12
8	-16.55	12.69	25.09
9	-6.09	40.32	43.75
10	-8.30	61.19	64.16
11	2.25	58.80	57.86
12	-7.94	44.75	48.82
13	-21.54	51.04	59.72
14	-0.49	31.51	31.85
15	-12.54	20.10	29.00
16	-0.99	46.12	46.65
Average	-4.24	32.66	35.05

The differences in the results of DE and MDE in Babu and Angira (2006) and in the present study are due to several reasons: (i) termination criterion (convergence to the global minimum is used in Babu and Angira (2006) whereas general termination criteria - Gen_{max} and Sc_{max} are used in this study), (ii) use of common set of optimal parameters for a set of functions in this study instead of for each function as in Babu and Angira (2006), (iii) use of local optimization at the end of each algorithm in this study, and (iv) NFE is calculated based on only the successful trials out of 100 in this study instead of calculating based on all 100 trials as in Babu and Angira (2006). The last one is to avoid under-estimating NFE since convergence to a

local minimum in the failed runs takes fewer NFE. In general, evaluation of the methods in this study is more realistic and has a wider applicability.

5.7 Mixed-Integer Non-linear Programming Problems (MINLPs)

The applicability and efficiency of DETL is also tested using several MINLPs related to process synthesis and design studied by different authors. Most of these problems are non-convex optimization problems and involve binary and integer variables. Examples 1 and 3 are process synthesis and process flow-sheeting problems respectively with non-convexities in the first constraint. Example 2 is also a process synthesis/design problem with non-linear equality constraint. Example 4 is a two-reactor problem, where the selection is to be made among two candidate reactors for minimizing the cost of producing a desired product. Example 5 is process synthesis problem whereas example 6 is a process design problem having multiple global solutions. Examples 7 and 8 refer to the optimum design of multi-product batch plants with several local minima. Number of binary/integer/real variables and equality/inequality constraints in each example as well as the reference from which it is taken are summarized in Table 5.1. The mathematical formulation, global and local optima for each of the problems are given in Appendix B.

5.7.1 Parameter Tuning

All the parameters of DE, MDE and DETL are tuned based on examples 2 to 4 which are found to be difficult in the preliminary trials. The optimal parameter values obtained for each method for MINLPs are also given in Table 5.2. The optimal values

are slightly different from those obtained for NLPs. This could be due to the less number of variables and constraints in MINLP examples (example 8) compared to NLP examples studied. All the examples are solved 100 times, each time with a randomly generated initial estimate. The three approaches - RG, FB and Mixed, for boundary violations are also tested. The performance results (SR and NFE) for all the three methods are given in Table 5.5.

5.7.2 Results and Discussion

SR of all the three methods is close to 100% for example 1 using RG approach. SR is not affected using FB approach for both DE and DETL, and is slightly decreased for MDE; MDE is converged to the nearest local minimum (f = 2.236067) in 11 out of 100 trials. This could be because of the premature convergence associated with early updating of population members in each generation compared to that of DE. For example 2, SR of all methods is less using RG approach. Most of the failed runs converged to a solution at x = 0.852605 and y = 0 with f = 2.557816, which is somewhat comparable to the function value at the global minimum (2.124). By forcing the violated members to the nearest bounds (i.e., using FB approach), SR of all the methods is increased for example 2. SR of MDE is less compared to that of both DE and DETL although it requires less NFE compared to that for DE for example 2. On the other hand SR of DETL is high (94%) and its NFE is significantly less compared to that for both DE and MDE for this example. This clearly shows the efficiency obtained using the tabu list in DE (i.e., DETL).

Table 5.5: SR and NFE of DE, MDE and DETL using RG, FB and Mixed approaches for MINLPs

Example Number	Approach	DE		MDE		DETL	
		SR	NFE	SR	NFE	SR	NFE
1	RG	100	1435	95	1531	100	660
	FB	97	1394	90	1414	100	635
	Mixed	97	1467	94	1554	100	665
2	RG	65	1431	55	1416	74	772
	FB	92	1551	83	1533	94	495
	Mixed	80	1445	78	1487	85	715
3	RG	94	2358	84	2385	86	1359
	FB	39	1890	34	1827	75	1176
	Mixed	62	2078	52	2276	82	1433
4	RG	59	2347	57	2367	51	2148
	FB	96	1934	89	1687	91	726
	Mixed	95	2191	95	2202	95	1582
5	RG	84	5624	67	5624	94	2804
	FB	80	5624	72	5624	84	1777
	Mixed	87	5624	83	5624	93	2800
6	RG	100	4024	100	4024	100	4027
	FB	100	1220	100	1213	100	1005
	Mixed	100	1426	100	1341	100	1444
7	RG	0	---	0	---	0	---
	FB	75	8056	67	7998	60	7021
	Mixed	10	7552	14	7411	10	6532
$7^{\#}$	RG	0	---	0	---	0	---
	FB	100	300108	100	296909	96	298738
	Mixed	100	251600	92	122946	88	119413
$8^{\#}$	RG	0	---	0	---	0	---
	FB	99	380730	93	402588	96	297635
	Mixed	59	361454	32	286109	15	247396

These results were obtained with the following parameter values: A = 0.9, CR = 0.6, NP = 100, Gen_{max} = 300N, Sc_{max} = 20N, tr = $N\times10^{-2}$ and tls = 100.

For example 3, SR of DE, MDE and DETL using RG approach is better compared to that of FB approach. This is because most of the failed runs converged to a local solution at x = {0.2, -1} and y = 0 with f = 1.25, where two variables (x_2 and y) are at their bounds resulting in poor performance of FB approach. Note that only the binary variable is at a bound (i.e., y = 1) at the global optimum. SR of DETL is comparable to that of DE and MDE with RG approach, and is better than that of both

DE and MDE using FB approach for this example. The latter could be because of the tabu list in DETL, which avoided many function evaluations at the local solution thus increasing the probability of locating the global solution (SR is 71%, 32% and 39% for DETL, MDE and DE). On the other hand, NFE for DETL is less than that for both DE and MDE using both approaches (RG and FB) for this example.

For example 4, SR with FB approach is found to be better compared to RG for all the three methods. For example 5, SR of both DE and DETL using RG approach is better compared to FB approach, whereas SR of MDE is better with FB approach compared to RG approach. Besides many local minima reported in Ryoo and Sahinidis (1995) for example 5, a new local minimum at x = {0.2, 0.8, 1.5} and y = {0, 1, 1, 1} with f = 5.636852 is also observed in this study. NFE for DE and MDE are same using both the approaches (RG and FB) for this example. This is because the termination criterion: Gen_{max} is satisfied before the Sc_{max} criterion. On the other hand, DETL has high SR compared to MDE and comparable to that of DE, and its NFE is less than that for both DE and MDE for example 5. For example 6, SR of DE, MDE and DETL is 100% using both RG and FB approaches. This could be due to the presence of many global minima and locating one of them is sufficient to achieve the best solution. NFE of all the methods is reduced using FB approach compared to RG approach for this example because three variables are at their bounds at global solution which makes FB approach to converge faster compared to RG.

For example 7, SR is zero with RG approach, and it is 60-75% with FB approach for all the methods. For example 8, SR is zero for both RG and FB approaches with all the methods. These results show that examples 7 and 8 are

complex due to the associated large number of variables, constraints and local minima (Salcedo, 1992) compared to examples 1 to 6. In order to improve the performance of the algorithms, their parameters are re-tuned for examples 7 and 8, and the good parameters values obtained are: A = 0.9, CR = 0.6, NP = 100, Gen_{max} = 300N, Sc_{max} = 20N, tr = $N\times10^{-2}$ and tls = 100. The results with the re-tuned parameter values for examples 7 and 8 are also given Table 5.5. SR with FB approach is improved and is close to 100% whereas it is still at zero with RG approach for all the methods for these two examples. This is because many decision variables (5 for both examples 7 and 8) are at their bounds at the global solution, and handling boundary violations by FB approach (i.e., setting to the nearest boundary value) resulted in better performance. In addition to the high SR, NFE of DETL is fewer compared to both DE and MDE for examples 7 and 8.

These results show that FB approach is comparable to or better than RG approach except for example 3. In addition to RG and FB approaches, a mixed approach is also tested for all MINLPs in this study and the results are given in Table 5.5. With the mixed approach, as expected, SR and NFE of DE, MDE and DETL are generally between those obtained using RG and FB approaches.

In general, SR of DETL is comparable to that of DE and MDE, and its NFE is less than that of both DE and MDE. NFE for MDE is comparable to that for DE for most of the MINLPs with an average reduction of 3% compared to DE (Table 5.6). The percentage reduction in NFE for DETL compared to DE and MDE using RG, FB and Mixed approaches is given in Table 5.6. On average, DETL requires 41% and

Table 5.6: Percentage reduction in NFE for MINLPs

Example Number	Approach	MDE Compared to DE	DETL Compared to DE	DETL Compared to MDE
1	RG	-6.69	54.01	56.89
	FB	-1.43	54.45	55.09
	Mixed	-5.93	54.67	57.21
2	RG	1.05	46.05	45.48
	FB	1.16	68.09	67.71
	Mixed	-2.91	50.52	51.92
3	RG	-1.15	42.37	43.02
	FB	3.33	37.78	35.63
	Mixed	-9.53	31.04	37.04
4	RG	-0.85	8.479	9.25
	FB	12.77	62.46	56.97
	Mixed	-0.5	27.75	28.11
5	RG	0	50.14	50.14
	FB	0	68.40	68.40
	Mixed	0	50.21	50.21
6	RG	0	-0.07	-0.075
	FB	0.57	17.62	17.15
	Mixed	5.96	-1.26	-7.68
7#	FB	1.07	81.58	81.38
	Mixed	51.13	52.54	2.87
8#	FB	-5.74	21.82	26.07
	Mixed	22.77	33.22	13.53
Average		2.96	41.45	38.47

38% fewer NFE compared to the original DE and the recent MDE, respectively. The results obtained for DE and MDE implemented in this study (Table 5.5) are different from those reported in Angira and Babu (2006). This could be due to several reasons stated under the NLPs section along with two additional reasons: (i) the results in this study are based on 100 trials compared to only 10 trials in Angira and Babu (2006), and ii) integers are handled by rounding the continuous variables to the nearest integers in this study compared to truncating in Angira and Babu (2006). This shows the need for thorough comparison of MDE and DE using practical stopping criterion, common set of parameters and more trials as in this study to bring out the relative merits of methods for applications where the global minimum is unknown *a priori*.

5.8 Summary

A method, namely, DETL, by combining the concepts of TS and DE is presented for finding the global minimum of NLP and MINLP problems in chemical engineering. DETL is tested for many NLP problems having different degrees of complexity. The results show that SR of DETL is high, and is comparable to that of DE and MDE in locating the global solutions with fewer NFE. The average reduction in NFE of DETL compared to both DE and MDE is about 34% for the NLPs studied. DETL is then applied to MINLPs which involve process design and synthesis problems. The results show that the SR of DETL is comparable to that of DE and MDE while its NFE, on average, is around 40% less than that of both DE and MDE. Overall, the performance of DETL is significantly better compared to both DE and MDE in terms of NFE for finding the global minimum of NLPs and MINLPs.

CHAPTER 6

A NEW TRANSFORMATION FOR STOCHASTIC GLOBAL OPTIMIZATION METHODS*

In this work, a new transformation of the given objective function is proposed in order to facilitate escaping from a local minimum and thus improve the reliability of stochastic global optimization methods. The main features of the proposed transformation are: it preserves the relative ordering of local and global minima as in terminal repeller and unconstrained subenergy tunneling (TRUST) algorithm (Barhen et al., 1997), it transforms the current local minimum to a global maximum so that the optimization method escapes from the current local minimum, and it is simple. The transformation involves two steps: transforming the objective function such that the transformed function value becomes zero where the original function value is greater than or equal to that of the current local minimum, and building a concave envelope such that the algorithm escapes from the local minimum. The proposed transformation is implemented with one of the popular evolutionary algorithms, differential evolution (DE), and is evaluated for several test functions. A benchmark problem similar to phase equilibrium calculations is also proposed and studied.

* This chapter is based on the paper – Mekapati Srinivas and Rangaiah, G. P. An effective transformation for enhancing the stochastic global optimization methods, Presented at AIChE Annual Meeting, Nov. 2006, San Francisco, USA.

6.1 Introduction

Smoothening of the given objective function and tracing the global optimum is of interest, particularly in protein structure prediction. In general, protein folding problems are addressed computationally by minimizing the potential energy over all possible protein structures, and the structure with the lowest potential energy is presumed to be the most stable protein structure. The objective function (potential energy) of the protein folding problem is usually complex since the number of local minima increases exponentially with the problem dimension. There are several smoothing techniques proposed especially for protein structure prediction which include diffusion equation method (Kostrowicki and Piela, 1991), packet annealing method (Shalloway, 1992) and effective energy simulated annealing method (Coleman, 1993). These methods apply the transformation to the probability distributions instead of applying directly to the given objective function. Wu (1996) proposed a method that transforms the given non-linear objective function into a class of gradually deformed but smoother or easy functions, and it can be applied directly to the given objective function. In practice, all the above smoothening techniques can be used only if the integral of the given objective function exists otherwise a modified function is considered instead.

Barhen et al. (1997) proposed a method, namely, TRUST which involves the sub-energy transformation of the given objective function followed by repelling mechanism to escape from the local minimum. This transformation modifies the given function in such a way that it becomes flat where the objective function value is greater than the current local minimum while preserving the better local minima

(Figure 2.1). The results of TRUST for several test functions showed that it is fast (Barhen et al., 1997) compared to other methods in the literature. Our numerical experience with TRUST for some of the test functions shows that choosing parameters in the algorithm is critical in achieving the global solution efficiently.

Parsopoulos et al. (2001) proposed a two-step transformation for the objective function, known as stretching technique, to avoid convergence of any stochastic algorithm to the local minima. The transformation is applied along with the particle swarm optimization (PSO) for two test functions and an application problem involving training of neural networks. The results show that the reliability of PSO is improved with the transformation. Toh (2002) proposed a monotonic transformation of the given objective function which magnifies the relative ordering of global and local minima. Based on this transformation, regions that contain the global solutions are identified using a level set and then a global descent algorithm is applied within this level set for global optimal solutions. Numerical experiments with several test functions and application to neural network problems show that the proposed transformation provides good convergence for simple penalty-based constrained search algorithms (Toh, 2002).

Parsopoulos and Vrahatis (2004) proposed a transformation-based approach to locate all the global solutions of an objective function using PSO. Their approach uses both the deflection and stretching techniques in Parsopoulos et al. (2001) along with an additional concept, namely, repulsion technique to avoid difficulties arising from the artificial local minima (i.e., the minima not present in the original objective function, but are created by the transformation) induced by the former techniques.

Experimental results with a few problems in science and economic theory show that the proposed approach is effective in locating more than one global solution (Parsopoulos and Vrahatis, 2004).

In this study, a new transformation for the objective function is proposed to enhance the reliability of the stochastic global optimization methods. It facilitates any stochastic method to escape from the local minimum by converting it into a global maximum. The proposed transformation is tested for several test functions involving 2 to 10 variables and a few to hundreds of local minima. In addition, a benchmark problem similar to phase equilibrium problems is also proposed and studied.

6.2 Proposed Transformation

In order to improve the reliability of stochastic global optimization methods, a transformation of the given objective function with the following features is proposed in this study.

i) Simple with minimal number of parameters

ii) Preserves the relative ordering of local and global minima as in TRUST algorithm (Barhen et al., 1997), and

iii) Transforms the current local minimum into a global maximum and creates a concave envelope which helps a stochastic optimization algorithm to escape from a local minimum.

Let us consider that a stochastic global optimization method converged to a local minimum, x^*, of the given objective function which has still better minima (i.e.,

minima where the objective function value is better than that at x^*). In order to escape from the current local minimum (x^*), the proposed transformation is:

$$f_{trans} = \log\left(\frac{1}{1+\exp(-(f(x)-f(x^*)))}\right) + \frac{(1+\mathrm{sgn}(f(x)-f(x^*)))}{c\left(1+\sum_{i=1}^{N}\frac{\left\|x_i - x_i^*\right\|}{\left\|x_i^u - x_i^l\right\|}\right)} \qquad 6.1$$

where c is the single parameter of the transformation and "sgn" is the signum function (equal to -1, 0 and +1 for negative, zero and positive argument value respectively). The first part in the right hand side (RHS) of equation 6.1 transforms the given objective function such that function values greater than the current local minimum (f(x*)) become equal to zero while function values less than or equal to $f(x^*)$ are preserved. The second part in the RHS of equation 6.1 converts the local minimum to a global maximum so that the method escapes from the local minimum.

The effect of the proposed transformation can be illustrated using the test function: modified Himmelblau function (mHB, equation 2.15). As shown in Figure 6.1a, mHB is highly non-linear and non-convex with 4 local minima (LM1, LM2, LM3 and GM) respectively at x = (-3.763429, -3.266052), (-2.787063, 3.128204), (3.581492, -1.820800) and (3, 2) with function values 7.367345, 3.487127, 1.504353 and 0. Consider a stochastic global optimization method reached LM1 and the proposed transformation is applied with respect to LM1. The effect of first part in the RHS of equation 6.1 is to make the transformed function flat with zero value where the original objective function values are greater than that (7.367345) at LM1 (Figure 6.1b). The effect of the whole transformation (including both the terms in the RHS of equation 6.1) is shown in Figure 6.1c, which includes the concave envelope developed due to the second term in the RHS of equation 6.1. The local minimum

(LM1) has become the global maximum for the given function which makes the stochastic global optimization method to escape from LM1 and thus increases the probability of locating better minima (LM2, LM3 or GM).

The proposed transformation has two advantages over that in the literature (Parsopoulos et al., 2001): (i) only one parameter compared to 3 parameters in the transformation of Parsopoulos et al. (2001), and (ii) it does not create any new local minima. Though the first part in the RHS of equation 6.1 is similar to that (equation 2.3) in TRUST algorithm, it does not require gradient of the function and the proposed transformation does not create new local minima (Figure 2.1).

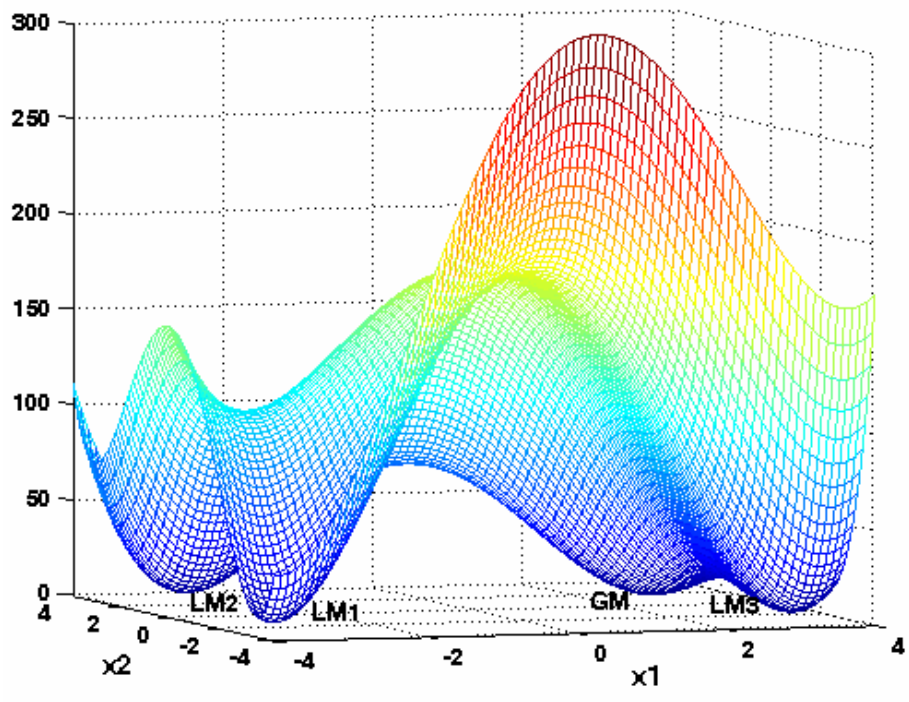


Figure 6.1a: Modified Himmelblau function

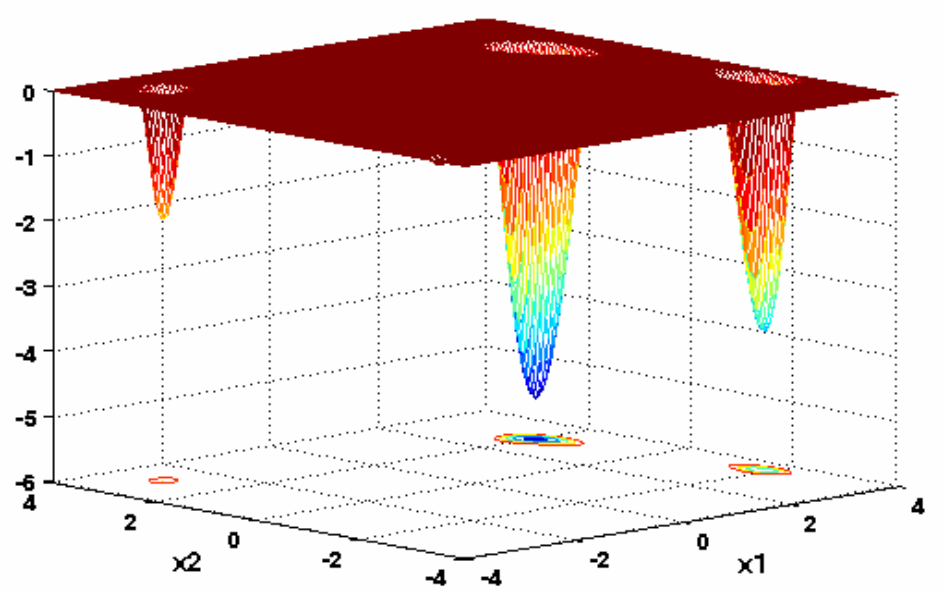


Figure 6.1b: Effect of 1st part in RHS of equation 6.1 for mHB function

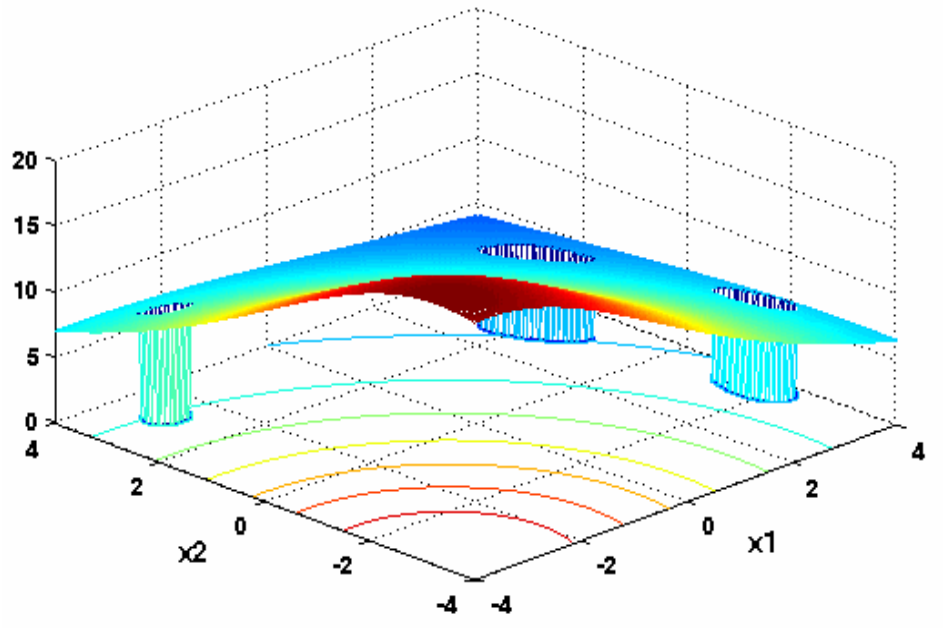


Figure 6.1c: Effect of the proposed transformation (equation 6.1) for mHB function

6.3 Implementation and Evaluation

In this study, the proposed transformation is applied with one of the evolutionary algorithms, namely, differential evolution (DE); however, it is equally applicable with any stochastic method such as genetic algorithms (GA), simulated annealing (SA), and tabu search (TS). To facilitate easy implementation, the proposed transformation is used at the end of DE i.e., after the local minimization step of DE (Figure 3.1). This is because DE alone may not provide accurate solution and the use of transformation after DE (without local optimization) may result in exploring the better regions in the same valley (i.e., a better function value but in the same optimum region). However, it can be implemented even at different stages (i.e., after a specified number of generations along with a local optimization technique) of DE. For local optimization, a fast convergent quasi-Newton method is used here. The transformation in the literature (Parsopoulos et al., 2001) is also implemented with DE, and the results obtained are compared with that of the proposed transformation.

The main steps in the implementation of DE with the proposed/literature transformation are:

Step 1: Apply DE with local optimization as shown in Figure 3.1.

Step 2: Check the solution obtained; if the global minimum is not obtained, perform the proposed/literature transformation, else go to step 5.

Step 3: Re-initialize certain percentage of the population in DE.

Step 4: Apply DE with local optimization (Figure 3.1) to the transformed objective function.

Step 5: Check the solution obtained; if the global minimum is obtained, declare the trial as successful, else declare it as failure.

From the above steps, it is clear that *a priori* information about the global optimum is required to apply the proposed/literature transformation. Hence, in this study, several test functions with known global minimum are chosen for evaluating the performance of the proposed and literature transformations, and this approach is similar to that of Parsopoulos et al. (2001).

Although the transformation is proposed mainly to improve the reliability, its performance is evaluated both in terms of improvement in the reliability (measured via success rate (SR) i.e., number of times DE located the global optimum out of 100 trials), and computational efficiency (measured via number of function evaluations (NFE) required to locate the global optimum) of DE. The termination criterion used for DE is the maximum number of generations i.e., Gen_{max}. Note that the proposed transformation is implemented only for unsuccessful trials (i.e., where DE converged to a local minimum) of DE out of 100.

6.4 A Benchmark Problem Similar to Phase Equilibrium Calculations

Phase equilibrium problems have a characteristic of a few but comparable minima i.e., the function values at local and global minima are close to each other. In addition, the local minimum in these problems is some times in a narrow valley. Although there are many benchmark problems available in the literature with different characteristics (such as flat objective function and huge number of local minima), none of them represents the comparable minima as in phase equilibrium problems.

Motivated from the unique characteristic of phase equilibrium problems, a benchmark problem, namely, modified N-dimensional test (mNDT) function is developed in the earlier study (Chapter 2). Though mNDT function has comparable minima, they are huge in number (i.e. increases exponentially with dimension of the problem) in contrast to a few minima in phase equilibrium problems. Hence, in this chapter, a benchmark problem with a few but comparable minima is developed to represent a corresponding test function for phase equilibrium problems.

The new benchmark problem is developed from the Rosenbrock function since it has a few minima and also narrow global minimum region as in phase equilibrium problems. The minima in Rosenbrock function are made comparable by adding a multiplier (α/N) to the quadratic term:

$$f(x)=\sum_{i=1}^{N}\left[100(x_i^2-x_{i+1})^2+\frac{\alpha}{N}(x_i-1)^2\right] \qquad 6.2$$

where 'α' is a constant and 'N' is the dimension of the problem. Both Rosenbrock function and the modified Rosenbrock (mROS) function do not have local minima up to and including 3 variables but they have several local minima (many of them are constrained minima) beginning from 4 variables. The minima in the mROS function are made comparable to the global minimum by decreasing the effect of quadratic term via α, but the global minimum is unaffected (i.e., 0 at $x_i = 1$ for i = 1, 2, …, N). This can be seen in Table 6.1 for the 5 variables case; as α/N decreases, the objective function becomes slightly flat and the minima become comparable but the global minimum is the same. The comparable minima for mROS function are given in Table 6.2 for $\alpha = 1.5\times10^{-3}$; the difference in function value between the nearest local and the global minimum is in the range 1.39×10^{-3} to 3×10^{-4} for 4 to 20 variables, and is within the range of phase equilibrium (2.28×10^{-2} to 5.7×10^{-5}) problems studied earlier.

Table 6.1: Trend of comparable minima with α for the modified Rosenbrock function (5 variables case)

α	1	2.5×10^{-2}	1.5×10^{-2}	5×10^{-3}	1.5×10^{-3}
Local minimum	3.930839	1.972×10^{-2}	1.183×10^{-2}	3.944×10^{-3}	1.183×10^{-3}
Global minimum	0.0	0.0	0.0	0.0	0.0

Table 6.2: Function values at the comparable minimum for the modified Rosenbrock function with $\alpha = 1.50\times10^{-3}$

Number of variables (N)	Function value at the comparable minimum
4	1.394×10^{-3}
5	1.183×10^{-3}
6	9.968×10^{-4}
8	7.498×10^{-4}
10	5.999×10^{-4}
12	4.999×10^{-4}
14	4.2857×10^{-4}
16	3.75×10^{-4}
18	3.333×10^{-4}
20	3×10^{-4}

Note: The global minimum is 0.0 at $x_i = 1$ for i = 1, 2, …, N.

6.5 Application to Benchmark Problems

Test functions, namely, Rastrigin (RA_2), Easom (ES), mHB, mNDT (Chapter 2) along with mROS function are chosen for studying the proposed transformation since these functions are found to have low SR with DE in our previous studies (Chapter 4).

6.5.1 Results and Discussion

The parameter values used for DE are same as the optimum values obtained earlier (Chapter 4) (i.e., CR = 0.2, A = 0.5, NP = 20 and Gen_{max} = 60N). The proposed transformation has one parameter, c, and is tuned using ES, mHB and RA_2 functions. The results are given in Table 6.3. SR and NFE for DE without implementing the transformation are 89% and 1986, 88% and 1338, and 74% and 2351 respectively for ES, mHB and RA_2 functions. The results in Table 6.3 show that the improvement in SR of DE with transformation is slightly affected by c for both ES and mHB functions whereas it is same for RA_2 function. The NFE for DE with the transformation is nearly unaffected for both ES and mHB functions, and it is same for RA_2 for all values of c. This is because SR of RA_2 is same for all values of c, and the NFE is calculated based on only the successful trials out of 100. Overall, the performance of the proposed transformation is only slightly affected by the parameter, c, and a value of 0.01 is chosen for further study.

Table 6.3: Effect of the parameter, c, on the performance of DE

Parameter (c) value	Proposed transformation					
	ES		mHB		RA_2	
	SR	NFE	SR	NFE	SR	NFE
0.1	96	1969	91	1423	82	2447
0.01	96	1974	97	1372	82	2447
0.001	94	1978	93	1346	82	2447
0.0001	96	1952	93	1435	82	2447

In order to see the effect of re-initialization (in step 3 of the algorithm on page 177), the proposed transformation and the transformation of Parsopoulos et al. (2001) are studied with 25%, 50% and 100% of the whole population in DE being re-

initialized for ES, mHB and RA_2 functions, and the results are given in Tables 6.4a and b. For the transformation of Parsopoulos et al. (2001), the reported values ($\gamma_1 = 10000, \gamma_2 = 1$ and $\mu = 10^{-10}$) are employed here. SR of DE without the transformation is respectively 89%, 88% and 74% for ES, mHB and RA_2 functions. The improvement in SR of DE for both ES and RA_2 functions increases (from 6 to 9% for ES, and 0 to 20% for RA_2) with the proposed transformation, and is nearly constant (at 2% and 1% respectively for ES and RA_2) for the transformation of Parsopoulos et al. (2001), with decrease in the percentage of the population re-initialized. The improvement in SR of DE for mHB function decreases for both the transformations with decrease in percentage of the population re-initialized. This could be because of the distribution and the number of local minima in the respective functions. ES is a flat function with the local minimum located near to the center, mHB has narrow global minimum region and RA_2 has huge number of local minima with the global minimum at center.

Table 6.4a: Effect of re-initialization on SR of DE

Percentage of the population re-initialized	With the proposed transformation			With the transformation in the literature (Parsopoulos et al., 2001)		
	ES	mHB	RA_2	ES	mHB	RA_2
100	95	98	74	91	92	75
50	95	97	80	91	87	74
25	98	92	94	91	88	75

NFE for DE without transformation is 1986, 1338 and 2351 for ES, mHB and RA_2 functions respectively. The results in Table 6.4b show that NFE for DE is slightly affected by the decrease in the percentage of population re-initialized using the proposed transformation and is nearly constant using the transformation of

Parsopoulos et al. (2001). It is also observed that NFE for DE (Table 6.4b) with the proposed transformation is slightly higher compared to that with the transformation of Parsopoulos et al. (2001). Based on the performance (SR and NFE in Tables 6.4a and b) results obtained, both the transformations are studied with 25% of the population in DE being re-initialized for the benchmark problems.

Table 6.4b: Effect of re-initialization on NFE for DE

Percentage of the population re-initialized	With the proposed transformation			With the transformation in the literature (Parsopoulos et al., 2001)		
	ES	mHB	RA_2	ES	mHB	RA_2
100	1955	1512	2350	1980	1319	2309
50	1978	1372	2447	1997	1323	2351
25	1979	1399	2591	1970	1353	2366

The performance results (SR and NFE) of DE with and without transformation for all the test functions studied are summarized in Table 6.5. NFE given is the average over successful trials out of 100 trials. SR of DE is improved by around 8% for both ES and mHB functions with the proposed transformation, compared to 2-4% with the transformation of Parsopoulos et al. (2001). SR of DE is improved around 20% for RA_2 function with the proposed transformation and is very high compared to that (1%) with the transformation of Parsopoulos et al. (2001). This clearly shows that the proposed transformation has higher chance of escaping from the local minima compared to that of Parsopoulos et al. (2001).

SR of DE for the modified Rosenbrock and modified N-dimensional test functions are also given in Table 6.5. Functions, $mROS_2$ and $mROS_3$, are not used in this study since they don't have multiple minima. SR of DE is improved around 12%

for $mROS_4$, and the improvement decreases as the number of variables increases from 4 to 10 for both the transformations. This is because as the number of variables increases, though the transformation facilitates to escape from the local minimum, DE could not find the better regions where the function value is better than that at the current local minimum.

For the $mNDT_2$ function, SR of DE is improved around 7% using both the transformations, and there is practically no improvement from 4 variables onwards. The improvement in the SR of DE for mNDT functions using both the transformations is less compared to that of mROS functions. This is because the number of local minima increases exponentially with the dimension of the problem in mNDT functions compared to only several local minima in mROS functions. It is also observed that, the improvement in SR of DE using both the transformations is almost comparable.

NFE for DE with and without transformation for both the proposed transformation and the transformation of Parsopoulos et al. (2001) are also given in Table 6.5. The NFE with either transformation is more than that without transformation for almost for all functions tested except for ES and $mNDT_2$. This is because DE is applied twice for the failure cases whereas it is applied only once for successful trials with DE applied to the original objective function (see the algorithm on page 177). The NFE with the proposed transformation is slightly high compared to that with the transformation of Parsopoulos et al. (2001), for most of the functions. This could be because the concave envelope created (Figure 6.1c) by the proposed transformation has a smaller gradient requiring more NFE for DE to escape from the

current local minimum. It is also observed that the average NFE with both the transformations increases with the number of variables.

Table 6.5: SR and NFE of DE with and without transformations

Function	SR			NFE		
	WOT	WT		WOT	WT	
		Proposed	Literature*		Proposed	Literature*
ES	89	97	91	1986	1979	1970
mHB	88	96	92	1338	1399	1353
RA_2	74	94	75	2351	2591	2366
$mROS_4$	79	91	91	4951	5379	5380
$mROS_5$	79	88	89	7348	7763	7642
$mROS_6$	85	93	92	9940	10,664	10,291
$mROS_7$	89	94	93	12,998	13,464	13,184
$mROS_8$	90	94	93	16,259	16,823	16,636
$mROS_9$	90	94	92	19,957	20,582	20,280
$mROS_{10}$	91	94	92	24,242	24,888	24,235
$mNDT_2$	58	65	65	2070	2043	2015
$mNDT_3$	70	71	71	3561	3564	3577
$mNDT_4$	59	59	60	4822	4822	4846
$mNDT_5$	55	55	55	6027	6027	6027
$mNDT_6$	39	39	39	7227	7227	7227
$mNDT_7$	33	33	33	8428	8428	8428
$mNDT_8$	24	24	24	9629	9629	9629
$mNDT_9$	19	19	19	10,830	10,830	10,830
$mNDT_{10}$	14	14	14	12,031	12,031	12,031

Note: WOT – Without Transformation; WT – With Transformation; ES – Easom; mHB – modified Himmelblau; RA_2 – Rastrigin 2 variables; $mROS_2$ to $mROS_{10}$ – modified Rosenbrock function with 2 to 10 variables; and $mNDT_2$ to $mNDT_{10}$ – modified N-dimensional test function with 2 to 10 variables. * - Results obtained using the transformation of Parsopoulos et al. (2001).

6.6 Summary

A transformation for the objective function to enhance the reliability of stochastic global optimization methods and a benchmark problem similar to phase equilibrium calculation, are proposed. The proposed transformation is studied with one of the evolutionary algorithms, DE, over several test functions involving up to 10 variables. The results show that the proposed transformation is more reliable and require slightly more NFE compared to the transformation of Parsopoulos et al. (2001). It is also found that the performance of the proposed transformation is less sensitive to the only parameter it has. The performance of both the transformations decreases as the number of variables in the objective function increases, and so further study is needed to enhance them.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

In this thesis, several recent stochastic global optimization methods have been carefully implemented and evaluated comprehensively for chemical engineering applications. In addition, a method, namely, differential evolution with tabu list (DETL) and two benchmark problems are proposed and studied. The major contributions and conclusions are summarized below.

1. The recent random tunneling algorithm (RTA) has been implemented and its potential is evaluated for benchmark problems involving 2 to 20 variables and challenging chemical engineering applications such as phase equilibrium calculations using Gibbs free energy minimization and parameter estimation in models. The results show that the reliability of RTA is comparable to or less than that of tabu search (TS) and genetic algorithms (GA), and its computational efficiency is lower than that of TS but is better than that of GA.
2. Subsequently, two promising stochastic algorithms, differential evolution (DE) and TS have been implemented with a local optimization technique at the end, and evaluated for benchmark, phase equilibrium calculations and phase stability problems. The neighbors in TS are generated in three different ways: using hyper-rectangles, mixed and random way of generation to study their effectiveness. Among the three ways for neighbors generation, TS using hyper-rectangles is found to be better for benchmark and phase equilibrium calculations, whereas TS

using mixed way is found to be better for phase stability problems. Overall, the reliability of DE is found to be better than that of TS whereas the latter has better computational efficiency compared to the former.

3. DETL is proposed by effectively integrating the strong features of DE and TS. It is then thoroughly tested over a wide range of test functions, phase equilibrium calculations and parameter estimation problems in differential and algebraic systems. DETL successfully located the global minimum of all these problems with high reliability and around 30% less number of function evaluations (NFE) compared to both DE and the modified differential evolution (MDE) (Babu and Angira, 2006). Overall, the performance of DETL is better than that of stand alone DE and TS algorithms.
4. DETL is then evaluated for non-linear programming problems (NLPs) and mixed-integer non-linear programming problems (MINLPs) with constraints, often encountered in chemical engineering practice. The results show that the reliability of DETL is high with around 35% and 51% less NFE compared to both DE and MDE respectively for NLPs and MINLPs tested. Overall, the performance of DETL is significantly better than that of DE and MDE for NLPs and MINLPs.
5. A transformation for the objective function to enhance the reliability of stochastic global optimization methods is proposed and studied with DE for several test functions. The reliability of DE is improved with the proposed transformation, and is better compared to the transformation in literature (Parsopoulos et al., 2001).
6. Two benchmark problems having comparable minima as in phase equilibrium calculations and phase stability problems have been proposed and studied. One of them, modified N-dimensional test function (mNDT) is more difficult with

numerous minima compared to the other, modified Rosenbrock function (mROS) with a few minima.

7.2 Recommendations for Future Works

Stochastic global optimization and its applications have wide scope for many studies in chemical engineering; some possible future works are outlined below.

DETL for Multi-Objective Optimization: Most of the times, the real-world applications involve more than a single objective and it is often very hard to formulate these problems into a single objective optimization problem due to the conflicting objectives. In general, these problems are now solved by considering all the objectives separately, and are known as multi-objective optimization (MOO) or multi-criteria optimization or vector optimization problems. MOO has numerous applications in chemical engineering (Bhaskar et al., 2000; and Tarafder et al., 2005) which include the MOO of industrial hydrogen plants (Rajesh et al., 2001), styrene reactor (Yee et al., 2003; and Babu et al., 2005), hydrocracker (Bhutani et al., 2006), fed-batch reactors (Sarkar and Modak, 2005) etc. The multiple conflicting objectives in MOO problems lead to a set of optimal solutions called the Pareto-optimal solutions, and none of them can be said to be better than another without additional information about the problem. These solutions provide more flexibility and 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 years, several algorithms have been developed to address MOO

problems (e.g., Miettinen, 1999; and Deb, 2001); these include the well known evolutionary algorithms.

In the present study, DETL has been developed and tested for several single objective optimization problems, and its performance is found to be better than DE, MDE and TS. Though DETL has been used for single objective, it has the capability of solving MOO problems too as it can generate a set of optimal solutions along with ability to handle non-linear and non-convex objective functions. Therefore, the study of DETL for MOO problems is one of the interesting and recommended works.

Auto-tuning of DETL parameters: Tuning of parameters is a crucial step and is essential while evaluating the performance of any stochastic global optimization method. For example, in our studies, the parameters of the algorithms (RTA, DE, TS and DETL) are tuned for each set of functions (i.e., benchmark, phase equilibrium calculations and parameter estimation in models etc.,) by considering a few difficult functions in the respective groups. However, one can improve the performance of each of these methods by fine tuning the parameters for each problem as in Angira and Babu (2006) and Cai and Shao (2002).

Fine tuning the parameters of a given algorithm for an application requires additional effort by the user not only in terms of exploring several parameter values but also some understanding of the algorithm and its parameters. Hence, working towards auto-tuning (i.e., the algorithm performs tuning on its own) of parameters in an optimization method plays a major role in achieving better performance of the method while encouraging more practitioners to optimize their processes. Several

studies have addressed this issue, particularly for evolutionary algorithms (Hinterding, 1995; Tan et al., 2001; Shi et al., 2003; Liu and Lampinen, 2005; and Teo, 2006). The two important approaches are: adaptation and self-adaptation. Adaptation involves changing values of parameters based on some feedback from the search whereas self-adaptation treats the parameters of the algorithm as additional decision variables in the optimization problem. Although these approaches increase the complexity of the algorithm/problem, results from the above studies show that the convergence property of evolutionary algorithms can be improved with auto-tuning of the parameters along with easy implementation for the end-user. Therefore, auto-tuning of parameters in DETL is recommended as one of the potential future works.

DETL for Dynamic Optimization: Dynamic optimization is becoming particularly important in chemical engineering and the applications (Balsa-Canto et al., 2005; and Papamichail and Adjiman, 2004) include (i) estimating kinetic parameters of reactions which is essential for the design and control of chemical and bio systems, (ii) optimal control of batch and semi-batch reactors, and (iii) start-up and shut-down of processes.

Numerous local solutions exist even for rather simple dynamic optimization problems (Luus and Cormack, 1972), and hence the use of traditional gradient-based algorithms has the possibility of getting trapped into local optimum. Consequently, there has been growing interest in developing global optimization algorithms for these problems (Horst and Pardalos, 1995). Several deterministic algorithms such as αBB method (Esposito and Floudas, 2000) and spatial branch and bound method (Papamichail and Adjiman, 2004) have been considered for dynamic optimization.

Though these methods provide mathematical guarantee to the global optimum, they need extensive and exhaustive search requiring more computational resources. On the other hand, several stochastic global optimization algorithms (Carrasco and Banga, 1997; Luus and Hennessy, 1999; Luus, 2000; Hanke and Li, 2000; Sarkar and Modak, 2003; and Angira and Santosh, 2006) have been successfully applied to the dynamic optimization, which include iterative dynamic programming, integrated controlled random search method, direct search procedure of Luus and Jaakola (1973), simulated annealing and evolutionary algorithms such as GA and DE.

In this study, DETL is found to be computationally efficient compared to DE and TS for the dynamic optimization problems tested (Chapter 4). So a thorough study of DETL over a wide range of dynamic optimization problems is recommended for a future work.

Comprehensive Study of the Proposed Transformation: In this study, a transformation for the objective function has been proposed (Chapter 6) to enhance the reliability of the stochastic global optimization methods. The proposed transformation is tested with one of the stochastic method, DE, for several test functions. It was found that the success rate (SR) of DE is improved and is better compared to the transformation of Parsopoulos et al. (2001). Though the reliability of DE is improved for the benchmark problems tested, the performance of the proposed transformation needs to be evaluated/improved in several ways: (i) by implementing and studying the proposed transformation even in the intermediate stages (i.e., after every specified number of generations) in addition to the final stage (as implemented in this study) to explore a suitable combination for DE, (ii) modifying the proposed

transformation in order to improve SR of DE for problems with more variables, and (iii) studying the proposed transformation with other stochastic methods such as TS and DETL especially for the examples where SR of these methods is low.

REFERENCES

Adjiman, C. S. and Floudas, C. A. Rigorous convex under-estimators for general twice-differentiable problems, *Journal of Global Optimization*, 9, pp. 23-40. 1996.

Adjiman, C. S., Androulakis, I. P. and Floudas, C. A. Global optimization of MINLP problems in process synthesis and design, *Computers and Chemical Engineering*, 21, pp. S445-450. 1997.

Adjiman, C. S., Dallwig, S. Floudas, C. A. and Neumaier, A. A global optimization method, α BB, for genera twice-differentiable constrained NLPs – theoretical advances, *Computers and Chemical Engineering*, 22, pp. 1137-1158. 1998.

Adjiman, C. S., Androulakis, I. P. and Floudas, C. A. Global optimization of mixed-integer non-linear problems, *American Institute of Chemical Engineers Journal*, 46, pp. 1769-1797. 2000.

Aggarwal, A. and Floudas, C. A. Global optimum search for the pooling problem. In *Proceedings of TIMS/ORSA Meeting*, pp. 25-27. 1988. Washington, USA.

Angira, R. and Babu, B. V. Optimization of process synthesis and design problems: a modified differential approach, *Chemical Engineering Science*, 61, pp. 4707-4721. 2006.

Angira, R. and Santosh, A. Optimization of dynamic systems: a trigonometric differential evolution approach, in press, *Computers and Chemical Engineering*, July 2006. URL: http://www.sciencedirect.com/science/journal/00981354.

Arsham, H. Global optima for linearly constrained business decision models, *Scientific Journal of Administrative Development*, 2, pp. 27-53. 2004.

Babu, B. V. *Process plant simulation*, India: Oxford University Press. 2004.

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

Babu, B. V. and Angira, R. Modified differential evolution (MDE) for optimization of non-linear chemical processes, *Computers and Chemical Engineering*, 30, pp. 989-1002. 2006.

Baker, L. E., Pierce, A. C. and Luks, K. D. Gibbs energy analysis of phase equilibria, *Society of Petroleum Engineers Journal*, 22, pp. 731-742. 1982.

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 Equilibria*, 212, pp. 257-267. 2003.

Balsa-Canto, E., Vassiliadis, V. S. and Banga, J. R. Dynamic optimization of single and multi-stage systems using a hybrid stochastic-deterministic method, *Industrial Engineering and Chemistry Research,* 44, pp. 1514-1523. 2005.

Banga, J. R., Versyck, K. J. and Van Impe, J. F. Computation of optimal identification experiments for non-linear dynamic process models: a stochastic global optimization approach, *Industrial and Engineering Chemistry Research*, 41, pp. 2425-2430. 2002.

Banga, J. R., Moles, C. G. and Alonso, A. A. Global optimization of bioprocesses using stochastic and hybrid methods. In *Frontiers in Global Optimization*, ed by Floudas, C. A. and Pardalos, P. M., Boston: Kluwer Academic Publishers. 2003.

Bard, Y. *Nonlinear parameter estimation*. New York: Academic Press. 1974.

Bard, J. F. *Practical bilevel optimization. Nonconvex optimization and its applications*. Netherlands: Kluwer Academic Publishers. 1998.

Barhen, J., Protopopescu, V. and Reister, D. TRUST: A deterministic algorithm for global optimization, *Science*, 276, pp. 1094-1097. 1997.

Barron, C., Gomez, S. and Romero, D. Archimedean polyhedron structure yields a lower energy atomic cluster, *Applied Mathematics Letters*, 9, pp. 75-78. 1996.

Beigler, L. T. and Grossmann, I. E. Retrospective on optimization, *Computers and Chemical Engineering*, 28, pp. 1169-1192. 2004.

Bergey, P. K. and Ragsdale, C. Modified differential evolution: a greedy random strategy for genetic recombination, *Omega*, 33, pp. 255-265. 2005.

Bhaskar, V., Gupta, S. K. and Ray, A. K. Applications of multi-objective optimization in chemical engineering, *Reviews in Chemical Engineering*, 16, pp. 1-72. 2000.

Bhat, T. R., Venkataramani, D., Ravi, V. and Murty, C. V. S. An improved differential evolution method for efficient parameter estimation in bio-filter modeling, *Biochemical Engineering Journal*, 28, pp. 167-176. 2006.

Bhushan, S. and Karimi, I. A. Heuristic algorithms for scheduling an automated wet-etch station, *Computers and Chemical Engineering*, 28, pp. 363-379. 2004.

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

Bingul, Z. A new PID tuning technique using differential evolution for unstable and integrating processes with time delay. In *Proceedings of International Conference on Neural Information Processing, Lecture Notes in Computer Science*, 3316, pp. 254-260. 2004.

Bjork, K. J. and Westerlund, T. Global optimization of heat exchanger network synthesis problems with and without isothermal mixing assumption, *Computers and Chemical Engineering*, 26, pp. 1581-1593. 2002.

Boender, C. G. and Romeijn, H. E. Stochastic methods. In *Handbook of Global optimization*, ed by Horst, R. and Pardalos, P.M., pp. 829-869. Netherlands: Kluwer Academic Publishers. 1995.

Bullard, L. G. and Biegler, L. T. Iterated linear programming strategies for non-smooth simulation: A penalty based method for vapor-liquid equilibrium applications, *Computers and Chemical Engineering*, 17, pp. 95-109. 1993.

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

Cai, W. and Shao, X. A fast annealing evolutionary algorithm for global optimization, *Journal of Computational Chemistry*, 23, pp. 427-435. 2002.

Cardoso, M. F., Salcedo, R. L., Feyo de Azevedo, S. and Barbosa, D. A simulated annealing approach to the solution of MINLP problems, *Computers and Chemical Engineering*, 21, pp. 1349-1364. 1997.

Carrasco, E. F. and Banga, J. R. Dynamic optimization of batch reactors using adaptive stochastic algorithms, *Industrial and Engineering Chemistry Research*, 36, pp. 2252-2261. 1997.

Castillo, J. and Grossmann, I. E. Computation of phase and chemical Equilibria, *Computers and Chemical Engineering*, 5, pp. 99-108. 1981.

Cetin, B. C., Barhen, J. and Burdick, W. J. Terminal repeller unconstrained subenergy tunneling for fast global optimization, *Journal of Optimization Theory and Applications,* 77, pp. 97-126. 1993.

Chelouah, R. and Siarry, P. Tabu search applied to global optimization, *European Journal of Operational Research*, 123, pp. 256-270. 2000.

Chiou, J. P. and Wang, F. S. Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process, *Computers and Chemical Engineering*, 23, pp. 1277-1291. 1999.

Chiou, J. P., Chang, C. F. and Su, C. T. Ant direction hybrid differential evolution for solving large capacitor placement problems, *IEEE Transactions on Power Systems*, 19, pp. 1794-1800. 2004.

Coleman, T. F., Shalloway, D. and Wu, Z. Isotropic effective energy simulated annealing searches for low energy molecular cluster states, *Computational Optimization and Applications*, 2, pp. 145-170. 1993.

Costa, L. and Oliveira, P. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems, *Computers and Chemical Engineering*, 25, pp. 257-266. 2001.

Danish, M., Kumar, S., Qamareen, A. and Kumar, S. Optimal solution of MINLP problems using modified genetic algorithm, *Chemical Product and Process Modeling*, 1, pp. 1-43. 2006.

Das, H., Cummings, P. T. and Levan, M. D. Scheduling of serial multi-product batch processes via simulated annealing, *Computers and Chemical Engineering*, 14, pp. 1351-1362. 1990.

Deb, K. An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186, pp. 311-318. 2000.

Deb, K. *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley and Sons. 2001.

Deb, K. *Optimization for engineering design*. New Delhi: Prentice-Hall. 2002.

Diener, I. Trajectory methods in global optimization. In *Handbook of Global optimization*, ed by Horst, R. and Pardalos, P.M., pp. 649-668. Netherlands: Kluwer Academic Publishers. 1995.

Diller, D. J. and Verlinde, C. L. M. J. A critical evaluation of several global optimization algorithms for the purpose of molecular docking, *Journal of Computational Chemistry*, 20, pp. 1740-1751. 1999.

Dixon, L. C. W., and Szego, G. P. *Towards Global Optimisation*, Volume 1. Amsterdam: North-Holland. 1975.

Dixon, L. C. W., and Szego, G. P. *Towards Global Optimisation*, Volume 2. Amsterdam: North-Holland. 1978.

Dorigo, M., Maniezzo, V. and Colorni, A. The ant system: optimization by a colony of co-operating agents, *IEEE Transactions on System, Man and Cybernetics Part B*, 26, pp. 29-41. 1996.

Duran, M. A. and Grossmann, I. E. An outer approximation algorithm for a class of mixed-integer non-linear programs, *Mathematical Programming*, 36, pp. 307-339. 1986.

Edgar, T. F., Himmelblau, D. M. and Lasdon, L. S. *Optimization of chemical processes*. Singapore: McGraw-Hill. 2001.

Esposito, W. R. and Floudas, C. A. Global optimization in parameter estimation of nonlinear algebraic models via the error-in-variables approach, *Industrial and Engineering Chemistry Research*, 37, pp. 1841-1858. 1998.

Esposito, W. R. and Floudas, C. A. Global optimization for the parameter estimation of differential-algebraic systems, *Industrial and Engineering Chemistry Research*, 39, pp. 1291-1310. 2000.

Faber, R., Jochenhover, T. and Tsatsaronis, G. Dynamic optimization with simulated annealing, *Computers and Chemical Engineering*, 29, pp. 273-290. 2005.

Faina, L. A global optimization algorithm for the three dimensional packing problem, *European Journal of Operational Research*, 126, pp. 340-354. 2000.

Floudas, C. A., Ciric, A. R., and Grossmann, I. E. Automatic synthesis of heat exchanger networks, *American Institute of Chemical Engineers Journal*, 32, pp. 276-290. 1986.

Floudas, C. A., Aggarwal, A. and Ciric, A. R. Global optimum search for non-convex NLP and MINLP problems, *Computers and Chemical Engineering*, 13, pp. 1117-1132. 1989.

Floudas, C. A. *Nonlinear and mixed-integer optimization – fundamentals and applications*. New York: Oxford University Press. 1995.

Floudas, C. A. Global optimization in design and control of chemical process systems, *Journal of Process Control*, 10, pp. 125-134. 2000a.

Floudas, C. A. *Deterministic global optimization: theory, methods and applications. Nonconvex optimization and its applications.* Netherlands: Kluwer Academic Publishers. 2000b.

Floudas, C. A. and Pardalos, P. M. *State of the art in global optimization*. Netherlands: Kluwer Academic Publishers. 1996.

Floudas, C. A. and Pardalos, P. M. *Optimization in computational chemistry and molecular biology. Nonconvex optimization and its applications*. Netherlands: Kluwer Academic Publishers. 2000.

Floudas, C. A., Akrotirianakis, I. G., Caratzoulas, S., Meyer, C. A. and Kallarath, J. Global optimization in the 21st century: advances and challenges, *Computers and Chemical Engineering*, 29, pp. 1185-1202. 2005.

Floquet, P., Pibouleau, and Domenech, S. Separation system synthesis: how to use a simulated annealing procedure, *Computers and Chemical Engineering*, 18, pp. 1141-1148. 1994.

Forster, W. Homotopy methods. In *Handbook of Global optimization*, ed by Horst, R. and Pardalos, P.M., pp. 669-750. Netherlands: Kluwer Academic Publishers. 1995.

Garrard, A. and Fraga, E. S. Mass exchange network synthesis using genetic algorithms, *Computers and Chemical Engineering*, 22, pp. 1837-1850. 1998.

Gau, C. Y. and Stadtherr, M. A. Reliable non-linear parameter estimation using interval analysis: error-in-variable approach, *Computers and Chemical Engineering*, 24, pp. 631-637. 2000.

Gautam, R., and Seider, W. D. Computation of phase and chemical equilibrium. Part II. Phase splitting, *American Institute of Chemical Engineers Journal*, 25, pp. 999-1006. 1979.

Glover, F. Tabu search: part 1. *ORSA Journal on Computing*, 1, pp. 190-206. 1989.

Glover, F. Tabu search: part 2. *ORSA Journal on Computing*, 2, pp. 4-32. 1990.

Glover, F. and Laguna, M. *Tabu Search*. Boston: Kluwer Academic Publishers. 1997.

Gomez, S., Castillo, N. D., Castellanos, L. and Solano, J. The parallel tunneling method, *Parallel Computing*, 29, pp. 523-533. 2003.

Grossmann, I. E. and Sargent, R. W. H. Optimum design of multi-purpose chemical plants, *Industrial and Engineering Chemistry Process Design and Development*, 18, pp. 343. 1979.

Grossmann, I. E. *Global optimization in engineering design*. Netherlands: Kluwer Academic Publishers. 1996.

Grossmann, I. E. and Beigler, L. T. Future perspective on optimization, *Computers and Chemical Engineering*, 28, pp. 1193-1218. 2004.

Groenwold, A. A. and Snyman, J. A. Global optimization using dynamic search trajectories, *Journal of Global Optimization*, 24, pp. 51-60. 2002.

Hanke, M. and Li, P. Simulated annealing for the optimization of batch distillation processes, *Computers and Chemical Engineering*, 24, pp. 1-8. 2000.

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

Hansen, P. and Jaumard, B. Lipschitz optimization. In *Handbook of Global Optimization*, ed by Horst, R. and Pardalos, P. M., pp. 407-493. Netherlands: Kluwer Academic Publishers. 1995.

Harding, S. T. and Floudas, C. A. Phase stability with cubic equations of state: a global optimization approach, *American Institute of Chemical Engineers Journal*, 46, pp. 1422-1440. 2000.

Harik, G. R., Lobo, F. G. And Goldberg, D. E. The compact genetic algorithm, *IEEE Transactions on Evolutionary Computation*, 3, pp. 287-297. 1999.

Haupt, R. L. and Haupt, S. E. *Practical genetic algorithms*. New York: Wiley. 1998.

Henderson, N., de Oliveria, J. R., Amaral Souto, H. P. and Pitanga Marques, R. Modeling and analysis of the isothermal flash problem and its calculation with the simulated annealing algorithm, *Industrial and Engineering Chemistry Research*, 40, pp. 6028-6038. 2001.

Hendtlass, T. A combined swarm differential evolution algorithm for optimization problems, *Lecture Notes in Artificial Intelligence*, 2070, pp. 11-18. 2001.

Heidemann, R. A. and Mandhane, J. M. Some properties of the NRTL equation in correlating liquid-liquid equilibrium data, *Chemical Engineering Science*, 28, pp. 1213-1221. 1973.

Hillenbrand, J. B. Studies in the synthesis of energy-efficient evaporation systems. *PhD. Thesis*, Carnegie-Mellon University, Pittsburgh, 1984.

Himmelblau, D. M. *Applied nonlinear programming*. New York: McGraw-Hill. 1972.

Hinterding, R. Gaussian mutation and self-adaptation in numeric genetic algorithms. In *Proceedings of 2nd IEEE Conference on Evolutionary Computation*, pp. 384-389. 1995. Piscataway, New Jersey.

Hoffman, K. L. A method for globally minimizing concave functions over convex sets, *Mathematical Programming,* 20, pp. 22-32. 1981.

Holland, J. H. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press. 1975.

Horst, R. and Pardalos, P. M. *Handbook of global optimization*. Netherlands: Kluwer Academic Publishers. 1995.

Horst, R. and Tuy, H. *Global optimization. Deterministic approaches*. Germany: Springer-Verlag. 1996.

Horst, R., Pardalos, P. M. and Thoai, N. V. *Introduction to global optimization. Nonconvex optimization and its applications.* Netherlands: Kluwer Academic Publishers. 2000.

Hu, N. Tabu search method with random moves for globally optimal design, *International Journal for Numerical Methods in Engineering*, 35, pp. 1055-1070. 1992.

Hua, J. Z., Brennecke, J. F., and Stadtherr, M. A. Enhanced interval analysis for phase stability: cubic EOS models, *Industrial and Engineering Chemistry Research*, 37, pp. 1519-1527. 1998.

Hyprotech. Hysys.Plant: Applications, Version 2.0. Canada: Hyprotech Ltd. 1998.

Iglesias-Silva, G. A., Bonilla-Petriciolet, A., Eubank, P. T., Holste, J. C. and Hall, K. R. An algebraic method that includes Gibbs minimization for performing phase equilibrium calculations for any number of components or phases, *Fluid Phase Equilibria*, 210, pp. 229-245. 2003.

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

Jiang, H., Cai, W. and Shao, X. A random tunneling algorithm for the structural optimization problem, *Physical Chemistry Chemical Physics*, 4, pp. 4782-4788. 2002.

Jockenhoevel, T., Biegler, L. T. and Wachter, A. Dynamic optimization of the Tennessee Eastman process using the OptControlCentre, *Computers and Chemical Engineering*, 27, pp. 1513-1531. 2003.

Jones, D. R., Schonlau, M. and Welch, W. J. Efficient global optimization of expensive global optimization functions, *Journal of Global Optimization*, 13, pp. 455-492. 1998.

Joulia, X., Maggiochi, P., Koehret, B., Paradowski, H. and Barturel, J. J. Hybrid method for phase equilibrium flash calculations, *Fluid Phase Equilibria*, 26, pp. 15-36. 1986.

Kanzow, C. Global optimization techniques for mixed complementarity problems, *Journal of Global Optimization*, 16, pp. 1-21. 2000.

Karaboga, N. and Cetinkaya, B. Performance comparison of genetic and differential evolution algorithms for digital FIR filter design. In *Third International Conference on Advances In Information Systems, Lecture Notes in Computer Science,* 3261, pp. 482-488. 2004.

Katare, S., Bhan, A., Caruthers, J. M., Delgass, W. N. and Venkatasubramanian, V. A hybrid genetic algorithm for efficient parameter estimation of large kinetic models, *Computers and Chemical Engineering*, 28, pp. 2569-2581. 2004.

Kennedy, J. and Eberhart, R. C. *Swarm intelligence*. San Fransciso: Morgan Kaufmann. 2001.

Kim, H., Hayashi, Y. and Nara, K. An algorithm for thermal unit maintenance scheduling through combined use of GA, SA and TS, *IEEE Transactions on Power Systems*, 12, pp. 329-335. 1997.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. Optimization by Simulated Annealing, *Scienc*e, 220, pp. 671-680. 1983.

Klepeis, J. L. and Floudas, C. A. Ab initio tertiary structure prediction of proteins, *Journal of Global Optimization*, 25, pp. 113-140. 2003.

Kocis, G. R. and Grossmann, I. E. Relaxation strategy for the structural optimization of process flow sheets, *Industrial and Engineering Chemistry Research*, 26, pp. 1869-1880. 1987.

Kocis, G. R. and Grossmann, I. E. Global optimization of non-convex mixed-integer non-linear programming (MINLP) problems in process synthesis, *Industrial and Engineering Chemistry Research*, 27, pp. 1407-1421, 1988.

Kocis, G. R. and Grossmann, I. E. A modeling and decomposition strategy for MINLP optimization of process flow sheets, *Computers and Chemical Engineering*, 13, pp. 797-819. 1989.

Kostrowicki, J. and Piela, L. Diffusion equation method of global minimization: performance for standard test functions, *Journal of Optimization Theory and Applications*, 69, pp. 269-284. 1991.

Ku, H. M. and Karimi, I. An evaluation of simulated annealing for batch process scheduling, *Industrial and Engineering Chemistry Research*, 30, pp. 163-169. 1991.

Lampinen J. A Constraint Handling Approach for the Differential Evolution Algorithm. In *Proceedings of the Congress on Evolutionary Computation,* pp. 1468–1473. 2002. Honolulu, Hawaii.

Lee, S. and Grossman, I. E. A global optimization algorithm for non-convex generalized disjunctive programming and applications to process systems, *Computers and Chemical Engineering*, 25, pp. 1675-1697. 2001.

Lee, M. H., Han, C. and Chang, K. S. Dynamic optimization of continuous polymer reactor using modified differential evolution algorithm, *Industrial and Engineering Chemistry Research*, 38, pp. 4825-4831. 1999.

Levy, A. V. and Montalvo, A. The tunneling algorithm for the global minimization of functions, *SIAM Journal on Scientific and Statistical Computing*, 6, pp. 15-29. 1985.

Lin, B. and Miller, D. C. Solving heat exchanger network synthesis problems with tabu search, *Computers and Chemical Engineering*, 28, pp. 1451-1464. 2004a.

Lin, B. and Miller, D. C. Tabu search algorithm for chemical process optimization, *Computers and Chemical Engineering*, 28, pp. 2287-2306. 2004b.

Lin, B., Chavali, S., Camarda, K. and Miller, D. C. Computer-aided molecular design using tabu search, *Computers and Chemical Engineering*, 28, pp. 145-1464. 2005.

Liu, J. and Lampinen, J. A fuzzy adaptive differential evolution algorithm, *Soft Computing*, 9, pp. 448-462. 2005.

Love, R. F., Morris, J. G. and Wesolowsky, G. O. *Facilities location, modes and methods*. Amsterdam: Noth-Holland. 1988.

Luus, R. and Cormack, D. E. Multiplicity of solutions resulting from the use of variational methods in optimal control problems, *Canadian Journal of Chemical Engineering*, 50, pp. 309-311. 1972.

Luus, R., and Jaakola, T. H. I. Optimization by direct search and systematic reduction of the size of search region, *American Institute of Chemical Engineers Journal*, 19, pp. 760-766. 1973.

Luus, R. and Hennessey, D. Optimization of fed-batch reactors by the Luus-Jaakola optimization procedure, *Industrial and Engineering Chemistry Research*, 35, pp. 1948-1955. 1999.

Luus, R. *Iterative Dynamic Programming*. University of Toronto: CRC Press. 2000.

Mantawy, A. H., Abdel-Magid, Y. L. and Selim S. Z. Integrating genetic algorithms, tabu search and simulated annealing for the unit commitment problem, *IEEE Transactions on Power Systems*, 14, pp. 829-836. 1999.

Mathur, M., Karale, S. B., Priye, S., Jayaraman, V. K and Kulkarni, B. D. Ant colony approach to continuous function optimization, *Industrial and Engineering Chemistry Research*, 39, pp. 3814-3822. 2000.

Mayer, D. G., Kinghorn, B. P. and Archer, A. A. Differential evolution – an easy and efficient evolutionary algorithm for model optimization, *Agricultural Systems*, 83, pp. 315-328. 2005.

McDonald, C. M. and Floudas, C .A. Global optimization for the phase and chemical equilibrium problem: application to the NRTL equation, *Computers and Chemical Engineering*, 19, pp. 1111-1139. 1995a.

McDonald, C. M. and Floudas, C. A. Global optimization for the phase stability problem, *American Institute Chemical Engineers Journal*, 41, pp. 1798-1814. 1995b.

McDonald, C. M. and Floudas, C. A. Global optimization and analysis for the Gibbs free energy function using the UNIFAC, Wilson and ASOG equations, *Industrial and Engineering Chemistry Research*, 34, pp. 1674-1687. 1995c.

McDonald, C. M. and Floudas, C. A. GLOPEQ: A new computational tool for the phase and chemical equilibrium problem, *Computers and Chemical Engineering*, 21, pp. 1-23. 1997.

Mekapati Srinivas and Rangaiah, G. P. A Study of Differential Evolution and Tabu Search for Phase Stability Problems, Presented at *American Institute of Chemical Engineers Annual Meeting,* November 2005, Ohio, USA.

Mekapati Srinivas and Rangaiah, G. P. An integrated stochastic method for global optimization of continuous functions. In *Proceedings of the 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, ed by Marquardt, W. and Pantelides, C., pp. 439-444. 2006a.

Mekapati Srinivas and Rangaiah, G. P. Implementation and evaluation of random tunneling algorithm for chemical engineering applications, *Computers and Chemical Engineering*, 30, pp. 1400-1415. 2006b.

Mekapati Srinivas and Rangaiah, G. P. A study of differential evolution and tabu search for benchmark, phase equilibrium and phase stability problems, *Computers and Chemical Engineering*, 31, pp. 760-772. 2007a.

Mekapati Srinivas and Rangaiah, G. P. Differential evolution with tabu list for global optimization and its application to phase equilibrium and parameter estimation problems, *Industrial and Engineering Chemistry Research*, 46, pp. 3410-3421. 2007b.

Mekapati Srinivas and Rangaiah, G. P. Differential evolution with tabu list for solving non-linear and mixed-integer non-linear programming problems, *Industrial and Engineering Chemistry Research*, 46, pp 7126-7135. 2007c.

Merlitz, H. and Wenzel, W. Comparison of stochastic optimization methods for receptor-ligand docking, *Chemical Physics Letters*, 362, pp. 271-277. 2002.

Merz, P. and Freisleben, B. A comparison of memetic algorithms, tabu search and ant colonies for the quadratic assignment problem. In *Proceedings of the Congress on Evolutionary Computation*, pp. 2063-2070. 1999.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, 21, pp. 1087. 1953.

Michalewicz, Z. Genetic algorithms, numerical optimization, and constraints. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 151-158, 1995. San Mateo, USA.

Miettinen, K. M. *Non-linear multi-objective optimization*. USA: Kluwer Academic Publishers. 1999.

Minoux, M. *Mathematical Programming. Theory and Algorithms*, New York: John Wiley and Sons. 1986.

Moles, C. G., Mendes, P. and Banga, J. R. Parameter estimation in biochemical pathways: a comparison of global optimization methods, *Genome Research*, 13, pp. 2467-2474. 2003.

Moore, R. Hansen, E. and Leclerc, A. Rigorous methods for global optimization. In *Recent advances in global optimization,*ed by Floudas, C. A. and Pardalos, P. M., pp. 321-342. Princeton University Press: New Jersey. 1992.

Muhunna, R. L. and Mullen, R. L. Uncertainity in mechanics problems – interval based approach, *Journal of Engineering Mechanics*, 127, pp. 557-566. 2001.

Nelder, J. A. and Mead, R. A simplex method for function minimization, *The Computer Journal*, 7, pp. 308-313. 1965.

Nichita, D. V., Gomez, S. and Luna, E. Multiphase equilibria calculation by direct minimization of Gibbs free energy with a global optimization method, *Computers and Chemical Engineering*, 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 Equilibria*, 194-197, pp. 411-437. 2002b.

Oblow, E. M. SPT: a stochastic tunneling algorithm for global optimization, *Journal of Global Optimization*, 20, pp. 195-212. 2001.

Ohanomah, M. O. and Thompson, D. W. Computation of multicomponent phase Equilibria - Part I. Vapor-liquid Equilibria, *Computers and Chemical Engineering*, 8, pp. 147-156. 1984a.

Ohanomah, M. O. and Thompson, D. W. Computation of multicomponent phase equilibria - Part II. Liquid-liquid and solid-liquid Equilibria, *Computers and Chemical Engineering*, 8, pp. 157-162. 1984b.

Ohanomah, M. O. and Thompson, D. W. Computation of multicomponent phase equilibria - Part III. Multiphase Equilibria, *Computers and Chemical Engineering*, 8, pp. 163-170. 1984c.

Ourique, C. O., Biscaia, E. C. and Pinto Jr, J. C. The use of particle swarm optimization for dynamical analysis in chemical processes, *Computers and Chemical Engineering*, 26, pp. 1783-1793. 2002.

Papamichail, I. and Adjiman, C. S. Global optimization of dynamic systems, *Computers and Chemical Engineering*, 28, pp. 403-415. 2004.

Pardalos, P. M., Romeijin, H. E. and Tuy, H. Recent developments and trends in global optimization, *Journal of Computational and Applied Mathematics*, 124, pp. 209-228. 2000.

Parsopoulos, K. E., Plagianakos, V. P., Magoulas, G. D. and Vrahatis, M. N. Objective function stretching to alleviate convergence to local minima, *Non-linear Analysis*, 47, pp. 3419-3424. 2001.

Parsopoulos, K. E. and Vrahatis, M. N. On the computation of all global minimizers through particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, 8, pp. 211-224. 2004.

Patel, A. N., Mah, R. S. H. and Karimi, I. A. Preliminary design of multi-product non-continuous plants using simulated annealing, *Computers and Chemical Engineering*, 15, pp. 451-469. 1991.

Patkar, P. R. and Venkatasubramanian, V. Genetic algorithms based computer aided molecular design. In *Computer Aided Molecular Design: Theory and Practice*, ed by Luke, E. K. A. and Gani, R., pp. 95-128. Boston: Elsevier. 2003.

Pinter, J. Lipschitz global optimization: some perspective applications. In *Recent Advances in Global Optimization*, ed by Floudas, C. A. and Pardalos, P. M., pp. 399-422. New Jersey: Princeton University Press. 1992.

Prausnitz, J. M., Anderson, T. F., Grens, E. A., Eckert, C. A., Hsieh, R. and O'Connell, J. P. *Computer calculations for multicomponent vapor-liquid and liquid-liquid equilibria*. New Jersey: Prentice Hall. 1980.

Price, K. and Storn, R. *Home page of differential evolution as on December 25*. URL: http://www.ICSI.Berkeley.edu/~storn/code.html. 2006.

Price, K. V., Storn, R. M. and Lampinen, J. A. *Differential evolution: a practical approach to global optimization*. Berlin: Springer. 2005.

Prokopakis, G. J. and Seider, W. D. Dynamic simulation of azeotropic distillation towers, *American Institute Chemical Engineers Journal*, 29, pp. 1017-1029. 1983.

PRO/II Casebook: air separation plant. USA: Simulation Sciences Incorporation. 1993.

Queseda, I. and Grossmann, I. E. Global optimization of bilinear process networks with multicomponent streams, *Computers and Chemical Engineering*, 19, pp. 1219-1242. 1995.

Rajesh, J. K., Gupta, S. K., Rangaiah, G. P. and Ray, A. K. Multi-objective optimization of industrial hydrogen plants, *Chemical Engineering Science*, 56, pp. 999-1010. 2001.

Rangaiah, G. P. Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems, *Fluid Phase Equilibria*, 187-188, pp. 83-109. 2001.

Rinnooy Kan, A. H. G. and Timmer, G. T. Stochastic global optimization methods; part I: clustering methods, *Mathematical Programming*, 39, pp. 27-56. 1987.

Rinnooy Kan, A. H. G. and Timmer, G. T. Stochastic global optimization methods; part II: clustering methods, *Mathematical Programming*, 39, pp. 57-78. 1987.

RoyChowdhury, P., Singh, Y.P. and Chansarkar, R.A. Hybridization of gradient descent algorithms with dynamic tunneling methods for global optimization, *IEEE Transactions on Systems, Man, and Cybernetics – part A: Systems and Humans*, 30, pp. 384-390. 2000.

Ryoo, H. S. and Sahinidis, N. V. Global optimization of non-convex NLPs and MINLPs with applications in process design, *Computers and Chemical Engineering*, 19, pp. 551-566. 1995.

Sahinidis, N. V. and Grossmann, I. E. Convergence properties of generalized Benders decomposition, *Computers and Chemical Engineering*, 15, pp. 481-491. 1991.

Sahinidis, N. V., Tawarmalani, M. and Yu, M. Design of alternative refrigerants via gobal optimization, *American Institute of Chemical Engineers Journal*, 49, pp. 1761-1775. 2003.

Sarkar, D. and Modak, J. M. Optimization of fed-batch bio-reactors using genetic algorithms, *Chemical Engineering Science*, 58, pp. 2283-2296. 2003.

Sarkar, D. and Modak, J. M. Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using non-dominated sorting genetic algorithm, *Chemical Engineering Science,* 60, pp. 481-492. 2005.

Salcedo, R. L. Solving non-convex non-linear programming and mixed-integer non-linear programming problems with adaptive random search, *Industrial and Engineering Chemistry Research*, 31, pp. 262-273. 1992.

Schwetlick, H. and Tiller, V. Numerical methods for estimating parameters in nonlinear models with errors in the variables, *Technometrics*, 27, pp. 17-24. 1985.

Sexton, R. S., Alidaee, B., Dorsey, R. E. and Johnson, J. D. Global optimization for artificial neural networks: a tabu search application, *European Journal of Operational Research*, 106, pp. 570-584. 1998.

Shalloway, D. Packet annealing: a deterministic method for global minimization. Application to molecular conformation. In *Recent Advances in Global Optimization*, ed by Floudas, C. A. and Pardalos, P. M., pp. 433-477. Princeton University Press: New Jersey. 1992.

Shelton, M. R. and Grossmann, I. E. Optimal synthesis of integrated refrigeration systems I: mixed-integer programming model, *Computers and Chemical Engineering*, 10, pp. 577-600. 1986.

Sherali, H. D. and Adams, W. P. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Nonconvex optimization and its applications*. Netherlands: Kluwer Academic Publishers. 1999.

Shi, X. H., Wan, L. M., Lee, H. P., Yang, X. W., Wang, L. M. and Liang, Y. C. An improved genetic algorithm with variable population size and a pso-ga based hybrid evolutionary algorithm. In *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, pp. 1735-1740. 2003. Piscataway, New Jersey.

Sinha, M., Achenie, L. and Ostrovsky, G. V. Environmentally benign solvent design by global optimization, *Computers and Chemical Engineering*, 23, pp. 1381-1394. 1999.

Stadtherr, M. A., Deterministic global optimization for data reconciliation and parameter estimation using error-in-variables approach. In *Recent Developments in Optimization and Optimal Control in Chemical Engineering*, ed by Luus, R., pp. 1-17. India: Research Signpost. 2002.

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

Sun, A. C. and Seider, W. D. Homotopy-continuation method for stability analysis in the global minimization of the Gibbs free energy, *Fluid Phase Equilibria*, 103, pp. 213-249. 1995.

Sundaram, A. and Venkatasubramanian, V. Parametric sensitivity and search-space characterization studies of genetic algorithms for computer aided polymer design, *Journal of Chemical Information and Computer Sciences*, 38, pp. 1177-1191. 1998.

Tan, K. C., Lee, T. H., and Khor, E. F. Evolutionary algorithms with dynamic population size and local exploration for multi-objective optimization, *IEEE Transactions on Evolutionary Computation*, 5, pp. 565-588. 2001.

Tarafder, A., Rangaiah, G. P. and Ray, A. K. Multi-objective optimization of industrial styrene monomer manufacturing process, *Chemical Engineering Science*, 60, pp. 347-363. 2005.

Tawarmalani, M. Ahmed, S. and Sahinidis, N. V. Product disaggregation in global optimization and relaxations of rational programs, *Journal of Global Optimizatio*n, 3, pp. 281-303. 2002.

Teh, Y. S. and Rangaiah, G. P. A study of equation-solving and Gibbs free energy minimization methods for phase equilibrium calculations, *Chemical Engineering Research & Design: Transactions of the Institution of Chemical Engineers, Part A*, 80, pp. 745-759. 2002.

Teh, Y. S. and Rangaiah, G. P. Tabu search for global optimization of continuous functions with application to phase equilibrium calculations, *Computers and Chemical Engineering*, 27, pp. 1665-1679. 2003.

Tessier, S. R., Brennecke, J. F. and Stadtherr, M. A. Reliable phase stability analysis for excess Gibbs energy models, *Chemical Engineering Science*, 55, pp. 1785-1796. 2000.

Teo, J. Exploring dynamic self-adaptive populations in differential evolution, *Soft Computing*, 10, pp. 673-686. 2006.

Timmer, G. T. Global optimization: a stochastic approach. *PhD Thesis*, Erasmus University Rotterdam, Netherlands, 1984.

Tjoa, I. B., and Biegler, L. T. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation of systems, *Industrial and Engineering Chemistry Research*, 30, pp. 376-385. 1991.

Toh, K. A. Global optimization by monotonic transformation, *Computational Optimization and Applications*, 23, pp. 77-99. 2002.

Torn, A. and Zilinskas, A. *Global optimization*. Berlin: Springer-Verlag. 1989.

Trefethen, N., A hundred-dollar, hundred-digit challenge. *SIAM News*, 35, pp. 1. 2002.

Tuy, H. On outer approximation methods for solving concave minimization problems, *Acta Mathematica Vietnamatica*, 8, pp. 3-34. 1983.

Tuy, H. *Convex analysis and global optimization. Nonconvex optimization and its applications*. Netherlands: Kluwer Academic Publishers. 1998.

Umeda, T., and Ichikawa, A. A modified complex method for optimization, *Industrial and Engineering Chemistry Process Design and Development*, 10, pp. 229-236. 1971.

van Laarhoven, P. J. M. and Aarts, E. H. L. *Simulated annealing: theory and applications*. Boston: Kluwer Academic Publishers. 1987.

Vasile, M., Summerer, L. and De Pascale, P. Design of Earth-Mars transfer trajectories using evolutionary-branching technique, *Acta Astronautica*, 56, pp. 705-720. 2005.

Visweswaran, V. and Floudas, C. A. A global optimization algorithm (GOP) for certain classes of non-convex NLPs – II. Application of theory and test problems, *Computers and Chemical Engineering*, 14, pp. 1419-1434. 1990.

Wang, Y. and Achenie, L. E. K. A hybrid global optimization approach for solvent design, *Computers and Chemical Engineering*, 26, pp. 1415-1425. 2002.

White, W. B., Johnson, S. M. and Dantzig, G. B. Chemical equilibrium in complex mixtures, *The Journal of Chemical Physics*, 28, pp. 751-755. 1958.

Wingo, D. R. Fitting three-parameter lognormal models by numerical global optimization – an improved algorithm, *Computational Statistics and Data Analysis*, 2, pp. 13-25. 1984.

Wu, Z. The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation, *SIAM Journal on Optimization*, 6, pp. 748-768. 1996.

Yang, R. L. Convergence of the simulated annealing algorithm for continuous global optimization, *Journal of Optimization Theory and Applications*, 104, pp. 691-716. 2000.

Yatsenko, V. A., Kochubey, S. M., Pardalos, P. M. and Zhan, L. Estimation of chlorophyll concentration in vegetation using global optimization approach. In *Proceedings of The International Society for Optical Engineering*, ed by Hamilton, M. K. and Boles, V. C., 5072, pp. 50-59. 2003.

Yee, A. K. Y, Ray, A. K. and Rangaiah, G. P. Multi-objective optimization of an industrial styrene reactor, *Computers and Chemical Engineering*, 27, pp. 111-130. 2003.

Yiqing, L., Xigang, Y. and Yongjian, L. An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints, *Computer and Chemical Engineering*, 31, pp. 153-162. 2007.

Youssef, H. M., Sait, S. and Adiche, H. Evolutionary algorithm, simulated annealing and tabu search: a comparative study, *Engineering Applications of Artifical Intelligence*, 14, pp. 167-181. 2001.

Yu, H., Fang, H., Yao, P. and Yuan, Y. A combined genetic algorithm/simulated annealing algorithm for large scale system energy integration, *Computers and Chemical Engineering*, 24, pp. 2023-2035. 2000.

Zabinsky, Z. B. Stochastic methods for practical global optimization, *Journal of Global Optimization*, 13, pp. 433-444. 1998.

Zabinsky, Z. B. *Stochastic adaptive search for global optimization. Nonconvex optimization and its applications*. Neterlands: Kluwer Academic Publishers. 2003.

Zamora, J. M. and Grossmann, I. E. A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits, *Computers and Chemical Engineering*, 22, pp. 367-384. 1998.

Zhang, B., Chen, D. and Zhao, W. Iterative ant-colony algorithm and its application to dynamic optimization of chemical processes, *Computers and Chemical Engineering*, 29, pp. 2078-2086. 2005.

APPENDIX A

The following paper is presented at 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering held in July 2006 at Garmisch-Parternkirchen, Germany, and is attached in its original format for the ready reference.

An Integrated Stochastic Method for Global Optimization of Continuous Functions

Mekapati Srinivas[a] and G. P. Rangaiah[a]

[a]*Department of Chemical & Biomolecular Engineering, National University of Singapore, 4 Engineering Drive 4, Singapore 117576*

Abstract

Stochastic methods have attracted growing interest in the recent past as they require less computational effort to provide the global optima. Some of the well known methods are Genetic Algorithm (GA), Differential Evolution (DE) and Tabu search (TS). Each of these methods has a unique feature of escaping from the local minima and/or improved computational efficiency. Though each of these methods has its own advantage(s), they may be trapped in the local minima at times because of the highly non-linear nature of the objective function. In this work, an integrated stochastic method (ISM) is proposed by identifying and then integrating the strong features of DE and TS. A local optimization technique is used at the end to improve the accuracy of the final solution and computational efficiency of the algorithm. The performance of ISM is tested on many benchmark problems and challenging phase equilibrium calculations. The former contain a few to hundreds of local minima whereas the latter has comparable minima. The results show that the performance of ISM is better compared to DE and TS.

Keywords: Differential evolution; Taboo search; Integrated stochastic method; Benchmark problems; Phase equilibrium calculations

1. Introduction

Global optimization methods and their applications are attracting greater attention and interest due to the non-convex nature of the objective functions and the need to find the global optimum. In general, these methods can be classified into two categories: deterministic and stochastic (Pardalos et al., 2000). The former methods guarantee the global optimum under certain conditions whereas the latter do not. However, stochastic methods do not require such restrictive conditions, find global optimum with good success rate and are also computationally more efficient than the latter. Among the many, DE (Storn and Price, 1997) and TS (Chelouah and Siarry, 2000) are some of the promising methods reported in the literature. DE is a population based direct search method especially for non-linear and non-differentiable continuous functions. It mimics biological evolution by performing mutation, crossover and selection steps as in GA to escape from the local minima. The main advantages of DE are its capability to escape from the local minima with a few parameters and fast convergence to the global optimum compared to GA (Karaboga and Cetinkaya, 2004). TS, developed by Glover (1989) for combinatorial optimization, has been used for continuous optimization (Teh and Rangaiah, 2003). It is

a meta heuristic algorithm that guides the heuristics to escape from the local minima. The main feature of TS is it avoids re-visits to the same place during the search thus providing good computational efficiency. Both DE and TS have their own merits and limitations. In this study, an attempt has been made to identify and combine the strong features of DE and TS, to develop a new method (ISM) with good reliability as in DE along with good computational efficiency like in TS. To the best of authors' knowledge, this is the first attempt to develop an integrated method combining the strengths of selected stochastic methods. ISM is tested thoroughly and systematically on benchmark and phase equilibrium problems.

2. Description of ISM

ISM is developed by combining the reliable escaping mechanism of DE (i.e., mutation, crossover and selection steps) with the concept of TS (i.e., avoiding the re-visits during the search using tabu list). This is because our extensive experience showed that DE is more reliable compared to TS whereas the latter is computationally efficient than the former. The authors have chosen DE instead of GA because the former has only a few parameters and computationally efficient compared to GA (Karaboga and Cetinkaya, 2004). The proposed algorithm works better compared to DE and TS alone because by implementing TS concept in DE, i.e., ISM avoids the revisits to the same place during the search providing good computational efficiency while preserving the good reliability of DE. For example, the number of function evaluations taken by ISM and DE for Easom function to locate the global minimum up to a six decimal accuracy is 1855 and 2135 respectively.

ISM (Figure 1) starts by choosing the optimal values for the parameters: population size (NP), amplification factor (A), crossover constant (CR), tabu list size (tls), tabu radius (tr) and maximum number of generations (Mgen). The algorithm initially generates a population of size NP using uniformly distributed random numbers to cover the entire feasible region. A boundary violation check is performed to see if any infeasible solution is generated; the infeasible points are replaced by generating new individuals.

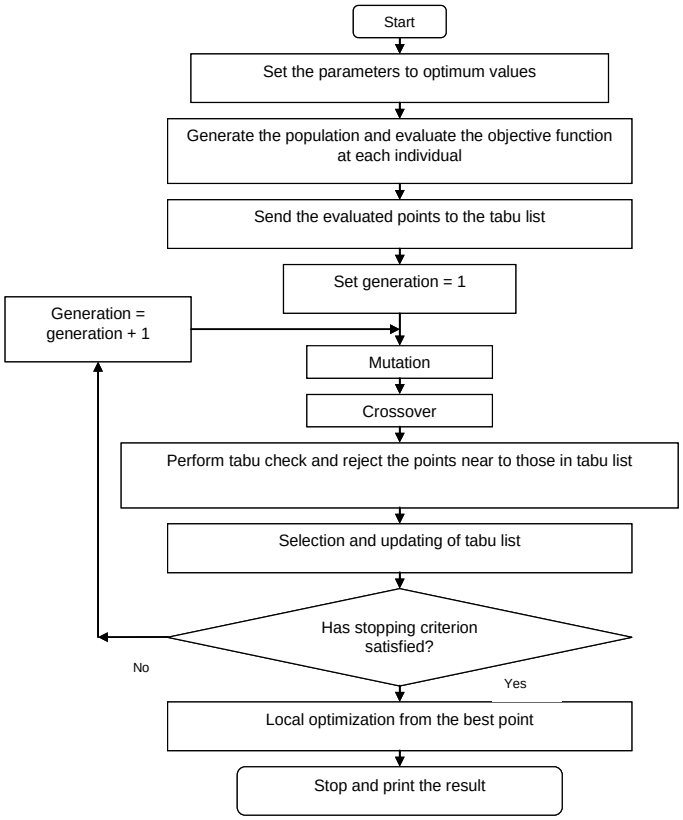


Figure 1: Flow chart of ISM

The objective function is evaluated at each individual, and the best point is captured. The evaluated points are then sent to the tabu list, which will be used to ensure that the algorithm does not search again close to these points. The three main steps: mutation, crossover and selection are performed on the population. The mutant individual is generated for each randomly chosen target individual ($X_{i, G}$) in the population by

$$V_{i, G+1} = X_{r1, G} + A\,(X_{r2, G} - X_{r3, G}); \qquad i = 1, 2, 3, \ldots, NP \qquad (1)$$

where $X_{r1, G}$, $X_{r2, G}$ and $X_{r3, G}$ are the three random individuals chosen in the population of the current generation G, to produce the mutant individual for the next generation, $V_{i, G+1}$. The random numbers r1, r2 and r3 should be different from the running index, i and hence NP should be ≥ 4 to do mutation. A has a value between 0 and 2, controls the amplification of the differential variation between two

random individuals. In the crossover step, a trial individual is produced by copying some elements of the mutant individual to the target individual with a probability equal to crossover constant (CR). The trial individual is then compared with the points in tabu list and is rejected if it is nearby to those in tabu list for the next step. The new individual then competes with $X_{i,G}$ for a place in the next generation; generally, a greedy criterion such as objective function value is used to select the best point for further generations. The tabu list is updated and the process of mutation, crossover and selection is repeated until a stopping criterion such as Mgen is satisfied. The best point found over all the generations is further refined using a local optimization technique, namely, quasi-Newton method.

3. Implementation and evaluation of ISM

The code for ISM is developed in FORTRAN and the performance of the method is compared to that of DE, GA and TS. The FORTRAN codes for TS and GA are taken from Teh and Rangaiah (2003), and DE code is taken from the website http://www.icsi.berkeley.edu/~storn/code.html. A local optimization technique is used at the end of each of these methods to improve the computational efficiency. For the quasi-Newton method, an inbuilt IMSL subroutine is used. The methods are evaluated based on the reliability and computational efficiency in locating the global optimum. The reliability is measured in terms of success rate (SR) i.e., the number of times the algorithm successfully located the global optimum out of 100 trials. A run is said to be successful only if the global optimum is obtained with a fractional error of 10^{-6} in the objective function value. The computational efficiency is measured in terms of number of function evaluations (NFE) required to locate the global optimum. The gradient in the local optimization is calculated numerically, and the NFE includes both the function calls for evaluating the objective and function calls for the gradient.

4. Application to benchmark problems

To asses the performance of ISM, it has been applied to several benchmark problems involving 2 to 20 variables and a few to hundreds of local minima. A brief description of these functions and their global minima are given in Teh and Rangaiah (2003). The stopping criterion used is the maximum number of generations, and not convergence to the global minimum. We have used the former because, in reality, global minimum of application problems is unknown a *priori.* The parameters of DE, GA, TS and ISM are tuned using six out of twelve test functions. The optimal parameters thus obtained are used for the remaining functions too. The tuning is performed to achieve good reliability and computational efficiency, and is performed by varying one parameter at a time while the rest are fixed at their nominal/recent optimum values. The nominal values of parameters in all the methods are chosen based on the literature and preliminary experience with some of the functions.

4.1. Results and discussion

The results for solving benchmark problems by TS, DE, GA and ISM are given in Table 1. Each problem is solved 100 times, each time by generating a random initial estimate. The results are compared in terms of SR and NFE, which is the average over 100 trials. SR is 100% unless otherwise stated.

Table 1: SR and NFE for different test functions

Functions[#]	**GA**	**DE**	**TS**		**ISM**	
	NFE	**NFE**	**SR**	**NFE**	**SR**	**NFE**
GP	20013	3224	100	918	100	976
ES	20007	3224	90	1040	100	2401
SH	20007	3246	100	1033	100	2252
ROS2	20038	3247	99	2021	100	1799
Z2	20003	3223	100	1009	100	420
H3	20009	4824	100	987	100	812
ROS5	20197	8194[c]	76	5275	99	4463
Z5	20077	8026	100	2629	100	1110
ROS10	21977[a]	16661[d]	74	17051	97	6303
Z10	20246	16031	100	8491	100	2477
ROS20	25378[b]	34294[e]	82	44869	99	12341
ZAK20	21170	32041	100	19157	100	5715

Note 1: SR is 94, 89, 99, 98 and 96 at [a], [b], [c], [d] and [e] respectively. Note 2: [#] GP – Goldstein and Price; ES – Easom; SH – Shubert; H3 – Hartmann 3 variable; ROS2, ROS5, ROS10 and ROS20 – Rosenbrock 2, 5, 10 and 20 variables; Z2, Z5, Z10, Z20 – Zakharov 2, 5, 10 and 20 variables.

It is clear from Table 1 that the reliability of ISM is equal to that of DE, GA and is better than that of TS. This is perhaps due to the different escaping mechanisms from local minima corresponding to each method. To escape from local minimum, ISM performs mutation and crossover over a set of individuals as in DE and GA whereas TS makes use of best point obtained in the current iteration to generate neighbors for the next generation, even though it is worse than the best points obtained in the previous iterations. The reliability of TS is less for Easom function because of the flat objective function. As the function is flat, all the neighbors generated in TS will have the same function value trapping the search in that region. On the contrary, GA, DE and ISM located the global minimum region with the help of mutation and crossover. The reliability of all the methods is high for Shubert function though it has 760 local minima. This may be because locating one of the several global minima (around 18) is sufficient to achieve the best function value. The reliability of TS for Rosenbrock functions is less because of the associated narrow global minimum region. NFE of DE and GA is more than that of ISM by a factor of 1.34 (ES) to 7.67 (Z2) and 2.05 (ROS20) to 20.5 (GP) respectively. NFE of ISM is less compared to TS for high dimension problems (a factor of 0.36 (ROS10) to 0.27 (ROS20)), even though it is more for low dimension problems (a factor of 1.06 (GP) to 2.3 (ES)). Overall, the computational efficiency of ISM is better compared to DE, GA and is comparable to that of TS. Both DE and GA are highly reliable but the former seems to be computationally efficient than the latter. Though TS has low NFE for low dimensional problems, its reliability is less compared to that of ISM and DE. By implementing TS concept in DE i.e., in ISM, we are able to maintain reliability as in DE and at the same time with low NFE. The NFE is increasing with number of variables (ROS2 to ROS20) with all the methods due to increase in the size of solution space which makes all the algorithms to generate more number of points to locate potential (global) regions.

5. Application to phase equilibrium problems

Phase equilibrium calculations play a significant role in the design, simulation and optimization of chemical process. Development of robust and efficient method for phase equilibrium has long been a challenge and still it is. The objective is to calculate the number of moles of each phase and its composition at equilibrium given the pressure, temperature, components and feed composition. Basically, methods for multiphase equilibrium calculations can be categorized into two types: equation-solving approach and Gibbs free energy minimization. The modern trend is the treatment of phase equilibrium problems by direct minimization of Gibbs free energy. The objective function in this approach is a highly non-linear and non-convex requiring reliable and efficient global optimization. Several researchers have applied different global optimization methods both deterministic (Burgos-Solórzano et al., 2004) and stochastic (Rangaiah, 2001) using this approach. A review of the works on free energy minimization, and problem formulation can be found in Teh and Rangaiah (2003). The examples considered in this study include vapor-liquid equilibrium (VLE), liquid-liquid equilibrium (LLE) and vapor-liquid-liquid equilibrium (VLLE) problems involving multiple components and popular thermodynamic models. More information about the examples and local and global minima can be found in Teh and Rangaiah (2003). Parameters of DE, GA, TS and ISM are tuned in a similar way as for the benchmark problems. Three out of 10 examples are chosen for tuning, and the optimal parameters thus obtained are used to solve the remaining functions.

5.1. Results and discussion

Table 2: SR and NFE of different methods for solving phase equilibrium problems

Example number	Problem Type	GA	DE	TS		ISM
				SR	NFE	NFE
1 (2)	VLE (2)	20017[a]	7607	99	1348	5882
2 (6a)	VLE (3)	20084	11440	96	1618	5660
3 (6b)	VLE (3)	20086	11445	96	1639	5755
4 (9)	VLE (9)	---	---	---	---	---
5 (11a)	LLE (2)	20017[b]	7628	98	1432	6534
6 (11b)	LLE (2)	20024	7619	100	1359	5255
7 (12)	LLE (2)	20026	7624	100	1367	5352
8 (13)	LLE (3)	20025[c]	11436	94	1575	9838
9 (17)	VLLE (6)	20238[d]	15351	68	5648	12965
10 (18)	VLLE (6)	20262	15355	81	5486	12723

Note: SR is 99, 97, 73 and 75 at [a], [b], [c] and [d] respectively. Bracketed number in the 1st column represents example number in Teh and Rangaiah (2003) and that in the 2nd column refers to the dimension of the problem.

Each example is solved 100 times, each time starting from a different randomly chosen point in the feasible region, and the performance results of the methods are summarized in Table 2. SR is 100% unless otherwise stated. The reliability of ISM and DE is 100% for all examples except for example 4, and is slightly less for GA especially for examples 8 and 9. The reliability of TS is comparable to that of ISM and DE for VLE and LLE problems except for example 8. This is due to the presence of comparable minima (i.e., function values at the trivial and global minimum are -0.35430 and -0.360353 respectively) in that example. As the minima are comparable, better regions where the function value is less than the local minimum become narrower and narrower and the algorithm fails to explore good points causing low SR. On the other hand, ISM and DE were able to escape from the local minimum with their mutation and crossover mechanisms. For example 4, all the methods failed because of comparable minima (function values at local and global minima are -161.5364 and -161.5416 respectively) and more number of variables (9). For VLLE, the reliability of TS is less compared to ISM and DE because of the comparable minima (stated in Teh and Rangaiah, 2003) in these problems. The reliability and computational efficiency of DE is better compared to GA. Computational efficiency of ISM is better compared to both DE and GA but is less than that of TS. NFE of GA and DE is more than that of ISM by a factor of 1.6 to 3.8 and 1.2 to 2.0 respectively. NFE of ISM is more than that of TS by a factor of 2.3 (example 9) to 4.6 (example 5). Although TS has good computational efficiency, its reliability is less for VLLE problems. On the other hand, ISM has good reliability like DE and is also more efficient than DE. Overall, the performance of ISM is better than that of GA, DE and TS. Comparison of CPU times with stochastic (i.e., GA and TS) and deterministic methods for some of these examples can be found in Teh and Rangaiah, 2003.

6. Conclusions and Future work

A new method, namely, ISM is developed by effectively integrating the strong features of DE and TS. The method is first tested for a set of benchmark problems involving 2 to 20 variables and a few to hundreds of local minima, and then for challenging phase equilibrium calculations involving several components and multiple phases. ISM successfully located the global minimum for all the problems, and the performance of ISM is better than that of GA, DE and TS. Our future work includes the implementation of simulated annealing concept in ISM to improve its performance further and its application to various problems such as phase stability analysis and parameter estimation in models.

References

F. Glover, 1989, Tabu search: part 1, *ORSA Journal on Computing*, 1, pp. 190-206.

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

G. P. Rangaiah, 2001, Evaluation of genetic algorithms and simulated annealing for phase equilibrum and stability problems, *Fluid Phase Equilibria*, 187-188, pp. 83-109.

N. Karaboga and B. Cetinkaya, 2004, Performance comparison of genetic and differential evolution algorithms for digital FIR filter design, *Advances In Information Systems: Third International Conference, LNCS*, 3261, pp. 482-488.

P. M. Pardalos, H. E. Romeijin and H. Tuy, 2000, Recent developments and trends in global optimization, *Journal of Computational and Applied Mathematics*, 124, pp. 209-228.

R. Storn and K. Price, 1997, A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 11, pp. 341-359.

R. Chelouah and P. Siarry, 2000, Tabu search applied to global optimization, *European Journal of Operational Research*, 123, pp. 256-270.

Y. S. Teh and G. P. Rangaiah, 2003, Tabu search for global optimization of continuous functions with application to phase equilibrium calculations, *Computers and Chemical Engineering*, 27, pp. 1665-1679.

APPENDIX B

Mathematical Formulation of NLPs and MINLPs in Chapter 5

NLP Problems

Example 1: Minimize $\left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$

Subject to

$$4.84 - \left(x_1 - 0.05\right)^2 - \left(x_2 - 2.5\right)^2 \geq 0$$
$$x_1^2 + \left(x_2 - 2.5\right)^2 - 4.84 \geq 0$$
$$0 \leq x_1, x_2 \leq 6$$

The first constraint is active at the global solution: f = 13.662216 at x = {2.246770, 2.380847}.

Example 2: Minimize

$$0.1365 - 5.843 \times 10^{-7} z_{17} + 1.17 \times 10^{-4} z_{14} + 2.358 \times 10^{-5} z_{13}$$
$$+ 1.502 \times 10^{-6} z_{16} + 0.0321\, z_{12} + 0.004324\, z_5 + 10^{-4} \frac{c_{15}}{c_{16}} + 37.48 \frac{z_2}{c_{12}}$$

Subject to

$$1.5x_2 - x_3 \geq 0$$
$$z_1 - 213.1 \geq 0$$
$$405.23 - z_1 \geq 0$$
$$z_j - a_j \geq 0; \quad j = 2,...,17$$
$$b_j - z_j \geq 0; \quad j = 2,...,17$$
$$z_4 - \frac{0.28}{0.72} z_5 \geq 0$$
$$21 - 3496 \frac{z_2}{c_{12}} \geq 0$$
$$\frac{62212}{c_{17}} - 110.6 - z_1 \geq 0$$

$$704.4148 \le x_1 \le 906.3855$$
$$68.6 \le x_2 \le 288.88$$
$$0 \le x_3 \le 134.75$$
$$193 \le x_4 \le 287.0966$$
$$25 \le x_5 \le 84.1988$$

Though **x** is not appearing in the objective function, z is indirectly related to **x.** The terms z_j, c_j, a_j and b_j are given in Himmelblau (1972). The global optimum reported in Himmelblau (1972) and Deb (2000) are respectively at x = {705.1803, 68.60005, 102.90001, 282.324999, 37.585041} with f = -1.90513 and at x = {707.337769, 68.600273, 102.900146, 282.024841, 84.198792} with f = -1.91460. It was found that one of the constraints ($\frac{62212}{c_{17}} - 110.6 - z_1 \ge 0$) is violated (has a value of -62.771183) at the latter solution and hence the former one (which satisfies all constraints) is used in this study.

Example 3: Minimize $5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

Subject to

$$2x_1 + 2x_2 + x_{10} + x_{11} \le 10$$
$$2x_1 + 2x_3 + x_{10} + x_{12} \le 10$$
$$2x_2 + 2x_3 + x_{11} + x_{12} \le 10$$
$$-8x_1 + x_{10} \le 0$$
$$-8x_2 + x_{11} \le 0$$
$$-8x_3 + x_{12} \le 0$$
$$-2x_4 - x_5 + x_{10} \le 0$$
$$-2x_6 - x_7 + x_{11} \le 0$$
$$-2x_8 - x_9 + x_{12} \le 0$$
$$0 \le x_i \le 1; \quad i = 1,...,9$$
$$0 \le x_i \le 100; \quad i = 10,11,12$$
$$0 \le x_{13} \le 1$$

Six of the constraints are active at the global solution: x = {1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1} with f = -15.

Example 4: Minimize $x_1 + x_2 + x_3$

Subject to

$$
\begin{aligned}
&1 - 0.0025(x_4 + x_6) \geq 0 \\
&1 - 0.0025(x_5 + x_7 - x_4) \geq 0 \\
&1 - 0.01(x_8 - x_5) \geq 0 \\
&x_1 x_6 - 833.33252 x_4 - 100 x_1 + 83333.333 \geq 0 \\
&x_2 x_7 - 1250 x_5 - x_2 x_4 + 1250 x_4 \geq 0 \\
&x_3 x_8 - x_3 x_5 + 2500 x_5 - 1250000 \geq 0 \\
&100 \leq x_1 \leq 10000 \\
&1000 \leq x_2, x_3 \leq 10000 \\
&10 \leq x_i \leq 1000; \quad i = 4,...,8.
\end{aligned}
$$

All six constraints are active at the global solution: x = {579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, and 395.5979} with f = 7049.330923.

Example 5: Minimize

$$(x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6 x_7 - 10x_6 - 8x_7$$

Subject to

$$
\begin{aligned}
&127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0 \\
&282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0 \\
&196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0 \\
&-4x_1^2 - x_2^2 + 3x_1 x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0 \\
&-10 \leq x_i \leq 10; \quad i = 1,..., 7.
\end{aligned}
$$

Two of the constraints (1st and 4th) are active at the global solution: x = {2.330499, 1.951372, -0.477541, 4.365726, -0.624487, 1.038131, 1.594227} with f = 680.630057.

Example 6: Minimize

$$5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Subject to

$$85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \geq 0$$
$$85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \leq 92$$
$$80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \geq 90$$
$$80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \leq 110$$
$$9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \geq 20$$
$$9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leq 25$$
$$78 \leq x_1 \leq 102$$
$$33 \leq x_2 \leq 45$$
$$27 \leq x_i \leq 45; \quad i = 3,4,5.$$

Two of the constraints (2nd and 5th) are active at the global solution: x = {78, 33, 29.995, 45, 36.776} with f = -30665.5.

Example 7: Minimize

$$x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2$$
$$+ 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

Subject to

$$105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \geq 0$$
$$-10x_1 + 8x_2 + 17x_7 - 2x_8 \geq 0$$
$$8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0$$
$$-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \geq 0$$
$$-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0$$

$$-x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \geq 0$$
$$-0.5(x_1 - 0.8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \geq 0$$
$$3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0$$
$$-10 \leq x_i \leq 10; \quad i = 1,...,10$$

Six of the eight constraints are active at the global solution: x = {2.171996, 2.363683, 8.773926, 5.095984, 0.990654, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927} with f = 24.306209.

Example 8: Minimize $-x_4$

Subject to

$$
\begin{aligned}
&x_1 - 1 + k_1 x_1 x_5 = 0 \\
&x_2 - x_1 + k_2 x_2 x_6 = 0 \\
&x_3 + x_1 - 1 + k_3 x_3 x_5 = 0 \\
&x_4 - x_3 + x_2 - x_1 + k_4 x_4 x_6 = 0 \\
&x_5^{0.5} + x_6^{0.5} \le 4 \\
&0 \le x_1, x_2, x_3, x_4 \le 1 \\
&0 \le x_5, x_6 \le 16
\end{aligned}
$$

where $k_1 = 0.09755988$, $k_2 = 0.99k_1$, $k_3 = 0.0391908$ and $k_4 = 0.9k_3$. The global solution is at x = {0.771462, 0.516997, 0.204234, 0.388812, 3.036505, 5.096052} with f = -0.388812. The local solutions reported in Ryoo and Sahinidis (1995) for this problem are at x = {0.390, 0.390, 0.375, 0.375, 16, 0} with f = -0.375 and at x = {1, 0.393, 0, 0.388, 0, 16} with f = -0.3881. The reformulated problem after eliminating the equality constraints is given in Babu and Angira (2006).

Example 9: Maximize

$$\left(\frac{k_1 k_4 \exp(-k_5 t)}{k - k_4}\right)\left[\frac{1 - \exp(-(k_4 - k_5)t)}{k_4 - k_5}\right] - \left(\frac{1 - \exp(-(k - k_5)t)}{k - k_5}\right)$$

Subject to
$$
\begin{aligned}
&0 \le t \le 10 \\
&200 \le T \le 2000
\end{aligned}
$$

where the rate constants (k_i) are given by $k_i = C_i \exp\left\{\frac{-E_i}{R}\left(\frac{1}{T}-\frac{1}{658}\right)\right\}$; $k = k_1 + k_2 + k_3$. The data for C_i and E_i are given in both Umeda and Ichikawa (1971) and Babu and Angira (2006). This example has two global solutions at (t, T) = {0.0781, 978.96} and {0.0757, 983.3} with f = 0.42308.

Example 10: Minimize $-x_1 + x_2$

$$\text{Subject to} \quad \begin{aligned} & x_1 x_2 \le 4 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \end{aligned}$$

The global minimum is at x = {6, 0.666667} with f = -6.666667. There is one local minimum at x = {1, 4} with f = -5.

Example 11: Minimize $400x_1^{0.9} + 1000 + 22(x_2 - 14.7)^{1.2} + x_4$

Subject to

$$\begin{aligned} & x_2 = \exp\left[\frac{-3950}{(x_3 + 460)} + 11.86\right] \\ & 144(80 - x_3) = x_1 x_4 \\ & 0 \le x_1 \le 15.1 \\ & 14.7 \le x_2 \le 94.2 \\ & -459.67 \le x_3 \le 80 \\ & 0 \le x_4 \le \infty \end{aligned}$$

The global solution is at x = {0, 94.177866, 80, 0} with f = 5194.866243 and there is one local solution at x = {15.954, 29.404, 5.864, 669.148} with f = 5339.253. In this study, the upper bound for the x_4 is taken as 1000 which covers both local and global minima. The equality constraints are eliminated by solving them for x_3 and x_4. The resulting reformulated problem has 2 variables and 3 inequality constraints, and is:

Minimize

$$400x_1^{0.9} + 1000 + 22(x_2 - 14.7)^{1.2} + \frac{144\left(540 + \frac{3950}{\ln x_2 - 11.86}\right)}{x_1}$$

Subject to

$$\frac{-3950}{\ln x_2 - 11.86} - 0.33 \geq 0$$

$$540 + \frac{3950}{\ln x_2 - 11.86} \geq 0$$

$$\frac{144\left(540 + \frac{3950}{\ln x_2 - 11.86}\right)}{x_1} \geq 0$$

Example 12: Minimize $-9x_5 - 15x_9 + 6x_1 + 16x_2 + 10x_6$

Subject to

$$x_1 + x_2 = x_3 + x_4$$
$$x_3 + x_7 = x_5$$
$$x_4 + x_8 = x_9$$
$$x_7 + x_8 = x_6$$
$$3x_1 + x_2 = x_{10}(x_3 + x_4)$$
$$x_{10}x_3 + 2x_7 \leq 2.5x_5$$
$$x_{10}x_4 + 2x_8 \leq 1.5x_9$$
$$0 \leq x_1, x_2, x_6 \leq 300$$
$$0 \leq x_3, x_5, x_7 \leq 100$$
$$0 \leq x_4, x_8, x_9 \leq 200$$
$$1 \leq x_{10} \leq 3$$

The global minimum is at x = {0, 100, 0, 100, 0, 100, 0, 100, 200, 1} with f = -400, and one local minimum is at x = {50, 0, 50, 0, 100, 50, 50, 0, 0, 3} with f = -100. There are infinite local solutions to this test problem with a function value of zero at x = {0, 0, 0, 0, 0, 0, 0, 0, 0, a}, where $1 \leq a \leq 3$. The equality constraints are eliminated

by solving them for x_4, x_7, x_8, x_9 and x_{10}. The resulting reformulated problem has 5 variables and 12 inequality constraints, and is:

$$\text{Minimize} \qquad -9x_1 + x_2 + 6x_5 - 5x_6$$

Subject to

$$\frac{3x_1 + x_2}{x_3 + x_4} - 1 \geq 0$$
$$3 - \frac{3x_1 + x_2}{x_3 + x_4} \geq 0$$
$$x_4 + x_8 \geq 0$$
$$200 - (x_4 + x_8) \geq 0$$
$$x_6 - x_7 \geq 0$$
$$200 - (x_6 - x_7) \geq 0$$
$$x_5 - x_3 \geq 0$$
$$100 - (x_5 - x_3) \geq 0$$
$$x_1 + x_2 - x_3 \geq 0$$
$$200 - (x_1 + x_2 - x_3) \geq 0$$
$$\left(\frac{3x_1 + x_2}{x_3 + x_4}\right)x_3 + 2(x_5 - x_3) \leq 2.5x_5$$
$$\left(\frac{3x_1 + x_2}{x_3 + x_4}\right)x_4 + 2(x_6 - x_5 + x_3) \leq 1.5(x_1 + x_2 + x_6 - x_5)$$
$$0 \leq x_1, x_2, x_6 \leq 300$$
$$0 \leq x_3, x_5 \leq 100$$

Example 13: $\text{Minimize} \qquad 35x_1^{0.6} + 35x_2^{0.6}$

Subject to

$$600x_1 - 50x_3 - x_1x_3 + 5000 = 0$$
$$600x_2 + 50x_3 - 15000 = 0$$
$$0 \leq x_1 \leq 34$$
$$0 \leq x_2 \leq 17$$
$$100 \leq x_3 \leq 300$$

The global solution is at x = {0, 16.666667, 100} with f = 189.311627, and there is a local solution at x = {33.333, 0, 300} with f = 286.943. The equality constraints are

eliminated by solving them for x_1 and x_3. The resulting reformulated problem has 1 variable and 3 inequality constraints, and is:

$$\text{Minimize } 35\left(\frac{10000-600x_2}{300+12x_2}\right)^{0.6}+35x_2^{0.6}$$

Subject to

$$\frac{10000-600x_2}{300+12x_2}\geq 0$$
$$34-\frac{10000-600x_2}{300+12x_2}\geq 0$$
$$200-12x_2\geq 0$$
$$0\leq x_2\leq 17$$

Example 14: Minimize x_1+x_2

Subject to

$$x_1^2+x_2^2\leq 4$$
$$x_1^2+x_2^2\geq 1$$
$$x_1-x_2\leq 1$$
$$x_2-x_1\leq 1$$
$$-2\leq x_1,x_2\leq 2$$

The global minimum is at x = {-1.414214, -1.414214} with f = -2.828427. There are two local minima at x = {-1, 0} and x = {1, 0} with function values -1 and 1 respectively.

Example 15: Minimize $x_1^4-14x_1^2+24x_1-x_2^2$

Subject to

$$-x_1+x_2-8\leq 0$$
$$x_2-x_1^2-2x_1+2\leq 0$$
$$-8\leq x_1\leq 10$$
$$0\leq x_2\leq 10$$

This problem has three local solutions at x = {2.60555, 10}, at x = {0.840, 0.386} and at x = {0.732, 0} with function values -86.422205, 10.631 and 10.354 respectively. The global solution is at x = {-3.173599, 1.724533} with f = -118.704860.

Example 16: Minimize $x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2x_4 + 5x_5 - x_6$

Subject to

$$-3x_1 + x_2 - 3x_4 = 0$$
$$-2x_2 + x_3 - 2x_5 = 0$$
$$4x_4 - x_6 = 0$$
$$x_1 + 2x_4 \le 4$$
$$x_2 + x_5 \le 4$$
$$x_3 + x_6 \le 6$$
$$0 \le x_1 \le 3$$
$$0 \le x_2, x_3 \le 4$$
$$0 \le x_4, x_5 \le 2$$
$$0 \le x_6 \le 6$$

The global solution is at x = {0.166667, 2, 4, 0.5, 0, 2} with f = -13.410904 and a local solution is reported with f = -4.259 in Ryoo and Sahinidis (1995). The equality constraints are eliminated by solving them for x_4, x_5 and x_6. The resulting reformulated problem has 3 variables and 8 inequality constraints, and is:

Minimize $x_1^{0.6} + x_2^{0.6} + x_3^{0.4} - 4x_3 + 2\left(\frac{x_2 - 3x_1}{3}\right) + 5\left(\frac{x_3 - 2x_2}{2}\right) - 4\left(\frac{x_2 - 3x_1}{3}\right)$

Subject to

$$\frac{x_2 - 3x_1}{3} \geq 0$$
$$2 - \frac{x_2 - 3x_1}{3} \geq 0$$
$$\frac{x_3 - 2x_2}{2} \geq 0$$
$$2 - \frac{x_3 - 2x_2}{2} \geq 0$$
$$6 - 4\left(\frac{x_2 - 3x_1}{3}\right) \geq 0$$
$$x_1 + 2\left(\frac{x_2 - 3x_1}{3}\right) \leq 4$$
$$x_2 + \frac{x_3 - 2x_2}{2} \leq 4$$
$$x_3 + 4\left(\frac{x_2 - 3x_1}{3}\right) \leq 6$$
$$0 \leq x_1 \leq 3$$
$$0 \leq x_2, x_3 \leq 4$$

MINLP Problems

Example 1: Minimize $2x + y$

Subject to

$$1.25 - x^2 - y \leq 0$$
$$x + y \leq 1.6$$
$$0 \leq x \leq 1.6$$
$$y = \{0,1\}$$

The global solution is at x = 0.5 and y = 1 with f = 2. There is a local minimum at x = 1.118 and y = 0 with f = 2.236.

Example 2: Minimize $-y + 2x_1 + x_2$

Subject to

$$x_1 - 2\exp(-x_2) = 0$$
$$-x_1 + x_2 + y \le 0$$
$$0.5 \le x_1 \le 1.4$$
$$y = \{0,1\}$$

The global solution is at x = {1.375, 0.375}, y = 1 with f = 2.124. The local solution is at x = {0.853, 0.853}, y = 0 with f = 2.558. The reformulated problem after eliminating the equality constraint is given in Costa and Oliveira (2001).

Example 3: Minimize $-0.7y + 5(x_1 - 0.5)^2 + 0.8$

Subject to

$$-\exp(x_1 - 0.2) - x_2 \le 0$$
$$x_2 + 1.1y \le -1$$
$$x_1 - 1.2y \le 0.2$$
$$0.2 \le x_1 \le 1$$
$$-2.22554 \le x_2 \le -1$$
$$y = \{0,1\}$$

The global solution is at x = {0.94194, -2.1}, y = 1 with f = 1.07654. The local solution noticed at x = {0.2, -1}, y = 0 with f = 1.25.

Example 4: Minimize $7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$

Subject to

$$y_1 + y_2 = 1$$
$$z_1 = 0.9[1 - \exp(-0.5v_1]x_1$$
$$z_2 = 0.8[1 - \exp(-0.4v_2]x_2$$
$$x_1 + x_2 - x = 0$$
$$z_1 + z_2 = 10$$
$$v_1 \le 10y_1$$
$$v_2 \le 10y_2$$
$$x_1 \le 20y_1$$
$$x_2 \le 20y_2$$

$$x_1, x_2, z_1, z_2, v_1, v_2 \geq 0$$
$$y_1, y_2 = \{0,1\}$$

The global solution is at $y_2 = 0$, $z_2 = 0$, $v_1 = 3.514237$, $v_2 = 0$ with function value 99.2396. The problem has a sub-optimal solution at x = {0, 15}, y = {0, 1}, v = {0, 4.479} with f = 107.376. The reformulated problem after eliminating the equality constraints is given in Costa and Oliveira (2001).

Example 5: Minimize

$$(y_1-1)^2 + (y_2-2)^2 + (y_3-1)^2 - \log(y_4+1) + (x_1-1)^2 + (x_2-2)^2 + (x_3-3)^2$$

Subject to

$$y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \leq 5$$
$$y_3^2 + x_1^2 + x_2^2 + x_3^2 \leq 5.5$$
$$y_1 + x_1 \leq 1.2$$
$$y_2 + x_2 \leq 1.8$$
$$y_3 + x_3 \leq 2.5$$
$$y_4 + x_1 \leq 1.2$$
$$y_2^2 + x_2^2 \leq 1.64$$
$$y_3^2 + x_3^2 \leq 4.25$$
$$y_2^2 + x_3^2 \leq 4.64$$
$$0 \leq x_1 \leq 1.2$$
$$0 \leq x_2 \leq 1.8$$
$$0 \leq x_3 \leq 2.5; \qquad y_1, y_2, y_3, y_4 = \{0,1\}$$

The problem has a global solution at x = {0.2, 0.8, 1.907878} and y = {1, 1, 0, 1} with f = 4.579582. The highly non-linear nature of the problem results in many local solutions which are reported in Ryoo and Sahinidis (1995).

Example 6: Maximize

$$-5.357854x_1^2 - 0.835689y_1x_3 - 37.29329y_1 + 40792.141$$

Subject to

$$s_1 = a_1 + a_2y_2x_3 + a_3y_1x_2 - a_4x_1x_3$$
$$s_2 = a_5 + a_6y_2x_3 + a_7y_1y_2 + a_8x_1^2 - 90$$
$$s_3 = a_9 + a_{10}x_1x_3 + a_{11}y_1x_1 + a_{12}x_1x_2 - 20$$
$$s_1 \le 92$$
$$s_2 \le 20$$
$$s_3 \le 5$$
$$y_1 = \{78,...,102\}; \quad \text{integer}$$
$$y_2 = \{33,...,45\}; \quad \text{integer}$$
$$33 \le y_2 \le 45$$
$$27 \le x_1, x_2, x_3 \le 45$$

where a_1 to a_{12} are constants, and are given in Angira and Babu (2006). There are many global solutions at $x_1 = 27$, $x_3 = 27$, $y_1 = 78$ with $f = 32217.4$ for any combination of x_2 and y_2.

Examples 7 and 8: Minimize $\sum_{j=1}^{M} \alpha_j P_j R_j^{\beta_j}$

Subject to

$$\sum_{i=1}^{N} \frac{Q_i T_{Li}}{B_i} \le H$$
$$R_j \ge S_{ij} B_i; \qquad i = 1,2,...,N;\ j = 1,2,...,M.$$
$$P_j T_{Lj} \ge t_{ij}; \qquad i = 1,2,...,N;\ j = 1,2,...,M.$$
$$1 \le P_j \le P_j^u \qquad j = 1,2,...,M.$$
$$R_j^l \le R_j \le R_j^u \qquad j = 1,2,...,M.$$
$$T_{Li}^l \le T_{Li} \le T_{Li}^u \qquad i = 1,2,...,N.$$
$$B_i^l \le B_i \le B_i^u \qquad j = 1,2,...,M.$$

where P_j is the integer. N is the number of products, M is the number of processing stages, Q_i is the production rate, P_j is the number of parallel units, R_j is the unit size,

B_i is the batch size, T_{Li} is the cycle time, H is the horizon time, S_{ij} is the size factor, t_{ij} is the processing time and α_j and β_j are the cost coefficients. The bounds over T_{Li} and B_i are calculated as in the following.

$$T_{Li}^{l} = \max \frac{t_{ij}}{P_j^u} \qquad i = 1,2,...,N; j = 1,2,...,M.$$

$$T_{Li}^{u} = \max_j (t_{ij}) \qquad i = 1,2,...,N; j = 1,2,...,M.$$

$$B_i^{l} = \frac{Q_i}{H} T_{Li}^{l} \qquad i = 1,2,...,N.$$

$$B_i^{u} = \min\left(Q_i, \min_j \frac{R_j^u}{S_{ij}} \right) \qquad i = 1,2,...,N; j = 1,2,...,M.$$

Number of variables is M+N and the constraints are 2MN+1. N=2 and M=3 for example 7, and N=5 and M=6 for example 8. The data for the constants and the information about the local and global minima for both examples 7 and 8 are given in Salcedo (1992).