APPLICATION OF DSPSO - TSA TO ECONOMIC DISPATCH AND OPTIMAL POWER FLOW PROBLEMS

SUWIT KHAMSAWANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INTITUE OF THECHNOLOGY LADKRABANG

2010

KMITL-2010-EN-D-018-032

Thesis Title	Application of DSPSO-TSA to Economic Dispatch and Optimal Power Flow Problems
Student	Mr. Suwit Khamsawang
Student ID.	48060001
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2010
Thesis Advisor	Assoc. Prof. Dr. Somchat Jiriwibhakorn

ABSTRACT

This thesis proposes a new approach based on particle swarm optimization (PSO) and tabu search algorithm (TSA). This proposed approach is called distributed Sobol PSO and TSA (DSPSO-TSA). In order to improve the convergence characteristic and solution quality of searching process, two mechanisms had been employed. Firstly, the Sobol sequence is applied to generate the inertia factor used by the conventional process. Secondly, the distributed process is used in order to rapidly reach the global solution.

The search process is divided into multi-stage and has a short-term memory for recognizing the best search history. Finally, to guarantee the global solution, TSA had been activated to adjust the obtained solution of the DSPSO algorithm. To show its effectiveness, the proposed DSPSO-TSA is applied to solve the four case studies of the economic dispatch (ED) problem considering nonsmooth and noncontinuous fuel cost functions of generating units, and two test systems of the optimal power flow (OPF) problem. The simulation results obtained from the DSPSO-TSA are compared with the conventional approaches, e.g. bee colony (BA), differential evolution (DE), genetic algorithm (GA), PSO, TSA, and others shown in the referred literatures. The results show that the proposed approach is very effective and can reach better solution and faster computational time than the conventional methods.

หัวข้อวิทยานิพนธ์	การประยุกต์ใช้วิธี Distributed Sobol Particle Swarm Optimization-Tabu Search Algorithm (DSPSO-TSA) สำหรับปัญหาการจ่ายโหลดอย่างประหยัดและการไหลของกำลังไฟฟ้าอย่างเหมาะสม
นักศึกษา	นายสุวิทย์ คำสว่าง
รหัสประจำตัว	48060001
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2553
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร. สมชาติ จิริวิภากร

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอวิธีการใหม่ที่อาศัยหลักการ particle swarm optimization และ tabu search algorithm แบบดั้งเดิม วิธีการใหม่ที่นำเสนอนี้เรียกว่า "distributed Sobol particle swarm optimization and tabu search algorithm (DSPSO-TSA)" วิธีการที่นำเสนอจะปรับปรุงวิธีการแบบดั้งเดิมโดยอาศัยกลไกในการค้นหาคำตอบสองรูปแบบ กล่าวคือ การสร้าง inertia factor ด้วยกระบวนการ Sobol sequences เพื่อทดแทน inertia factor แบบดั้งเดิม ลำดับต่อมาคือ การกระจายขบวนการค้นหาคำตอบออกเป็นหลายขบวนการ เพื่อช่วยให้กระบวนการค้นหาตอบสามารถลู่เข้าหาคำตอบที่เหมาะสมได้ในเวลาที่สั้นลง

กระบวนการค้นหาคำตอบจะถูกแยกออกเป็นหลายกระบวนการ และใช้เทคนิคการจดจำระยะสั้นในการจำค่าที่ดีที่สุดที่ได้จากการค้นหา และเพื่อรับประกันว่ากระบวนการค้นหาจะสามารถให้ค่าคำตอบที่เหมาะสมที่สุดได้ วิธี TSA จะถูกนำมาใช้เพื่อช่วยเพิ่มประสิทธิภาพในการลู่เข้าหาคำตอบ เพื่อแสดงให้เห็นว่าวิธีการที่นำเสนอมีประสิทธิภาพ วิธี DSPSO-TSA จะถูกนำไปทดสอบกับปัญหา การจ่ายโหลดอย่างประหยัด 4 กรณีศึกษา และปัญหาการไหลของกำลังไฟฟ้าที่เหมาะสม 2 กรณีศึกษา ผลการทดสอบจะถูกนำไปเปรียบเทียบกับวิธีการอื่นๆ ตัวอย่างเช่น วิธี BA, DE, GA, PSO และ TSA รวมทั้งผลการศึกษาที่ได้เคยรายงานไว้ในวารสารที่เป็นที่ยอมรับ พบว่าวิธีการที่นำเสนอสามารถลู่เข้าหาคำตอบที่เหมาะสมได้เร็วกว่าวิธีการดังที่กล่าวมาข้างต้น อีกทั้งยังให้ผลคำตอบที่ดีกว่า

Acknowledgement

I would like to thank my supervisor, Assoc. Prof. Dr. Somchat Jiriwibhakorn for his knowledge, guidance, fruitful discussion, and support in this thesis.

I would also like to thanks to my committee members, i.e. Assoc. Prof. Dr. Vijit Kinnares the chairman, Prof. Dr. Bundhit Eua-Arporn, Assoc. Prof. Dr. Chaiwut Chat-Uthai and Asst. Prof. Dr. Chai Chompoo-Inwai for their valuable comments my thesis.

I would also like to thank my family and my friends for their support and friendship.

Last but not least, thanks for dharma of the Buddha for giving me a common lifestyle.

Suwit Khamsawang

Table of Contents

Table of Contents (Continued)

List of Tables

List of Figures

List of Figures (Continued)

Figure Title **Page**

List of Figures (Continued)

Figure Title **Page**

Chapter 1
Introduction

Power system operation and planning is one important task for power industry. There are several optimization problems, e.g. load frequency control, optimal power flow, and economic dispatch requires real-time solution. Power engineers require special tools to optimally analyze, monitor, and control different aspects of power systems operation and planning. Most of these tools are properly formulated as some sort of optimization problems. The scheduling and type of optimization problems over the time periods i.e. second, minute, hour, day, and year are shown in Figure 1.1 [1].

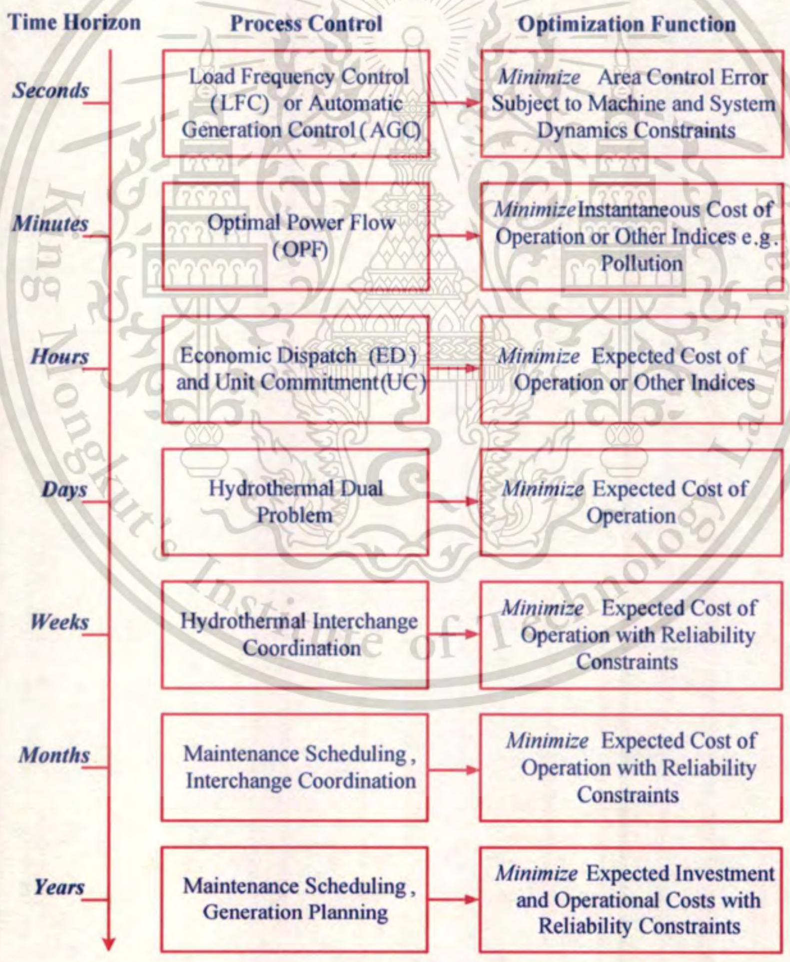


Figure 1.1 Time horizon of the power system optimization problem

Since energy resources are generally decreased and electricity demand keeps growing, electrical generation in all power plants must be economically used. Then, in power system operation, economic dispatch (ED) problem and optimal power flow (OPF) are one of the most important problems to solve to fully utilize fuel for electricity generation which can save generation cost. The main objective of the ED and the OPF problems is to decrease the fuel cost of generation, satisfying various inequality and inequality constraints. In the Classical economic dispatch and the optimal power flow problems, mathematical model of fuel cost for each generating unit was approximated as a single quadratic cost function subjected to generating unit constraints and power system constraints [1].

Classical ED problem is solved using classical mathematical optimization methods, such as the lambda method, gradient method and Newton method [2-3]. These methods based on derivative process and had only good potential for continuous monotonically increasing fuel cost function [1-4]. In practical economic dispatch problem the fuel cost function take account the practical effect of electrical generation, the effect of multi fuel cost function is represented generating unit require the multiple fuel for generation electrical output power which the fuel cost function could be separated as piecewise quadratic cost function (PQCF), valve-point loading effect is represented the opening of steam valves for increase the power generation and prohibited operating zones represented the tabu operating point of generating unit for healthy the steam turbine. In large-scale system the problem is more complex and difficult to find out optimal solution because it is nonlinear function and existent many local optimal solutions.

Many researchers exert to improve many optimization techniques for solving economic dispatch problem with various types of fuel cost function [3-4]. A hierarchical method (HM) using lagrangian function was introduced by C. E. Lin and G. L. Viviani in 1984 [5]. A hopfield neural network to solve ED problem with piecewise quadratic cost function was presented by K.Y. Lee *et el.* [6-7]. K.P. Wong *et el.* had presented simulated annealing (SA) and hybrid GA/SA solve ED problem [8] and [10]. Gerald B.Sheble *et el.* had introduced GA to solve valve-point loading effect added into original quadratic fuel function cost [9]. An improved GA with multiplier updating (IGA_MU) to solve ED problem with multiple fuels and add valve-point effect into fuel cost function was proposed by Chao-Lung Chiang [11]. Zwe-Lee Giang proposed an initially PSO to solve ED problem with prohibited operating zones and compared his proposed with genetic algorithm (GA); the results show that PSO is better than GA in computational times and quality solution terms [13-14]. B.K. Panigrah and *et el.* proposed a clocal algorithm, namely AIS to solve the ED with generators had prohibited operating zones [15]. Cai Jiejin *et el.* applied PSO with chaotic algorithm namely, chaotic particle swarm optimization (CPSO) to solve ED problem with prohibited operating zones [16]. Tabu search is introduced to solve ED problem by many researchers, S. Khamsawang *et el.* are presented an improved tabu search for solving ED problem [17-18], e.g. In addition, S. Pothiya *et el.* had presented multiple tabu search algorithm solve dynamic economic dispatch [19]. Genetic algorithm had applied to solve ED problem with many kinds of fuel function cost.

The optimal power flow (OPF) problem has been well studied over the past few decades [36-38]. The OPF problem could be treated as a nonlinear optimization problem with nonlinear objective function and subject to several equality and inequality constraints. The objective of an OPF problem is to find steady state operation point which minimizes generation cost, loss, reactive power etc. while maintaining an acceptable system performance in terms of limits on generators real and reactive powers, line flow limits, voltage limits. A main obstacle of the OPF problem is the nature of the control variables since some of them are continuous (e.g. real power outputs and voltages) and others are discrete (e.g. transformer tap setting, phase shifters, and reactive injections). The presence of discrete variables makes the optimization problem a non-convex one, which in turn complicates the solution methodology. The most objective function of OPF problem commonly used is the minimization of the overall fuel cost function (convex and non-convex) and reactive power. Researchers proposed different mathematical formulations of the OPF problem, which can be broadly classified as follows:

Many optimization techniques have been applied to solve this problem. Conventional techniques such as Newton method, gradient methods, linear programming, dynamic programming and interior point methods often have problems of convergence and difficulties in locating the global optimum [35-39]. These methods rely on convexity to obtain the global optimum solution and as such are forced to simplify relationships in order to ensure convexity. Despite the fact that some of these techniques have excellent convergence characteristics and various among them are widely used in the industry, some of their drawbacks are [35]:

1. Convergence to the global or local solution is highly dependent on the selected initial guess, i.e. they might converge to local solutions instead of

global ones if the initial guess happens to be in the vicinity of a local solution.

2. Each technique is tailored to suit a specific OPF optimization problem based on the mathematical nature of the objectives and/or constraints.
3. They are developed with some theoretical assumptions, such as convexity, differentiability and continuity, among other things, which may not be suitable for the actual OPF conditions.

During the last year, non-conventional methods such as particle swarm optimization (PSO) [39-42], genetic algorithm (GA) [44], differential evolution (DE) [47-48] and simulated annealing (SA) [91] had been applied to solve the ED problem and the OPF problem.

Although these heuristic methods do not always guarantee to find the globally optimal solution in finite time, they often provide a reasonable solution in an acceptable computation time. For large-scale complex problems, these algorithms need a considerable number of iterations to converge and the solution might get trapped in a suboptimal state. Therefore, to solve the large scale problem, the new method which can find high-quality solutions reliably, with better convergence characteristics and in a reasonably good computation time, is highly expected.

This thesis proposes the distribution procedure of standard particle swarm optimization for solving the ED problem and the OPF problem. The proposed approach is tested on four case studies of the ED problem and two test system of the OPF problem. Simulation results obtained from empirical tests are reported, shown and compared with many non-conventional optimizations had been appeared in the literatures.

1.1 Objective of the thesis

The basic PSO was introduced by J. Kenedy and R. Eberhart in 1995 [21], PSO is a type of modern optimization techniques and a kind of swarm intelligence. PSO has been tested and seen to be high efficiency in solving continuous nonlinear optimization problems [22-25], application to power system optimization for solving voltage security assessment and state estimation by S. Naka *et el.* [28-29] respectively. PSO based on a population searching mechanism such as genetic algorithm and evolutionary programming. PSO inspired by the social behavior of bird flocking or fish schooling. To change population positions from starting points to the final globally optimum point (bird flying to exert abundant food region) using three significant, namely, inertia factor (ω), cognitive (own experience, p_{best}) and social (flocks experience, g_{best}). However, original PSO has often suffers from the problem and return local optima solution.

The objectives of this dissertation are to propose a technique based on distributed particle swarm optimization. To guarantee the global optimal solution and reduce calculation time, the original PSO procedure is developed by using:

- The original PSO procedure was distributed to many procedures. A social significant (g_{best}) of each procedure are compared and elected to be the best social significant (G_{best}) among all group, cognitive experience (p_{best}) of another procedures are used also. Exploitation of G_{best} and p_{best} from another procedure are expected for improvement and guarantee that the proposed procedure can converge to the global solution and reduce the executed time.
- Furthermore, the additional mechanisms i.e. *Sobol inertia factor, high cost elimination* and *reducing search space* are proposed to help and improve the search process in terms of both solution quality and computational time.

To show the effectiveness and robustness of propose techniques, the DSPSO-TSA is tested with:

- The well known numeric function such as Ackley's function, Griewank's function, Rastrigin's function and Schwefel's function. The results obtained the DSPSO-TSA are compared to many heuristic methods i.e. bee colony algorithm (BA), differential evolution (DE), genetic algorithm (GA), particle swarm optimization (PSO) and tabu search algorithm (TSA).
- The ED problem considers different kind of fuel cost function and generating unit constraints. The results yielded from four case studies of the DPSO-TSA are compared with genetic algorithm (GA), particle swarm optimization (PSO), tabu search algorithm (TSA) and other optimization methods in literatures.

- Two case studies of the OPF problem with two test system i.e. six bus test system and IEEE 30-bus test system. In addition, improving voltage profile at load bus case study is considered.

1.2 Outline of the thesis

The thesis is organized as follows:

Chapter 1 depicts the general introduction on the problem description of power system planning and operation. Then, the introduction and surveying on the ED and OPF problem from the journals and the conferences are demonstrated. After that the main objectives and the outline of this dissertation are mentioned.

Chapter 2, presents the basic concept of conventional heuristic methods i.e. BA, DE, GA, PSO and TSA are described. The results obtained from these methods are compared to the proposed method for all problems. The principal of the proposed method which is called "distributed Sobol particle swarm optimization (DSPSO-TSA)", the Sobol sequences which used for generating inertia weight of PSO, high cost elimination and reducing search space are also described.

Chapter 3 shows the application of the proposed optimization technique for exploring the minimum value of five numeric functions. The robustness and effectiveness of DSPSO-TSA are demonstrated in terms of solution quality i.e. average value of function and standard deviation of solution, which are compared to another optimization methods i.e. BA, DE, GA, PSO and TSA.

Chapter 4 the DSPSO-TSA is tested with two power system optimization problems i.e. the ED problem and the OPF problem. In the ED problem, three fuel cost function characteristics and operation constraints of generators such as ramp rate limits and prohibited operating zone are considered for testing the performances of the proposed method. The numerical results obtained from the proposed method are compared to GA, PSO, TSA and emerged results from literatures. The DSPSO-TSA applied to solve the OPF problem is depicted, consequent. The number of particle for each flock and the new parameters of the DSPSO-TSA i.e. c_3 and c_4 are tuned and shown in these case studies. The results yielded from the proposed method are compared to BA, DE, GA, PSO and TSA. The numerical results obtained from the DSPSO-TSA applies for solving the ED and OPF problem clear that the DSPSO-TSA has outperformed than other approaches i.e. yields the high solution quality, low standard deviation of solution and fast converges to global optimum solution.

Chapter 5 gives the conclusion of this thesis and application of the DSPSO-TSA to another problem in future.

Chapter 2

Background and the Proposed Approach

In solving optimization problems, classical optimization techniques, e.g. branch-and-bound, gradient-base methods etc, can be used. However the meta-heuristics such as Genetic Algorithm (GA), Tabu Search Algorithm (TSA), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Bee Algorithm (BA) can also be employed. Figure 2.1 illustrates the evolution of meta-heuristic. In the past, it was clear that the concentration on a sole meta-heuristics is rather restrictive. Meta-heuristics can provide a better solution and higher flexibility when applied for solving actual large scale problems. Among the existing meta-heuristic algorithms, the PSO, which is a stochastic search procedure based on observations of social behaviors of animals, such as bird flocking and fish schooling [21-22], is an interesting option, since it simultaneously evaluates several points in the search space and can utilize global information. Therefore, it is more likely to find the global solution of the given problem. Currently, PSO has been successfully applied in optimizing various continuous nonlinear functions in practice [23-25]. In this chapter, details of the meta-heuristic and the proposed methods are described.

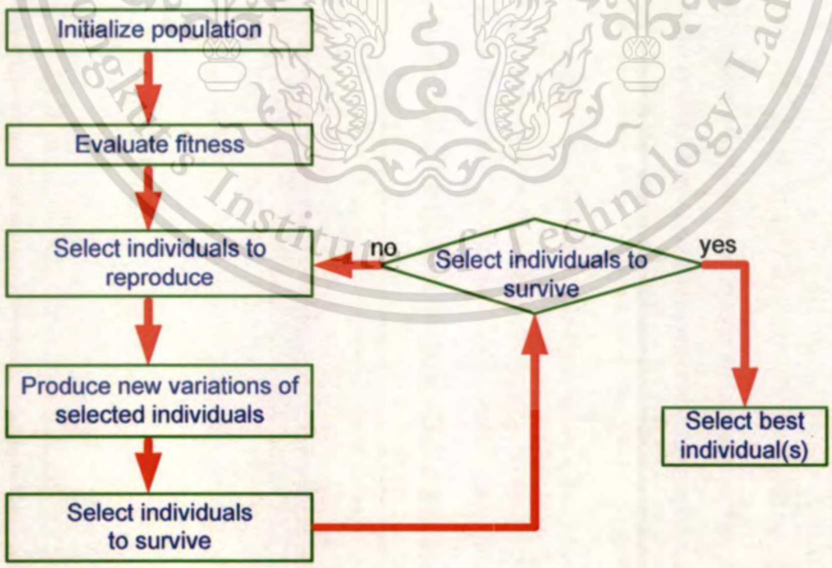


Figure 2.1 Evolution of meta-heuristic

2.1 Genetic Algorithm (GA)

Genetic algorithms are search and optimization techniques based on the dynamics of natural selection and genetics. First proposed by John Holland in 1975 [49], The GA is an optimization technique that performs a parallel, stochastic and directed search to evolve the fittest solution. Different from conventional optimization methods, GA employs the principles of evolution, natural selection and genetics, as inspired by natural biological systems, in a computer algorithm to simulate evolution. Genetic algorithms are now being put to use in a wide range of applications. The algorithm imitates in the process the evolution of population by selecting only fit individuals for reproduction. Therefore, a GA is an optimum search technique based on the concepts of natural selection and survival of the fittest. It works with a fixed-size population of possible solutions of a problem, called individuals, which are evolving in time.

The GA is envisaged an algorithmic concept based on a Darwinian-type survival-of-the-fittest strategy with sexual reproduction, where stronger individuals in the population have a higher chance of creating an offspring. The GA will generally include the three fundamental genetic operations of selection, crossover and mutation. These operations are used to modify the chosen solutions and select the most appropriate offspring to pass on to succeeding generations. After the population is created, the fitness of its individuals is evaluated using the fitness function. The fitness function is a property of the problem being optimized and not the algorithm. It reflects how good is a potential solution and how close it is to the global optimum. So, a good knowledge of the problem is required to create a good fitness function that describes the problem being optimized as accurately as possible. After evaluating their fitness, some individuals are chosen to reproduce and are copied to the mating pool. The number of individuals to be chosen (the size of the mating pool) is a parameter of the algorithm. The crossover ratio defines the percentage of parents in the mating pool which will be affected by the crossover operator. This value is algorithm dependent but it varies around 0.9 for most GA implementations [50]. As shown in Figure 2.2, a chromosome consists of a string of bits, each one of those bits is known as an *allele*. A *gene* is a combination of one or more alleles which determine a characteristic of the individual. After the crossover is done, the offspring chromosomes are mutated. Mutation of chromosomes in binary representation is done by flipping one or more bits in a chromosome. As shown in Figure 2.3, offspring1 was mutated by flipping its 4^{th} bit from 1 to 0. As the case with the crossover operator, the mutation operator may not affect all individuals. The ratio of mutated bits to the total number of bits is known as the mutation ratio and is typically below one percent [53]. As the mutation ratio increases, the algorithm becomes more a random search algorithm. The offspring join the population after evaluating their fitness to fight for survival. This stage is crucial for all individuals; based on their fitness, some of them will survive to the next generation, while others will eliminate. Different mechanisms can be used to select surviving individuals. Many of which are probabilistic techniques that favor more fit individuals. The surviving individuals make-up a new generation and restart the cycle of fitness evaluation, mating selection, reproduction and survival selection. This cycle is repeated until a stopping condition is met, which could be the number of cycles, the solution error of the best individual or any other condition or mix of conditions set by the algorithm designer.

2.1.2 Genetic Components

1) Representation

How can a GA be less dependent on the problem being optimized than a traditional technique? The answer is because the problem is being transformed to the algorithm domain before the GA starts solving the problem. Some people even argue that a GA is problem independent because this transformation is done before the algorithm is invoked. The transformation to the problem domain is done mainly by providing an encoding mechanism for possible solutions to the problem, aka "*representation*", and by creating a *fitness function*. The choice of which representation to use should be done within the context of the problem [55]

In genetic algorithms the design variable or features that characterize an individual are represented in order list called string. Each design variable corresponds to gene and the string of design variables. The oldest and most used scheme of representation is the binary representation.

It could be more suitable for problems with real valued solutions to be represented using a real-valued representation. In this representation, each individual is a real valued number that expresses the value of the solution. Using this representation, there is no need to define a solution precision, such as the one defined for the binary representation, because the search or variation operators will be able to transform this solution to any of the infinite number of possible solutions in the solution space. However this type of representation requires another mechanism of crossover and mutation.

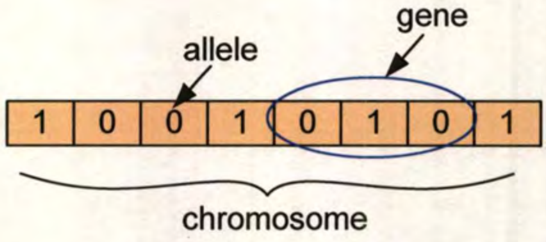


Figure 2.2 Chromosome structure

2) Initialization

A GA solver starts its procedure by creating a population of individuals representing solutions to the problem. The population size (the number of individuals in the population) is among the parameters of the algorithm itself. The initial population is chosen at random. The initial population should contain a wide variety of structures. A big population size helps exploring the solution space but is more computationally expensive than a smaller population which may not have the exploration power of the bigger population.

3) Evaluation function

Each population member is now evaluated for its fitness. Fitness is the value of the only objective function to be optimized. Evaluation is a procedure to determine the fitness of each string in the population and is very much application oriented. The performance of the algorithm is highly sensitive to the fitness values because GA proceeds in the direction of evolving better fit strings and the fitness value is the only information suitable to the GA.

4) Selection

Using the fitness evaluated as above, selection operation is performed to emphasize good population members in mating pool from which a child population is created. The main goal of the selection mechanism is to find an efficient balance between the exploration and exploitation abilities of the search during the run. Too much selection pressure increases the exploitation and the probability of turning the population homogeneous sooner, rather than later. This could diminish the ability of the reproductive operators to produce variation in the population and could decrease the likelihood of converging to a global optimum [52]. However, if the selection procedure strictly selects the very few top of the population. As the selection pressure increases, the algorithms tends to choose the very best of the population to mate and produce the next generation leading to a population that converges to a local optima, this situation is known as premature convergence. On the other hand, if the selection pressure decreases, the algorithm will converge slowly and wander in the search space. It should be clear that the selection pressure is not a parameter that the algorithm designer explicitly set its value; instead, it is influenced by different aspects of the algorithm, especially the selection mechanism used.

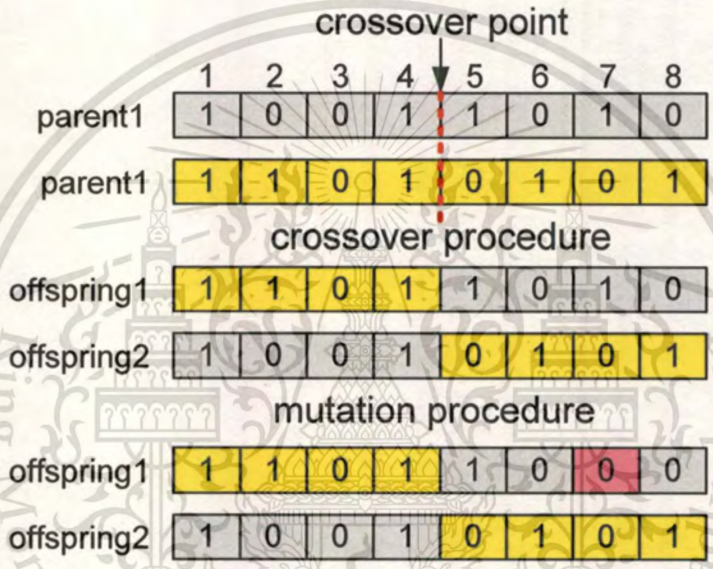


Figure 2.3 Processes of genetic operators

5) Reproduction

Reproduction is a procedure in which individual string are copied according to their objective function value. This operator emphasizes the survival of the fittest in GA's. It selects individual strings in the population according to their fitness. Population size affects the efficiency of the algorithm. The reproductive operators support in some degree both – exploration and exploitation, but, in general, depending on the homogeneity of the population, the recombination (crossover) supports more the exploitation and mutation enforces more exploration in the search. A larger poly would cover more space and prevents premature convergence to local solutions. At the same time a large population needs more evaluations per on and may slow down the convergence rate.

2.1.3 Genetic Operators

In real-world, although parents and their offspring have common characteristics, they are never identical; like father like son, but the son is not a clone of the father. This variation among individual was noted by Charles Darwin in his controversial book [53] where he emphasized that this variation is a major force that drives the evolution process. To mimic this process in GA, researchers have developed variation operators that help the algorithm search the solution space, henceforth; they are sometimes known as search operators [54].

Genetic operators are the stochastic transition rules employed by GA. These operators are applied on each string during each generation to generate a new improved population from old one. Casual produces a population consisting of gene. A gene is a key to the problem and consists of a number of bits. In the next pace each gene is evaluated and given a positive fitness rate. The fitness rate represents the probability and the optimality of a given gene. The gene and the equivalent fitness rate are referred to an individual. Depending on their fitness rate, a certain proportion of the population is chosen and deleted. The existing individuals are recombined and mutated. After the population has been evaluated, the selection process starts again.

These variation operators have two goals: The first one is to produce offspring that resemble their parents, while the second one is to slightly perturb their characteristics. The oldest and most widely used variation operators are the crossover and the mutation operators [50]. They were proposed by John Holland to operate on binary GA, however many other variation operators were proposed to operate on other forms of GA representations. It is to be noted that all syntactic manipulations by variation operators must yield semantically valid results.

1) Recombination or Crossover

The selection process performed does not create new solutions but only ensures that good solutions are emphasized. To create new solutions, recombination or crossover operation is performed on the population members in the mating pool created by selection operation. New solutions *(children)* are created by interchanging or swapping corresponding portions of the grid between two mating pool members *(parents)* which are selected randomly [56]. This operator functions in exactly the same way for real-coded chromosomes as for the binary equivalent. After strings for mating are selected, two chromosomes are chosen and a crossover point is randomly selected.

Crossover is performed at a fairly high probability (i.e. 0.8 to 1). The cross over rate is the parameter that affects the rate at which the cross over operator is applied. A higher crossover rate introduces new strings more quickly into the population. If the cross over rate is too high performance strings are eliminated faster that selection can produce improvements. A low cross over rate may cause stagnation due to the lower exploration rate [51-57].

Let $x = (x_1,\ x_2, ..., x_n)$ and $y = (y_1,\ y_2,\ ...,\ y_n)$ are two individuals from the *parent* population and n is the dimensionality of the problem. Then, a random integer k between 1 and $n-1$ is drawn and two *children* $\bar{x}$ and $\bar{y}$ are obtained as follows:

$$\bar{x} = (x_1,\ x_2, ..., x_k, \bar{x}_{k+1}, ..., \bar{x}_n) \tag{2.1}$$

$$\bar{y} = (y_1,\ y_2, ..., y_k, \bar{y}_{k+1}, ..., \bar{y}_n) \tag{2.2}$$

where $\bar{x} = x_i + \alpha_i(y_i\text{-}x_i)$ and $\bar{y} = y_i + \beta_i(x_i\text{-}y_i)$ for $i = k+1, ..., n$, and α_i, β_i are randomly drawn numbers in the interval [0, 1]. Figure 2.4 shows a two-dimensional illustrative example of how children are produced by shifting one of the coordinates of each *parent*.

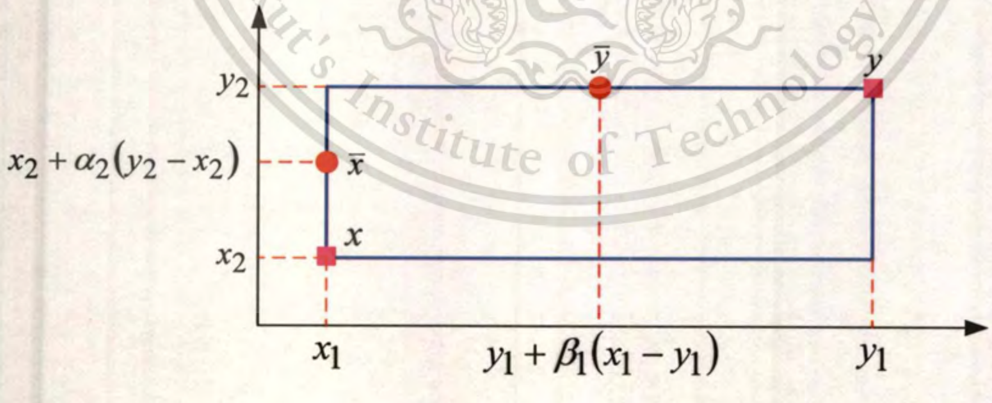


Figure 2.4 Recombination in two-dimensional case

2) Mutation

Unlike recombination operators, mutation operators do not make use of the knowledge of the search space acquired through generations, they do perturb the population by providing random genetic material provided they result in semantically valid results. For binary GA, the mutation operator first determines the positions of the alleles that will undergo mutation. However, the choice of these alleles is made at random with equal probability for each one of them to be mutated (uniform distribution), and their number is determined using the mutation ratio P_m, which is the probability of independently inverting one allele. Second, the operator flips the selected alleles to produce the mutated offspring. However, in the real number case, an infinite set of alternatives exists. Gaussian mutation is an attractive method of choosing an alternative. This operator generates a new value based on a normal distribution, centered over the current value. The standard deviation defines the likelihood of generating a value close to the original. The advantage of this operator is that the standard deviation is amenable to variation during the search, hence providing an adaptable operator.

If the mutation operator does operate on specific values of the parents, it will allow the offspring escape local optima and will help the algorithm explore new regions of the search space. But it will break the link between the parents and their offspring instead of causing small perturbation. On the other hand, if the mutation operator does operate on specific values of the parents to produce their offspring, it may not be very helpful in escaping local optima, but will keep the link between the parents and their offspring.

If an individual is selected for mutation (with probability Pm), we draw a random integer k, $1 \le k \le n$. Then, the k^{th} coordinate of this individual is replaced by a randomly drawn value in the search interval for this coordinate.

2.1.4 Stopping condition

The two stopping criteria were represented, first – to stop when specific generation of the population is reached, and second – to stop when given computational resource is exhausted.

The general form of the performed GA is

Step 1 From the current population $p(G)$, each individual is selected to undergo recombination with probability P_r. If the number of selected individuals is odd, we dispose of the last one selected. All chosen individuals are randomly paired for mating. Each pair produces two new individuals by recombination.

Step 2 Each individual from the current population $p(G)$ is also selected to undergo mutation with probability P_m.

Step 3 From the parent population and the offspring generated by recombination and mutation, the best G individuals are selected to form the new generation $p(G)$.

Step 4 If the halting condition is not satisfied, the algorithm is repeated.

2.2. Tabu Search Algorithm (TSA)

Tabu search algorithm is a heuristic search approach devised for finding optimum solutions and was introduced by Glover [31] in 1989. TSA uses a flexible memory to keep the information about the history search and uses it to create and explore the new solution in the search space [32]. Bland and Dawson were presented TSA to solve optimization problems [33].

Two main components of the TSA are the tabu list (TL) and the aspiration criterion (AC). The tabu list stores all tabu moves that are not permitted to be applied to the present solution. The tabu list consists of moving directions, frequency and recency. Aspiration criterion is employed to determine which move should be freed in such case, i.e. if a certain move is forbidden by tabu restriction, when the aspiration criterion is satisfied, this move can be allowed.

The TSA employed in this thesis faces two tabu restrictions, which are recency and frequency memories. The recency in TL is recording of difference between the current iteration count and the last iteration count at which that moves are changed. The frequency in TL measures the count of each direction of the move. Aspiration criterion is used for measuring the objective function value of each neighborhood, which neighborhood has lower objective function value than pervious count tabu status will be suppressed, otherwise tabu status will be decayed. In order to solve a problem using tabu search, the details of the TSA are given in Figure 2.5 [17-18].

The following notations are used through the description of the TS algorithm for a general combinatorial optimization problem.

S: the set of feasible solutions.

s: the current solution, $s \in S$

s_{best} : the best solution among the trial solutions.

$F(s)$: the objective function of solution s

$s_{neighbor}$: the set of neighborhood of $s_{neighbor} \in S$

TL : tabu list

AC : aspiration criterion

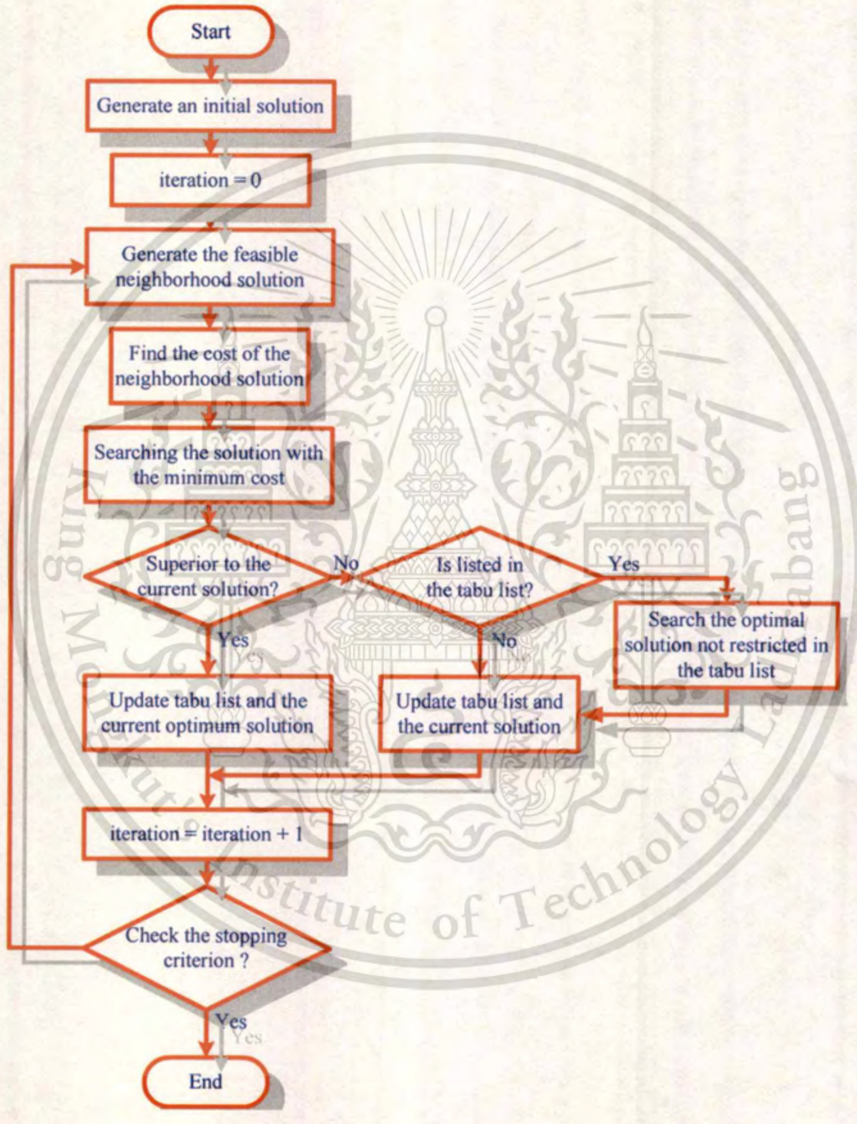


Figure 2.5 Flow chart of tabu search algorithm

The pseudo code of the TSA is described as following [17-19]:

***Step* 1.** Specify *neighborhood size, tubu list, tabu list length, restriction period, frequency maximum, maximum count* and search direction *direction=0.*

***Step* 2.** Set *tabu list* is empty and set *count = 0*

***Step* 3.** Let initial solution $S=[s_1,s_2,...,s_n]$ in search space be a trial vector signifies the d^{th} individual of a population, *n* is the number of dispatchable unit.

***Step* 4.** Calculate the total fuel cost $F(s)$ from s, define F_T is total fuel cost.

***Step* 5.** Set the *Best solution = Current best solution*, *FB=Fb*. Set *FAC=FB*, *FAC* is criterion solution in *AC* for testing.

***Step* 6.** Find a set of trial solution that are neighbor to the s and calculate total fuel cost value $F(s_{neighbor})$ and sort $s_{neighbor}$ that yields best total fuel cost value is s_{best} Set the best of $F(s_{neighbor})$ is $F(s_{best})$.

***Step* 7.** If $F(s_{best}) > FB$ go to step 7, else update solution and TL, set $s^{count+1} = s_{best}$, $FB = F(s_{best})$ and go to step 9.

***Step* 8.** Test *AC*, if $F(s_{best}) < FAC$ set $s^{count+1} = s_{best}$

***Step* 9.** If *recency > restriction period* reset *tabu status*, if *frequency > frequency maximum* set that direction is tabu status.

***Step* 10.** Stop if *count=maximum count*, else increased *count* and go to step 6.

2.3 Particle Swarm Optimization (PSO)

Particle swarm optimization is one of the modern heuristic algorithms and a kind of evolutionary computation technique. Kenedy and Eberhart first introduced PSO in 1995 [21]. PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles. Unlike in the other evolutionary computation techniques, each particle in PSO is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have a tendency to fly towards the better and better search area over the course of search process. Since its introduction in 1995, PSO has attracted a lot of attentions from the researchers around the world. A lot of research results have been reported in the literature. Special sessions have being organized in several conferences including the Congress on Evolutionary Computation since 1998. In 2003, the first IEEE Symposium on Swarm Intelligence was held in Indianapolis, Indiana, USA. The first book dedicated to PSO, Swarm Intelligence, coauthored by James Kennedy, Russell Eberhart with Yuhui Shi (Kennedy, Eberhart and Shi 2001) was published in 2001 by Morgan Kaufmann Publisher.

The PSO algorithm mimics the individual behavior in social system such as fish schooling and bird flocking. The mechanism of this algorithm is the searching parallel processing, which uses a group of individuals similar to other artificial intelligences (AI) as heuristic optimization techniques [34]. Similar to GA, PSO is a population based optimization tool. Unlike GA, PSO has no evolution operators such as crossover and mutation. Since there are only few parameters to adjust in PSO, it is easy to implementation.

In the PSO process, changing the position of each population is called particles change (step size). The particles change is flying around in a multidimensional search space. During search, each particle changes its position by own information of its previous generation and use of best position of neighboring particles. Therefore, the particle change based on the set of particle, particles neighboring and its information history (called velocity of particles).

Let x and v are particle position and velocity of particles in the search space. Therefore, the position of i^{th} particle is represented as vector $x_i = [x_{i1}, x_{i2},...x_{id}]$ and the velocity of i^{th} particle is represented as vector $v_i = (v_{i1}, v_{i2},...v_{id})$ in the d – dimensional space. The best position of particle i^{th} is stored and represented as $pbest_i = (pbest_{i1}, pbest_{i2},...pbest_{id})$. The best position among all particles is represented as $gbest$. Using the previous information and own experienced, the updated velocity of particle i^{th} is calculated by the following equation.

$$v_i^{(t+1)} = \omega \times v_i^{(t)} + c_1 \times rand() \times (pbest_i - x_i^{(t)}) + c_2 \times rand() \times (gbest - x_i^{(t)}) \tag{2.3}$$

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{iter_{\max}} \times iter \tag{2.4}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}, \ \mathrm{i} = 1,2,...,\mathrm{n}, \tag{2.5}$$

where $v_i^{(t)}$ is velocity of particle i at generation t, which $V_d^{\min} \le v_i^{(t)} \le V_d^{\max}$, c_1 and c_2 are weighting factors, $rand()$ is random number between 0 and 1, t is number of generations (or iterations), ω is inertia weight factor, $\omega_{\min}$ and $\omega_{\max}$ are initial and final inertia weight factor, $iter$ is current iteration number, $iter_{\max}$ is maximum iteration number and $x_i^{(t)}$ is position of particle i^{th} at generation t.

Each particle flies from the current position to the next position by using the updated velocity in (2.3). Therefore, new position of the particle is defined by equation (2.5). Figure 2.6 shows the position updating mechanism of particle in PSO. The inertia weight factor ω is provided to improve performance of the PSO, thus large value of ω provided global exploration feature and small value provided local exploration feature. Generally, the inertia factor is usually calculated by equation (2.4). During search process, the value decreases linearly from $\omega_{\max}$ .and $\omega_{\min}$. To enhance the performance of the PSO, it can be obtained by carefully selecting inertia factor. In this thesis, a new technique based on the Sobol sequence is proposed to generate a more propoer inertia factor. Figure 2.7 illustrates the flow chart of the PSO.

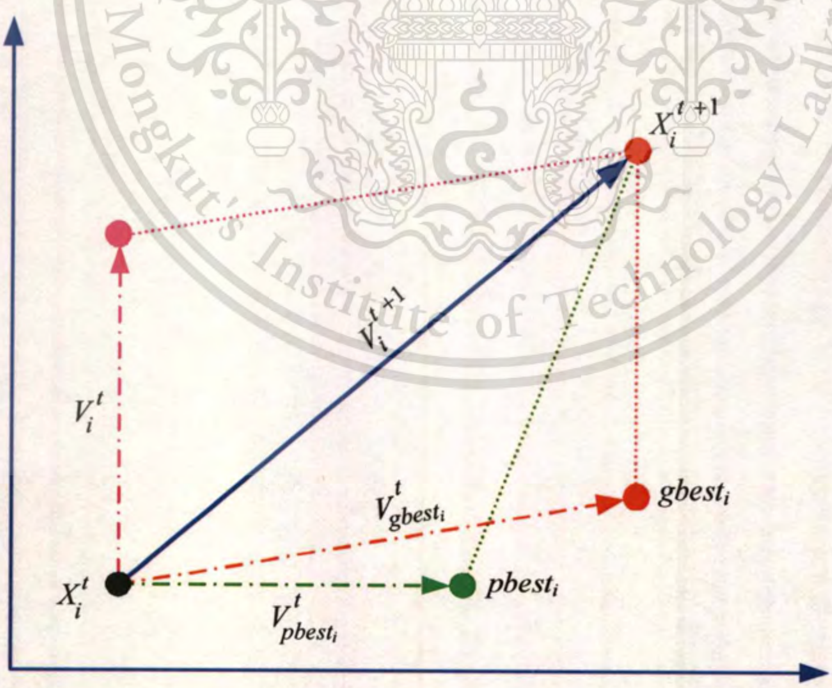


Figure 2.6 Illustrating the dynamics of a particle in PSO

The pseudo code of the conventional PSO is described as follows:

Step 1. Randomly generate the particles, searching point and velocities within the limits specified for each individual in each procedure. These initial individuals must be satisfied the operation constraints.

Step 2. The objective function of each particle is evaluated.

Step 3. These values are set as $pbest$ and the minimum value from all evaluation values is defined as $gbest$.

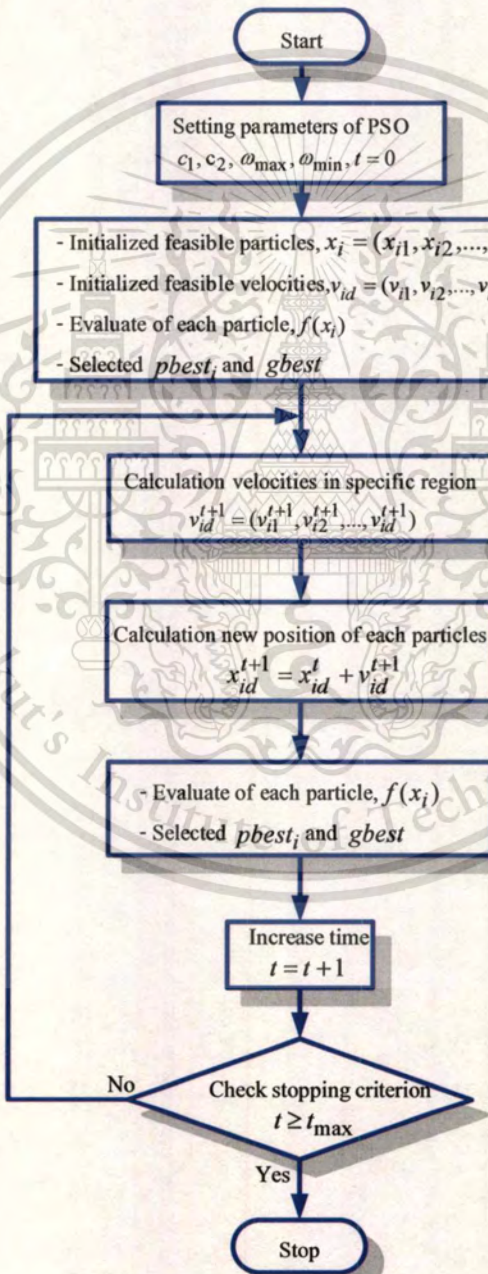


Figure 2.7 Flow chart of particle swarm optimization

Step 4. Using *pbest* and *gbest* modified the new velocity of all the particles in each dimensions, that given by equation (2.3).

Step 5. The position of each particle for all the dimensions is adjusted given be equation (2.5).

Step 6. The adjusted positions of all particles are evaluated for new objective function. If the new objective function value is better than previous *pbest* and satisfied the operation constraints, the current value is to *pbest* . If the best new value of *pbest* is better than *gbest* , best *pbest* is set to *gbest* .

Step 7. If the number of iterations reaches to the appropriate condition, interchange solutions between procedures, the best solution of all procedure is sorted as the new best solution and used as forward initial solution for every procedure.

Step 8. Stop if the terminations criterions are satisfied, then go to step 9. Otherwise, go to step 2.

Step 9. The minimum of the *gbest* in each procedure at last iteration is set as global solution, on the other hand, it's the optimal real power generation of each generating unit with the minimum total generation cost.

2.4 Differential Evolution (DE)

Differential evolution (DE) is an evolutionary algorithm proposed by Storn and Price (1995) [58]. The DE is a simple population based, stochastic parallel search evolutionary algorithm for global optimization and is capable of handling non-differentiable, nonlinear and multi-modal objective functions [59-60]. In DE the population consists of real-valued vectors with dimension D that equals the number of design parameters. The size of the population is adjusted by the parameter Np . While DE shares similarities with other evolutionary algorithms (EA), it differs significantly in the sense that distance and direction information from the current population is used to guide the search process. DE uses the differences between randomly selected vectors (individuals) as the source of random variations for a third vector (individual), referred to as the target vector. Trial solutions are generated by adding weighted difference vectors to the target vector. This process is referred to as the mutation operator where the target vector is mutated. A recombination or crossover step is then applied to produce an offspring which is only accepted if it improves on the fitness of the parent individual.

Easy method for implementation and has a few parameters for tuning made the algorithm quite popular very soon [47-48]. Like other evolutionary algorithms, the DE also starts with a population of NP D-dimensional search variable vectors. The basic DE algorithm is described in more detail below with reference to the three evolution operators: mutation, crossover, and selection. The initial population is uniformly distributed in the search space, represented as

$$X_i^G = [x_{r1}^G, x_{r2}^G, ..., x_{rD}^G] \tag{2.6}$$

For each variable, there may be a certain range within upper and lower limits. The initial population should cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds:

2.4.1 Mutation

DE does not use a predefined probability density function to generate perturbing fluctuations. It relies upon the population itself to perturb the vector parameter. According to the mutation operator, for each individual, after initialization, DE creates a *donor* vector V_i^G corresponding to each population member or *target* vector X_i^G in the current generation through mutation and sometimes using arithmetic recombination too. It is the method of creating this donor vector that differentiates one DE scheme from another. Five most frequently referred strategies, implemented in the public-domain DE codes for producing the donor vectors are listed below:

DE/rand/1: $$V_i^G = x_{r1}^G + F \times (\mathrm{x}_{r2}^G - x_{r3}^G) \tag{2.7}$$

DE/best/1: $$V_i^G = x_{best}^G + F \times (\mathrm{x}_{r2}^G - x_{r3}^G) \tag{2.8}$$

DE/target-to-best/1: $$V_i^G = x_i^G + F \times (\mathrm{x}_{best}^G - x_i^G) + F \times (\mathrm{x}_{r1}^G - x_{r2}^G) \tag{2.9}$$

DE/best/2: $$V_i^G = x_{best}^G + F \times (\mathrm{x}_{r1}^G - x_{r2}^G) + F \times (\mathrm{x}_{r3}^G - x_{r4}^G) \tag{2.10}$$

DE/rand/2: $$V_i^G = x_{r1}^G + F \times (\mathrm{x}_{r2}^G - x_{r3}^G) + F \times (\mathrm{x}_{r3}^G - x_{r4}^G) \tag{2.11}$$

The indices $r1$, $r2$, $r3$, $r4$ and $r5$ are mutually exclusive integers randomly chosen from the range [1,*NP*], and all are different from the base index *i*. These indices are randomly generated once for each donor vector. The scaling factor F is a positive control parameter for scaling the difference vectors which controls the amplification of the differential variation. Larger value for F result in higher diversity in the generated population and lower values cause faster convergence. The x_{best} is the best individual vector with the best fitness (i.e. lowest objective function value for a minimization problem) in the population at generation G. Note that some of the strategies for creating the donor vector may be mutated recombinants, for example, equation (2.8) listed above basically mutates a two-vector recombinant: $x_i^G + F \times (x_{best}^G - x_i^G)$.

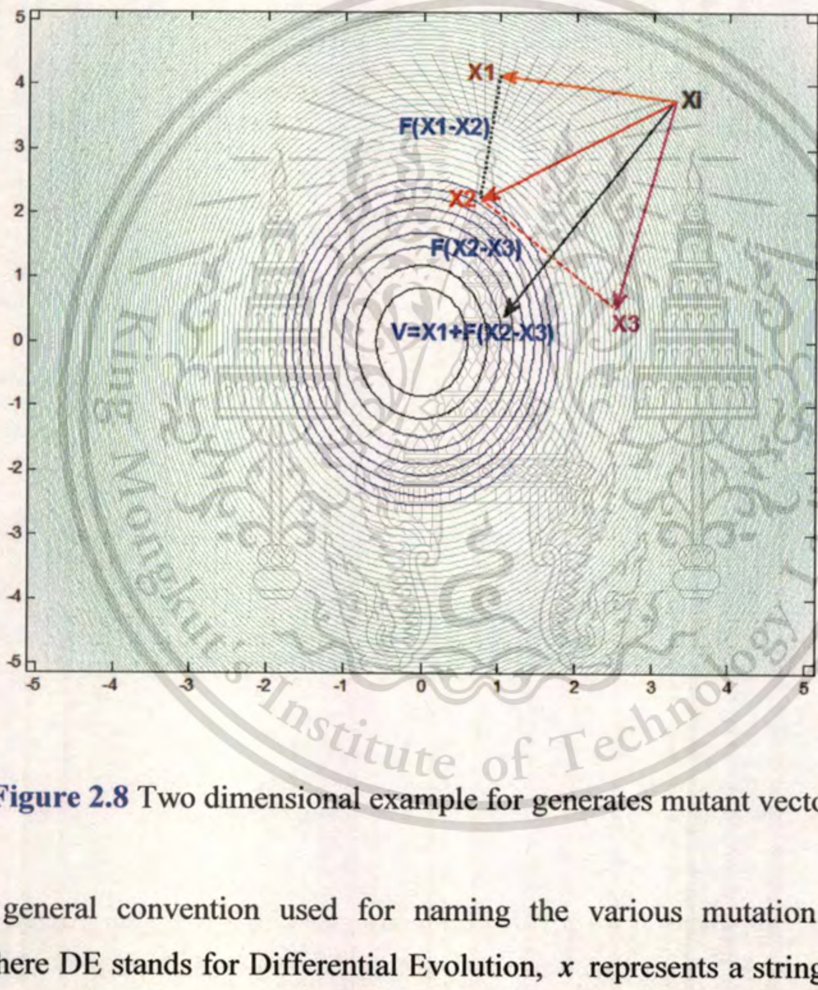


Figure 2.8 Two dimensional example for generates mutant vector.

The general convention used for naming the various mutation strategies is DE/x/y/z, where DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used (exponential, binomial) [61]. The following section discusses the crossover step in DE. The process of two dimensional mutation examples is illustrated in Figure 2.8.

2.4.2 Crossover

Following the mutation phase, the crossover (recombination) operator is applied on the population to increase the potential diversity of the population. Crossover scheme comes to play, the following vector is adopted:

$$U_i^G = [u_{i,1}^G, u_{i,2}^G, ..., u_{i,D}^G] \tag{2.12}$$

In exponential crossover, first one chooses an integer n randomly among the numbers [1, D]. This integer acts as a starting point in the target vector, from where the crossover or exchange of components with the donor vector starts. Choose another integer L from the interval [1, D]. L denotes the number of components; the donor vector actually contributes to the target. Where the angular brackets $\langle\ \rangle_D$ denote a modulo function with modulus D. After a choice of n and L the trial vector is obtained as:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G & \text{for } \ j = \langle n \rangle_D, \langle n+1 \rangle_D, ... \langle n+L-1 \rangle_D \\ x_{i,j}^G & \text{for all other } j \in [1, \text{D}], \end{cases} \tag{2.13}$$

In this way, each trial vector X_i^G, an offspring vector U_i^G is created. This idea of crossover is illustrated in Figure 2.9, for D=7, n=2 and L=3. In order to decide the new vector U_i^G shall become a population member of the next generation, the selection process is evolved, at selection process can be expressed as

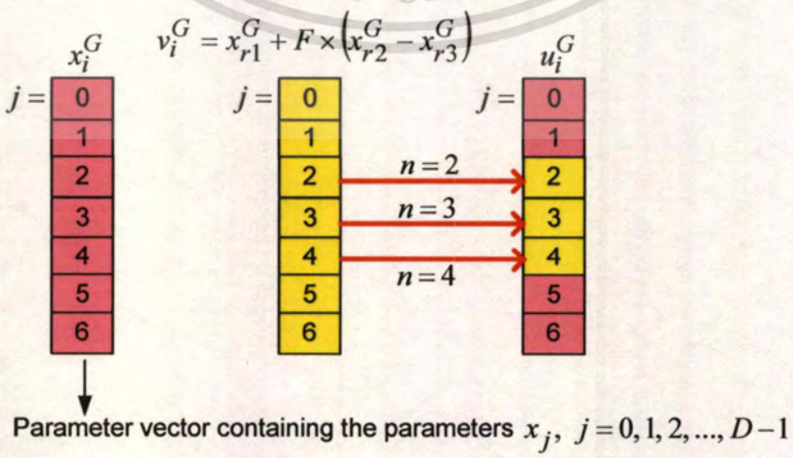


Figure 2.9 Illustration of the exponential crossover processes.

In binomial crossover, crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is less than or equal to the Cr value [71]. In this case the number of parameters inherited from the donor has a (nearly) binomial distribution. The scheme may be outlined as:

$$u_{i,1}^{G} = \begin{cases} v_{i,1}^{G} & \text{if } \left(rand_{i,,1} \le Cr\right) \\ x_{i,1}^{G} & \text{otherwise,} \end{cases} \tag{2.14}$$

where CR is crossover rate in the range [0, 1] and appears as a control parameter of DE just like F, $rand_{i,1}$ is a uniformly distributed random number, which is called anew for each 1^{th} component of the i^{th} parameter vector.

2.4.3 Selection

It is an approach which must decide which vector $\left(U_i^G \; or \; X_i^G\right)$ should be a member of next generation $(G+1)$. DE evolution implements a very easy simple selection procedure. The generated offspring, $\left(U_i^G\right)$ replaces the parent, $\left(X_i^G\right)$ only if the fitness of the offspring is better than that of parent. The selection operation is depicted as:

$$X_i^{G+1} = \begin{cases} U_i^G & \text{if } \; f(U_i^G) \le f(X_i^G) \\ X_i^G & \text{otherwise.} \end{cases} \tag{2.15}$$

where $f()$ is the objective function to be minimized. Thus, if the new trial vector (X_i^{G+1}) yields the better objective function value than X_i^G, X_i^{G+1} replaces its target in the next generation; otherwise the target vector (X_i^G) is retained.

The pseudo code of the DE describes as following steps:

Step 1: Set the generation number $G = 0$ and randomly initialize a population of NP individuals $X_i^G = [x_{r1}^G, x_{r2}^G, ..., x_{rD}^G]$ and each individual uniformly distributed in the range [$X_{\min}, X_{\max}$].

Step 2: Generate a donor vector $V_i^G = [V_{r1}^G, V_{r2}^G, ..., V_{rD}^G]$ corresponding to the i^{th} target vector X_i^G via one of the different mutation schemes of DE (equations (2.8) to (2.12))

Step 3: Generate a trial vector $U_i^G = [u_{i,1}^G, u_{i,2}^G, ..., u_{i,D}^G]$ for the i^{th} target vector X_i^G through binomial crossover (equation (2.14)) or exponential crossover (equation (2.15))

Step 4: Evaluate the trial vector U_i^G and applied selection procedure vector to choose vector U_i^G *or* X_i^G should be a member of next generation $(G+1)$.

Step 5: If stopping criterion is not met, increase the generation number $G = G + 1$ and go to step 2-5 repeat until stopping criterion are met.

2.5 The Bee Colony Optimization

2.5.1 Bee foraging process in nature

During the harvesting season, a bee colony employs part of its population to scout [62] the fields surrounding the hive. Scout bees move randomly looking for food sources. In particular, scout bees look for flower patches where nectar is abundant, easy to extract, and rich in sugar content. When they return to the hive, scout bees deposit the nectar (or pollen) that they have collected during the search process. Those bees that found a high-quality food source signal the position of their discovery to resting mates through a ritual known as the "waggle dance" [63]. The waggle dance is performed in a particular area of the hive called the "dance floor", and communicates three basic pieces of information regarding the flower patch: the direction where it is located, its distance from the hive, and its quality rating [64]. After the waggle dance, the dancer bee goes back to the flower patch, followed by other nest mates recruited from the hive. The number of recruited bees depends on the quality rating of the patch. Flower patches that contain rich and easily available nectar or pollen sources attract the largest number of foragers [63]. Once a recruited forager returns to the hive, it will in turn waggle dance to direct other idle bees towards the food source. Thanks to this mechanism, the trail towards the most profitable food sources is reinforced [65], and the bee colony optimizes the efficiency of the food foraging process.

2.5.2 The Bee Algorithm

As mentioned before, the Bees Algorithm takes inspiration from the food foraging strategy of honey bees to search for the best solution of a given optimization problem. Each point in the solution space is thought of as a food source. "Scout bees" randomly sample the solution space, and via a "fitness function" report the quality of the visited locations. The solutions are ranked, and other "bees" are recruited to search the solution space in the neighborhood of the highest ranking locations. The neighborhood of a solution is called a "flower patch". The Bees Algorithm locates the most promising flower patches, and selectively explores those patches looking for the global peak of solution fitness. The rest of this section describes more detail. The Bees Algorithm is characterized by a number of parameters that are varied to achieve more explorative or exploitative search strategies. These parameters are,

- number of scout bees *n*.
- number of site selected out of n visited sites *m*.
- number of best sites out of m selected sites *e*.
- number of bees recruited for neighborhood search around the *e* elite sites (*nep*).
- number of bees recruited for the other *m-e* selected sites *nsp*.

The stopping criterion depends on the nature of the problem domain, and can be either the location of a solution of fitness above a pre-defined threshold, or the completion of a predefined number of evolution cycles. Figure 2.10 shows the flowchart of the basic Bees Algorithm. The algorithm starts with *n* scout bees randomly scattered across the solution space. Then, the algorithm enters the main loop, which is composed of four calculation phases. In the fitness evaluation phase, the fitness of each visited site is evaluated.

In the local search phase, the scout bees that located the *m* sites of highest fitness are selected for waggle dancing, that is to recruit nest mates for neighborhood search. For each selected site, the recruited bees are randomly placed in the surroundings of the high fitness location reported by the scout bee. More bees are assigned to search in the vicinity of the best *e* sites. Accordingly, the local search is more thorough in the neighborhood of the best *e* sites, which represent the most promising locations of the solution space. This fitness-based differential recruitment is a key operation of the Bees Algorithm, since it defines the extent of the exploitative search. For each patch, the fitness of the locations visited by the recruited bees is evaluated. If one of the recruited bees locates a position of higher fitness than the scout bee, that recruited bee is chosen as the new scout bee and becomes the dancer once returned to the hive. In nature, the feedback mechanism is different since all the bees involved in the foraging process perform the waggle dance once returned to the hive. In the Bees Algorithm, for each patch only the bee that visited the highest location of the fitness landscape performs the dance. The highest local peak visited so-far is thus taken as a representative of the whole neighborhood.

In the global search phase, a number of bees are placed randomly across the search space to scout for new flower patches. Random scouting represents the exploration effort of the Bees Algorithm. At the end of each iteration, the new scout bees population of the colony is formed out of two population groups. The first group is composed of the bees associated with the centre (the best solution) of each flower patch, and represents the results of the local exploitative search. The second group is composed of scout bees associated with a randomly generated solution, and represents the results of the global explorative search. A new optimization cycle is then started, selecting from the overall scout population *m* sites for local exploration. This sequence of evolution cycles is interrupted when the stopping criterion is met. Two new procedures were recently introduced in order to increase the performance of the algorithm and avoid superfluous computations. The first new procedure is called henceforth "neighborhood shrinking". The size of the flower patches is initially set to a large value. For each patch, the initial size is kept unchanged as long as the local search procedure yields higher points of fitness. If the neighborhood search fails to bring any improvement of performance, the patch size is decreased. This strategy aims at making the local search more exploitative, searching more densely the area around the local optimum.

The second new procedure is applied when no fitness improvement is gained within a flower patch from shrinking the neighborhood. In this case, the local search procedure is assumed to have reached the top of a local fitness peak, and no further progress is possible. Consequently, the location of the peak is recorded and the exploration of the patch is terminated. This procedure is called henceforth "site abandonment".

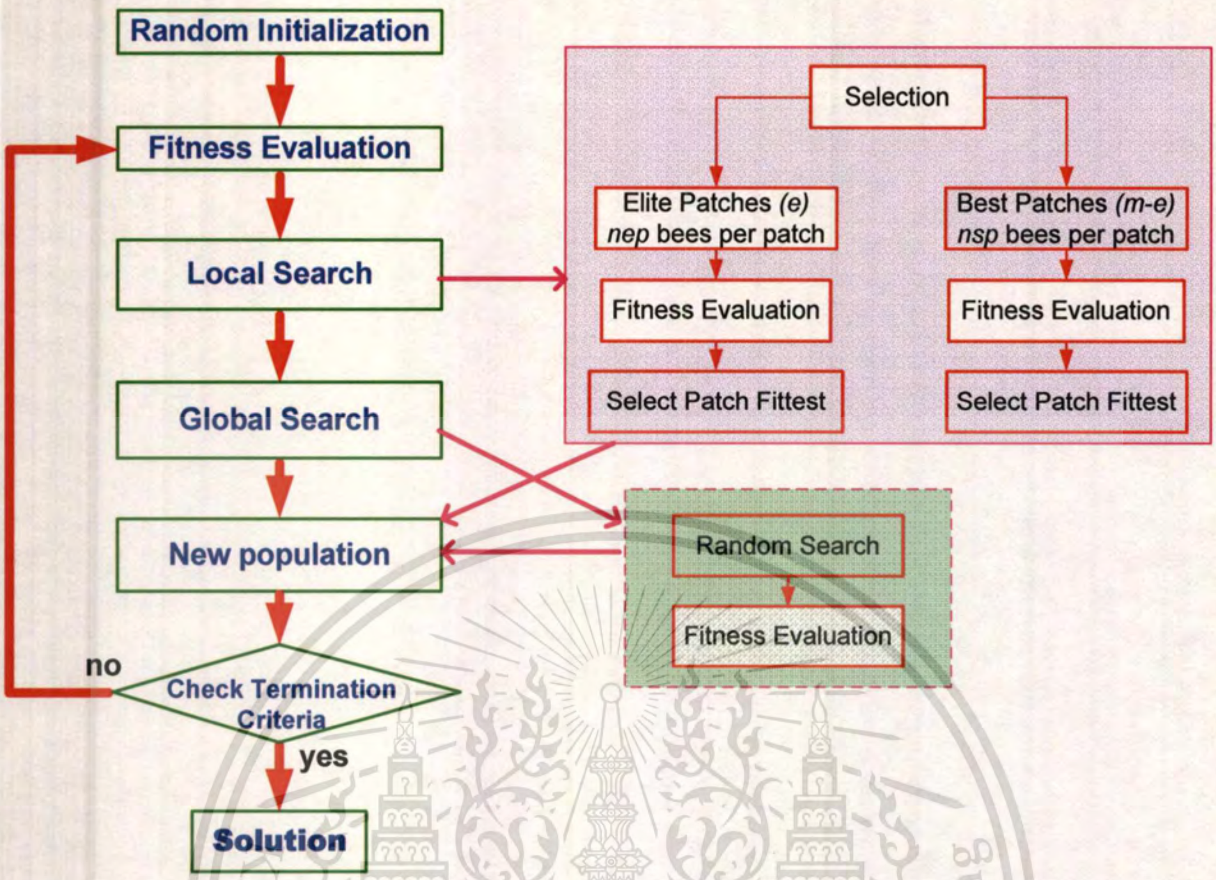


Figure 2.10 Flow chart of the basic bees algorithm

The pseudo code of the basic Bees Algorithm [66-67] is shown:

Step 1: Initialize population with random solutions.

Step 2: Evaluate fitness of the population.

Step 3: While stopping criterion not met, forming new bee population.

Step 4: Select elite bees.

Step 5: Select sites for neighborhood search.

Step 6: Recruit bees around selected sites and evaluate fitness.

Step 7: Select the fittest bee from each site.

Step 8: Assign remaining bees to search randomly and evaluate their fitness.

Step 9: End While.

In step 1, the algorithm starts by initializing n scout bees randomly. The sites visited by the scout bees are evaluated using a fitness function in step 2. In steps 4 and 5 and based on the fitness value, good bees are selected and the sites visited by them are chosen for neighborhood search. In step 6, more bees will be assigned for the chosen good sites. The best bee in each patch is chosen to form the next bee population in step 7. In step 8, the remaining bees in the population are assigned randomly searching for new potential solutions. These steps are repeated until the stopping criterion is met.

2.6 Hybrid Distributed Sobol Particle Swarm Optimization-Tabu Search Algorithm

In this thesis, a new hybrid technique based on PSO and TSA is proposed. This proposed approach is called distributed Sobol PSO and TSA (DSPSO-TSA). Three additional mechanism, i.e. Sobol sequence, distributed processing, and hybrid PSO-TSA will be presented. These mechanisms are described in this section.

2.6.1 Sobol Sequences

Sobol sequences (Sob) has introduced by Sobol I.M. [26], the improved Sobol sequences version was proposed by Antonov and Saleev [27]. The Sob generates a sequence of numbers between zero and one over an S - dimensional input space, in which successive points at any stage are filled in the gaps between the previously generated points [28]. Each different sequence of Sob is based on different primitive polynomials. Suppose P is a primitive polynomial of degree q of the form,

$$P = x^q + a_1 \times x^{q-1} + a_2 \times x^{q-2} + \ldots + a_{q-1} \times x + x^0 \qquad (2.16)$$

where the coefficients $(a_1,\ a_2, \ldots, a_{q-1})$ of the primitive polynomial are either 0 or 1, and the coefficient of x^q and of x^0 are unity. If a sequence of integers M_i is defined by the recurrence relationship

$$M_i = 2a_1 M_{i-1} \hat{} 2^2 a_2 M_{i-2} \hat{} \ldots \hat{} 2^{q-1} a_{q-1} M_{i-q+1} \hat{} 2^q M_{i-q} \hat{} M_{i-q}, \quad i > q \qquad (2.17)$$

where ^ is denotes a bit-by-bit exclusive-or (XOR) operation, then using a primitive polynomial of degree q. The value of $M_1, M_2, ..., M_q$ can be chosen freely provided that M_i is an odd integer less than $2, 2^2, ..., 2^q$ respectively. As a consequence, the value of $M_{q+1}, M_{q+2}, ...$ are then determined by the recurrence. A direction number (V_i) can be defined as

$$V_i = \frac{M_i}{2^i} \tag{2.18}$$

In [26], instead of generating the sequence z_j, $j = 1,2,...,j$ using the direction number(s) represented by the binary of j, one could use the direction number(s) represented by the Gray code of j. As a consequence, the z_{j+1} Sobol-Antonov Saleev sequences' can be obtained from the z_j by XORing with a single V_i, where i is the position of rightmost zero bit in j [27]. This makes the calculation of the sequence very efficient. Jiriwibhakorn S and et el [29-30] applied Sob for solving the critical clearing time of the fault conditions to enhance the power system transient stability, which application was applied to the selection of the training patterns of the critical clearing time under variation of load levels, fault locations and network structures, depicted that the Sob can be generated with confidence and an adequate range of input variables will be covered and training time can be reduced. Figure 2.11 shown two dimensions of 500 sequences generated by using Sob. Figure 2.12 shown two dimension values of 500 sequences obtained from random sequences (Rand). Sob has provided a good distribution of the input data better than Rand as shown.

The proposed technique for improvement solution quality generates the inertia factor with Sobol sequences as described above. The classical PSO has been generated the inertia factor by using Sobol sequences is called Sobol particle swarm optimization.

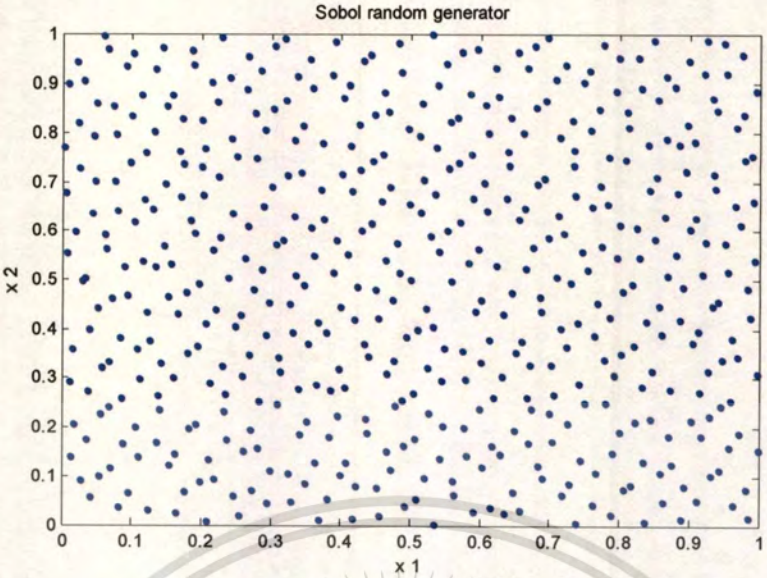


Figure 2.11 The 500 samplings by using the Sobol generator (Sob)

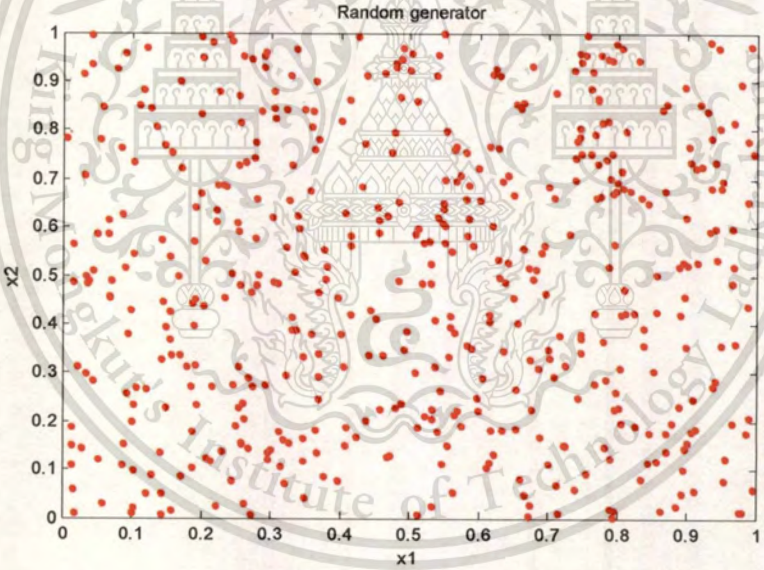


Figure 2.12 The 500 samplings by using random generator (Rand)

2.6.2 Distributed Sobol Particle Swarm Optimization (DSPSO)

The velocity mechanism of the conventional PSO is calculated by using either equation 12 or 13. Then the position of particle is modified by using equation (14). The velocity updating of conventional PSO relies on three influence components: its current particle (x_i), the best particle (*pbest*) and the best particle of among all the particles in the group ($gbest$). The particles fly around the multidimensional search space until they found the plentiful region (optimal solution).

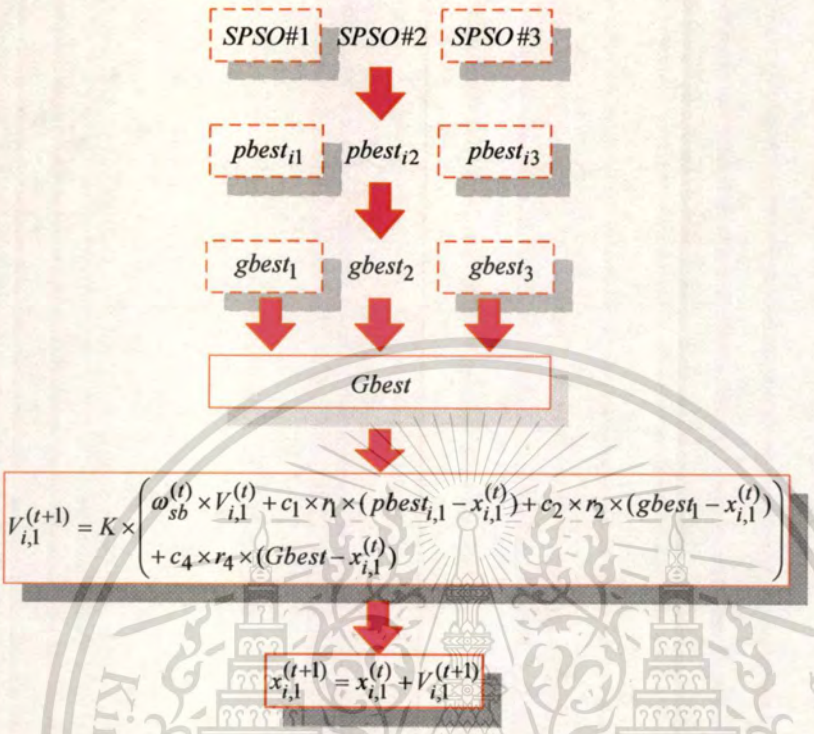


Figure 2.13 Flowchart of DSPSO procedure

This thesis introduces the new PSO process which splits the group of particles to many subgroups. Each subgroup has a memory which stores its current particle, the best particle and the best particle of among all the particles. All subgroups will use these memories together in order to look into the best particle among all groups ($Gbest$). Therefore, a new velocity equation has two cognitive behaviors and two cognitive socials. These cognitive behaviors are represented by the best particle in the group and the best particle neighbor group. The cognitive socials are modeled by the best particle among all the particles in the group and the best particle among all the particles in all groups. The new velocity equation developed by the proposed technique is expressed as follows:

$$V_i^{(t+1)} = K \times \begin{pmatrix} \omega_{sb} \times V_i^{(t)} + c_1 \times r_1 \times (pbest_i - x_i^{(t)}) + c_2 \times r_2 \times (gbest - x_i^{(t)}) \\ + c_4 \times r_4 \times (Gbest - x_i^{(t)}) \end{pmatrix} \tag{2.19}$$

where ω_{sb} is the Sobol inertia factor, *Gbest* is the best particles among all groups, r_4 is random number between 0 and 1, and c_4 is the accelerator factor.

The technique distributes the PSO to many groups with Sobol inertia factor is called Sobol distributed particle swarm optimization (DSPSO). The velocities and position updating of all the groups are executed by using equations (2.19) and (2.6), respectively. The procedure of the three groups of the DSPSO and the process to gain the *Gbest* for the new velocity equation are shown in Figure 2.13.

2.6.3 DSPSO-TSA process for solving optimization problems

The proposed DSPSO is applied to solve the optimization problem considering generator constraints, nonsmooth cost function, and noncontinuos cost function. The DSPSO process splits the group of particles to several subgroups. Each subgroup has a memory which stores its current particle in the subgroup p^{th} $(x_{i,p})$, the best particle in the subgroup p^{th} $(pbest_{i,p})$ and the best particle of among all the particles in the subgroup p^{th} $(gbest_{i,p})$. All subgroups will use these memories together in order to look into the best particle among all groups ($Gbest$). The inertia factor is created by the Sobol random generator and compared with the conventional inertia factor, as presented in Figure 2.14.

Detail of the DSPSO which is applied to solve the optimization problem is described by the following steps.

Step 1. Set these parameters c_1, c_2, c_3, c_4. Generate the inertia factor using Sobol random generator. Set the first iteration $t = 0$ and determine the number of subgroup (p).

Step 2. Initialize the positions and velocities of the particles of each subgroup.

Step 3. Calculate the objective function of each particle

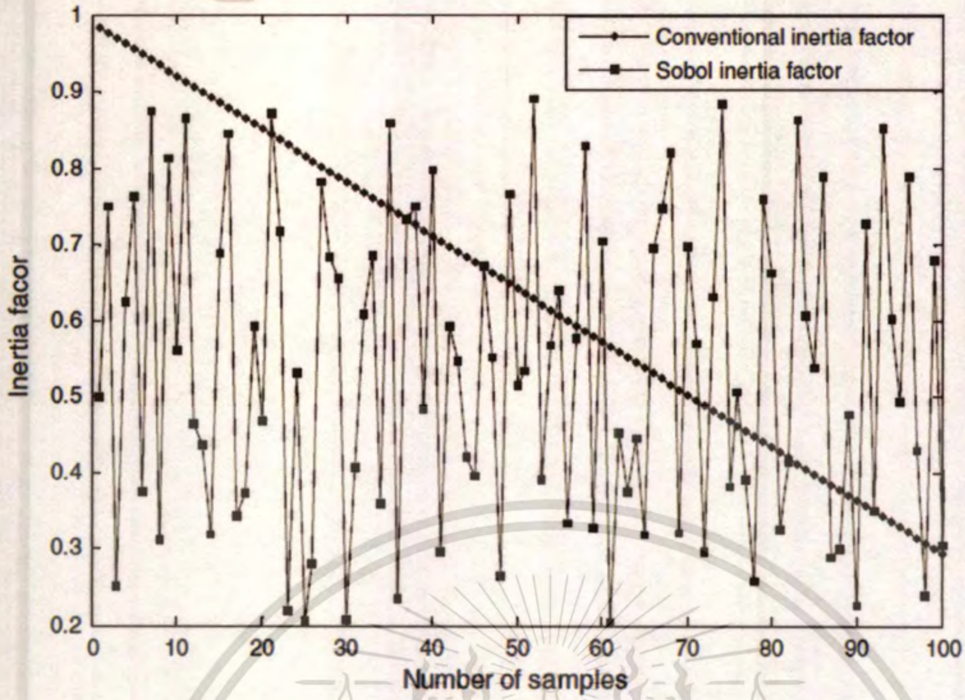


Figure 2.14 The 100 samples of Sobol inertia factor for DSPSO-TSA.

Step 4. The best particle of the each subgroup will be represented as $pbest_{i,p}$. The best particle among all particles of each subgroup will be set as $gbest_p$. Finally, the best particle among all group will be set as $Gbest$.

Step 5. Calculate the velocity of all the particles is given in equation (2.19).

Step 6. Check the velocity limits, if a particle exceed the velocity limits that is $V_i^{t+1} > V_i^{\max}$ then set $V_i^{t+1} = 0.5 \times P_i^U$, otherwise set $V_i^{t+1} = -0.5 \times P_i^L$.

Step 7. Update the position of all particles using equation (2.5), the updating position must be satisfied all the constraints.

Step 8. Adjust the position of the particle which violated the constraint. The new velocity must be adjusted by using equation (2.20), and modified the new position by using equation (2.5)

$$V_{i,p}^{(t+1)} = K \times \begin{pmatrix} \omega_{sb} \times V_{i,p}^{(t)} + c_1 \times r_1 \times (pbest_{i,p} - x_{i,p}^{(t)}) + c_2 \times r_2 \times (gbest_{i,p} - x_{i,p}^{(t)}) \\ + c_3 \times r_3 \times (pbest_{i,j} - x_{i,p}^{(t)}) + c_4 \times r_4 \times (Gbest - x_{i,p}^{(t)}) \end{pmatrix} \tag{2.20}$$

where $pbest_{i,j}$ is the best position of the particle in the subgroup j, r_3 is random number between 0 and 1, c_3 is accelerator factor.

Step 9. Check stopping criterion. If satisfy the stopping criterion go to step 10, otherwise, increase iteration $t = t + 1$ and go back to step 3.

Step 10. Apply the TSA, the particles of the *Gbest* which obtained by DSPSO in the latest iteration will be defined as an initial solution of TSA which described in the section 2.2.

In order to show the effectiveness, the proposed method is tested to solve four mathematic benchmark functions, economic dispatch problem and optimal power problem with various case studies. The simulation results are shown in next chapters.

Chapter 3
DSPSO-TSA Application

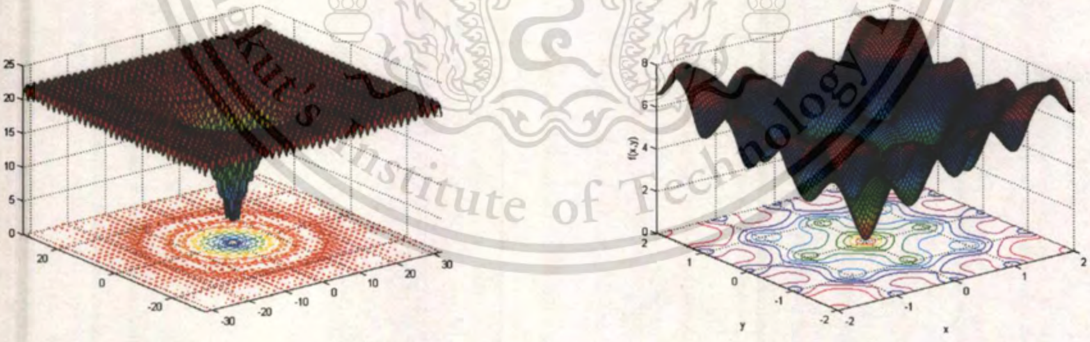
To illustrate the performance and effectiveness of the DSPSO-TSA algorithm for solving the optimization problem, four famous benchmark functions [68] are employed for comparison, comprising PSO, BA, DE, GA and TSA. Many researchers tested algorithm using them widely, that are the non-linear continuous mathematical functions. The following subsections describe the characteristics of the tested functions.

3.1 Tested Functions

3.1.1 Ackley's function

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + \exp \tag{3.1}$$

which has range between $-32 < x_i < 32$



(a) (b)

Fig. 3.1. Two-dimension of Ackley's function; (a) surface plot in search space range from -32 to 32; (b) zoom in an area of the global optimum from -2 to 2

Ackley's function is a widely used multimodal test function. The global minimum is $f(x)=0, x_i=0, i=1,2,....n$, Ackley's function has thousands of minima in the region and very difficult to be optimized. Fig. 3.1 shows the function at two different zoom ratios. The graphic on the left side employs the whole definition area of the function from -32 to 32. The graphic on the right side is zoomed into the area of the global minimum give a better impression of the properties of the function.

3.1.2 Griewank's function

$$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1, \qquad -600<x_i<600 \tag{3.2}$$

(b) (b)

Fig. 3.2. Demonstration of Griewangk's function; (a) Full definition area from -500 to 500, (b) Reduced area of the function from -100 to 100

Griewangk's function has many widespread local minima. However, the locations of the minima are regularly distributed [69]. The global minimum is at *f(x)=0; x(i)=0, i*=1…*n.* The graphics of Griewangk's function in all search spaces range are demonstrated in Fig. 3.2. The graphic on the left side shows the full definition range of the function. When approaching the inner area, the function looks different. Many small peaks and valleys are visible in the right graphic. The term of summation produce a parabola, while the local optima are above parabola level. The dimension of the search range increase on the basis of the product term, which results in the decrease of local optimums [70].

3.1.3 Rastrigin's function

$$f(x) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right], \qquad -5.12 < x_i < 5.12 \qquad (3.3)$$

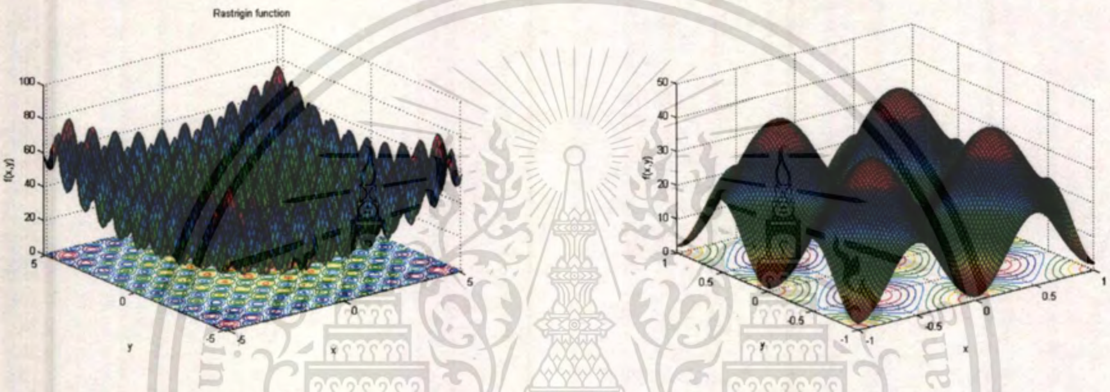


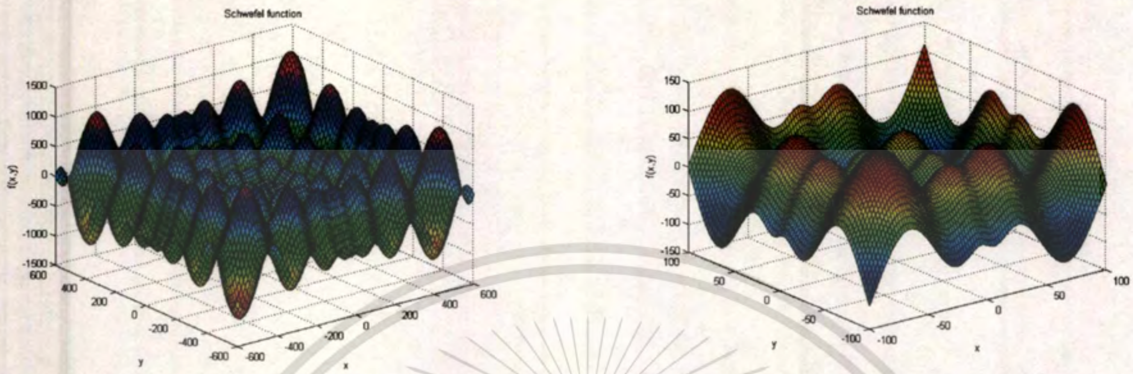
Fig. 3.3. Rastrigin's function; (a) surface plot in full range, (b) zoom around the area of the global optimum in an area from -1 to 1

Rastrigin's function is the test function that highly multimodal and non-linear [71]. The locations of the minima are regularly distributed. The global minimum is at *f(x)=0; x(i)=0, i*=1…*n.* It can be shown in Fig. 5.3.

3.1.4 Schwefel's function

$$f(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} x_j\right)^2, \qquad -600 < x_i < 600 \qquad (3.4)$$

Schwefel's function is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction. Fig. 3.4 shows its graph with full ranges and reduced ranges.



(d) (b)

Fig. 3.4. (a) Surface plot of Schwefel's function with full ranges; (b) Surface plot of Schwefel's function with reduced ranges.

It can be grouped as multimodal function where the local optimum increases exponentially with the problem dimensions.

3.2 Experimental Setup

The DSPSO-TSA, BA, DE, GA, PSO and TSA are implemented in MATLAB language and executed on an Intel(R) Core2 Duo 3.0 GHz personal computer with a 4.0 GB of RAM. The parameters of each algorithm have to be defined before solve the non-linear mathematic functions. The results of each method are compared with the proposed method. The parameters of each method describe as following:

Bee Algorithm's parameters

- Number of the initialized bees = 100
- Number of selected sites = 3
- Number of elite sites = 7

- Number of bees for selected site = 10
- Number of bees for elite site = 15

Differential Evolution algorithm's parameters

- Number of populations = 100
- Scaling factor = 1
- Crossover rate = 0.5

Genetic Algorithm's parameters

- Number of populations = 100
- Number of precision bit = 14
- Generation gap = 0.9
- Crossover rate = 0.7 (Roulette wheel selection)
- Mutation rate = 0.5

Particle Swarm Optimization's parameters

- Number of particles = 100
- Maximum inertia factor = 0.9 ($\omega_{\max}$)
- Minimum inertia factor = 0.1 ($\omega_{\min}$)
- Accelerating factors = 2.00 (c_1, c_2)
- Maximum velocity is $0.5 \times$ *search space dimension*
- Minimum velocity are $-0.5 \times$ *search space dimension*

Tabu Search Algorithm's parameters

- Number of neighborhoods size = $2 \times$ *problem dimension*
- Tabu list size = *neighborhood size* $\times 3$
- Restriction period = 50
- Frequently period = 25

DSPSO-TSA's parameters

- Number of groups = 3
- Number of particles for each group = 20
- Accelerating factors = 2.05 (c_1, c_2)
- Accelerating factors, $c_3 = 1.35, c_4 = 0.70$
- Maximum velocity is $0.5 \times$ *search space dimension*
- Minimum velocity are $-0.5 \times$ *search space dimension*
- The parameters of the TSA are same as above

3.3 Experimental Results and Comparison

The experiment results consist of the mean value, standard deviation of the solutions and computational time are found in 100 runs of each algorithm.

Table 3.1

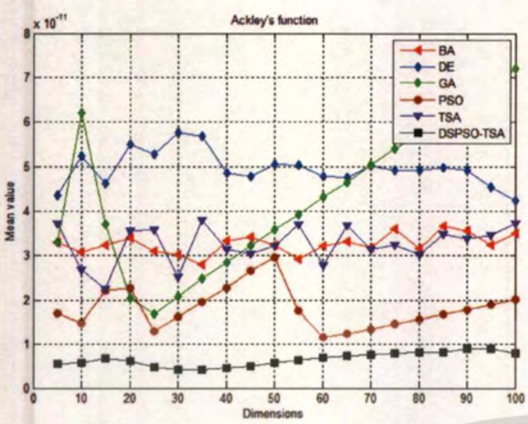
Simulation results obtained from six methods for Ackley's function

Problem dimension	BA			DE			GA		
	Mean. f(xi,yi)	Mean. time (s)	Standard Deviation	Mean. f(xi,yi)	Mean. time (s)	Standard Deviation	Mean. f(xi,yi)	Mean. time (s)	Standard Deviation
5	3.308E-11	0.20	2.663E-11	4.367E-11	0.12	2.688E-11	3.295E-11	1.36	1.488E-11
10	3.069E-11	0.31	2.602E-11	5.243E-11	0.23	2.972E-11	6.206E-11	1.46	1.747E-11
15	3.237E-11	0.42	2.218E-11	4.623E-11	0.35	2.889E-11	3.694E-11	1.54	3.241E-11
20	3.397E-11	0.52	2.773E-11	5.513E-11	0.51	2.756E-11	2.035E-11	1.69	2.186E-11
25	3.104E-11	0.65	2.321E-11	5.290E-11	0.63	3.022E-11	1.679E-11	1.75	2.452E-12
30	3.021E-11	0.74	2.280E-11	5.763E-11	0.82	2.815E-11	2.068E-11	1.85	2.610E-12
35	2.806E-11	0.84	2.347E-11	5.684E-11	0.99	2.773E-11	2.478E-11	1.93	2.900E-12
40	3.348E-11	0.91	2.585E-11	4.864E-11	1.22	2.921E-11	2.842E-11	2.14	2.955E-12
45	3.418E-11	1.02	2.597E-11	4.783E-11	1.44	2.736E-11	3.211E-11	2.23	3.166E-12
50	3.242E-11	1.15	2.682E-11	5.071E-11	1.53	2.793E-11	3.572E-11	2.24	3.518E-12
55	2.910E-11	1.29	2.434E-11	5.037E-11	1.84	3.203E-11	3.918E-11	2.36	3.743E-12
60	3.211E-11	1.34	2.177E-11	4.784E-11	2.34	2.616E-11	4.320E-11	2.39	3.519E-12
65	3.323E-11	1.46	2.417E-11	4.755E-11	2.89	2.914E-11	4.646E-11	2.46	3.414E-12
70	3.179E-11	1.56	2.205E-11	5.024E-11	3.58	2.847E-11	5.046E-11	2.61	4.260E-12
75	3.594E-11	1.79	2.696E-11	4.932E-11	3.73	2.949E-11	5.403E-11	2.71	4.106E-12
80	3.162E-11	1.96	2.321E-11	4.930E-11	4.29	2.827E-11	5.734E-11	2.80	4.234E-12

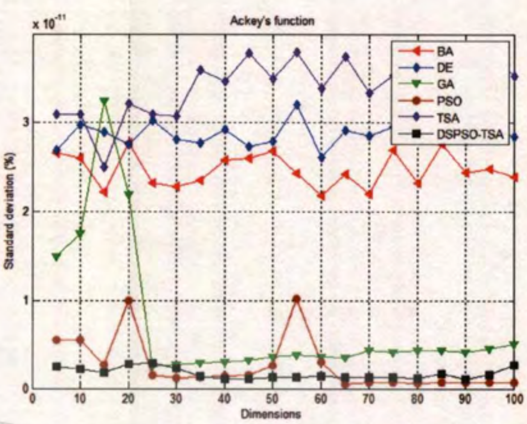
Table 3.1
Simulation results obtained from six methods for Ackley's function (cont.')

Problem dimension	BA			DE			GA		
	Mean. f(xi,yi)	Mean. time (s)	Standard Deviation	Mean. f(xi,yi)	Mean. time (s)	Standard Deviation	Mean. f(xi,yi)	Mean. time (s)	Standard Deviation
85	3.667E-11	2.07	2.763E-11	4.974E-11	4.77	3.069E-11	6.098E-11	2.94	4.239E-12
90	3.564E-11	2.19	2.436E-11	4.916E-11	5.25	2.787E-11	6.506E-11	3.07	4.043E-12
95	3.231E-11	2.39	2.481E-11	4.540E-11	5.84	2.950E-11	6.814E-11	3.13	4.426E-12
100	3.493E-11	2.47	2.395E-11	4.249E-11	6.52	2.839E-11	7.214E-11	3.23	4.945E-12

Problem dimension	PSO			TSA			DSPSO-TSA		
	f(xi,yi) Mean.	time (s) Mean.	Standard Deviation	f(xi,yi) Mean.	time (s) Mean.	Standard Deviation	f(xi,yi) Mean.	time (s) Mean.	Standard Deviation
5	1.686E-11	0.13	5.469E-12	3.726E-11	0.03	3.096E-11	5.508E-12	0.10	2.481E-12
10	1.484E-11	0.20	5.459E-12	2.676E-11	0.05	3.094E-11	5.626E-12	0.18	2.184E-12
15	2.212E-11	0.28	2.661E-12	2.261E-11	0.15	2.499E-11	6.626E-12	0.25	1.740E-12
20	2.277E-11	0.35	9.942E-12	3.549E-11	0.17	3.211E-11	6.196E-12	0.31	2.807E-12
25	1.291E-11	0.44	1.442E-12	3.569E-11	0.58	3.093E-11	4.709E-12	0.37	2.905E-12
30	1.612E-11	0.52	1.150E-12	2.515E-11	0.95	3.075E-11	4.107E-12	0.42	2.221E-12
35	1.948E-11	0.61	1.235E-12	3.790E-11	1.57	3.597E-11	4.057E-12	0.49	1.385E-12
40	2.280E-11	0.70	1.340E-12	3.141E-11	2.99	3.461E-11	4.468E-12	0.55	1.032E-12
45	2.650E-11	0.79	1.468E-12	3.028E-11	3.26	3.788E-11	4.913E-12	0.60	1.091E-12
50	2.957E-11	0.88	2.544E-12	3.219E-11	3.96	3.493E-11	5.755E-12	0.66	1.231E-12
55	1.746E-11	0.99	1.020E-11	3.691E-11	4.03	3.790E-11	6.288E-12	0.71	1.223E-12
60	1.162E-11	1.10	2.928E-12	2.785E-11	4.28	3.383E-11	6.881E-12	0.78	1.319E-12
65	1.223E-11	1.18	4.989E-13	3.680E-11	5.17	3.742E-11	7.300E-12	0.84	1.246E-12
70	1.335E-11	1.27	6.032E-13	3.131E-11	5.89	3.329E-11	7.612E-12	0.91	1.267E-12
75	1.446E-11	1.39	6.130E-13	3.244E-11	6.00	3.534E-11	7.927E-12	0.97	1.295E-12
80	1.563E-11	1.47	5.863E-13	3.026E-11	6.52	3.442E-11	8.146E-12	1.03	1.197E-12
85	1.672E-11	1.57	6.140E-13	3.471E-11	6.83	3.653E-11	8.184E-12	1.13	1.630E-12
90	1.774E-11	1.67	6.174E-13	3.381E-11	7.05	3.652E-11	8.851E-12	1.16	1.099E-12
95	1.890E-11	1.79	6.623E-13	3.453E-11	7.59	3.820E-11	8.890E-12	1.25	1.598E-12
100	2.016E-11	1.90	6.746E-13	3.715E-11	8.39	3.527E-11	7.923E-12	1.43	2.675E-12



(a)



(b)

Fig. 3.5. (a) Mean value of function vs. problem dimension; (b) Standard deviation of solution vs. problem dimension

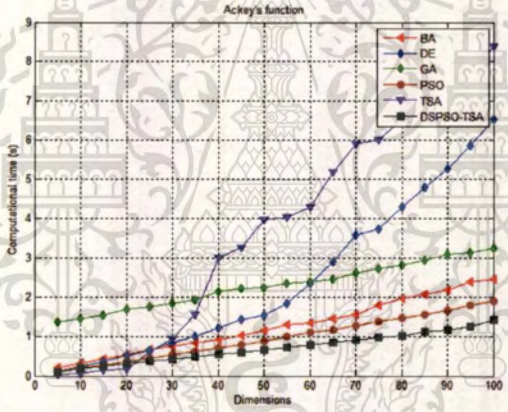


Fig. 3.6. Computational time vs. problem dimension

Table 3.2

Simulation results obtained from six methods for Rastrigin's function

Problem dimension	BA			DE			GA		
	f(xi,yi) Mean.	time (s) Mean.	Standard Deviation	f(xi,yi) Mean.	time (s) Mean.	Standard Deviation	f(xi,yi) Mean.	time (s) Mean.	Standard Deviation
5	3.065E-11	0.80	3.171E-11	5.638E-11	0.19	3.233E-11	4.312E-11	0.66	3.204E-11
10	3.366E-11	0.94	2.938E-11	5.714E-11	0.31	2.264E-11	2.331E-11	0.72	3.621E-11
15	3.113E-11	1.01	2.763E-11	5.435E-12	0.41	2.435E-11	4.738E-12	0.76	1.364E-11
20	3.195E-11	1.08	3.075E-11	5.504E-11	0.42	2.497E-11	3.045E-12	0.78	3.451E-11
25	3.188E-11	1.14	3.104E-11	6.149E-11	0.31	1.824E-11	3.965E-12	0.80	3.374E-11
30	4.237E-11	1.19	3.243E-11	3.465E-11	0.61	2.440E-11	4.612E-12	0.83	2.022E-11
35	3.591E-11	1.28	3.038E-11	9.234E-11	0.76	2.580E-11	5.752E-12	0.85	2.247E-11

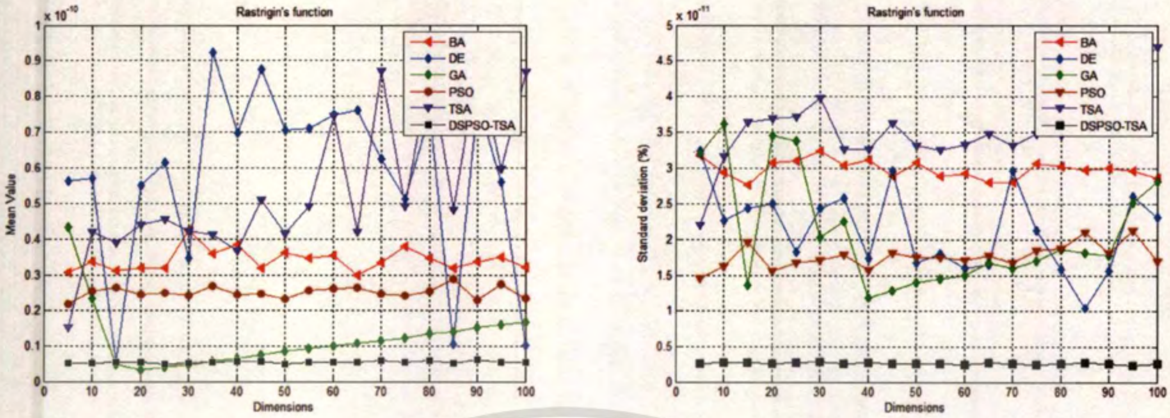
Table 3.2

Simulation results obtained from six methods for Rastrigin's function (cont')

Problem dimension	BA f(xi,yi) Mean.	BA time (s) Mean.	BA Standard Deviation	DE f(xi,yi) Mean.	DE time (s) Mean.	DE Standard Deviation	GA f(xi,yi) Mean.	GA time (s) Mean.	GA Standard Deviation
40	3.859E-11	1.26	3.109E-11	6.970E-11	0.63	1.737E-11	6.456E-12	0.88	1.178E-11
45	3.202E-11	1.25	2.893E-11	8.754E-11	0.93	2.958E-11	7.410E-12	0.92	1.285E-11
50	3.634E-11	1.25	3.078E-11	7.044E-11	0.66	1.669E-11	8.404E-12	0.93	1.402E-11
55	3.479E-11	1.45	2.891E-11	7.108E-11	0.45	1.795E-11	9.097E-12	0.95	1.450E-11
60	3.551E-11	1.51	2.932E-11	7.486E-11	0.64	1.600E-11	9.993E-12	0.98	1.492E-11
65	2.996E-11	1.49	2.796E-11	7.594E-11	0.68	1.644E-11	1.068E-11	0.99	1.676E-11
70	3.354E-11	1.57	2.807E-11	6.241E-11	1.09	2.962E-11	1.147E-11	1.03	1.600E-11
75	3.793E-11	1.48	3.064E-11	5.137E-11	0.94	2.127E-11	1.208E-11	1.05	1.698E-11
80	3.460E-11	1.56	3.029E-11	7.517E-11	1.11	1.588E-11	1.343E-11	1.09	1.860E-11
85	3.199E-11	1.61	2.972E-11	1.029E-11	1.37	1.028E-11	1.398E-11	1.17	1.808E-11
90	3.371E-11	1.72	3.001E-11	7.845E-11	1.18	1.557E-11	1.516E-11	1.21	1.769E-11
95	3.494E-11	1.63	2.959E-11	5.595E-11	1.29	2.595E-11	1.588E-11	1.23	2.516E-11
100	3.229E-11	1.68	2.859E-11	1.010E-11	1.54	2.314E-11	1.677E-11	1.26	2.812E-11

Problem dimension	PSO f(xi,yi) Mean.	PSO time (s) Mean.	PSO Standard Deviation	TSA f(xi,yi) Mean.	TSA time (s) Mean.	TSA Standard Deviation	DSPSO-TSA f(xi,yi) Mean.	DSPSO-TSA time (s) Mean.	DSPSO-TSA Standard Deviation
5	2.18E-11	0.13	1.46E-11	1.524E-11	0.03	2.205E-11	5.2E-12	0.09	2.6E-12
10	2.55E-11	0.19	1.62E-11	4.207E-11	0.06	3.166E-11	5.14E-12	0.11	2.67E-12
15	2.65E-11	0.27	1.96E-11	3.897E-11	0.12	3.640E-11	5.34E-12	0.12	2.72E-12
20	2.45E-11	0.34	1.56E-11	4.410E-11	0.19	3.686E-11	5.46E-12	0.13	2.56E-12
25	2.5E-11	0.40	1.67E-11	4.544E-11	0.28	3.710E-11	4.96E-12	0.14	2.68E-12
30	2.41E-11	0.49	1.71E-11	4.231E-11	0.38	3.973E-11	5.12E-12	0.16	2.78E-12
35	2.7E-11	0.60	1.78E-11	4.131E-11	0.52	3.259E-11	5.39E-12	0.17	2.56E-12
40	2.45E-11	0.67	1.57E-11	3.662E-11	0.68	3.263E-11	5.52E-12	0.19	2.63E-12
45	2.47E-11	0.81	1.81E-11	5.099E-11	0.84	3.628E-11	5.52E-12	0.20	2.51E-12
50	2.32E-11	0.92	1.75E-11	4.136E-11	1.02	3.311E-11	4.99E-12	0.21	2.56E-12
55	2.57E-11	1.04	1.75E-11	4.925E-11	1.26	3.249E-11	5.46E-12	0.23	2.57E-12
60	2.62E-11	1.18	1.71E-11	7.464E-11	1.45	3.326E-11	5.4E-12	0.25	2.4E-12
65	2.65E-11	1.21	1.77E-11	4.189E-11	1.79	3.473E-11	5.37E-12	0.26	2.66E-12
70	2.47E-11	1.24	1.67E-11	8.709E-11	1.99	3.313E-11	5.78E-12	0.28	2.55E-12
75	2.42E-11	1.37	1.85E-11	4.926E-11	2.34	3.476E-11	5.49E-12	0.31	2.5E-12
80	2.55E-11	1.45	1.87E-11	8.834E-11	2.57	3.468E-11	5.89E-12	0.33	2.54E-12
85	2.89E-11	1.47	2.1E-11	4.823E-11	2.95	3.535E-11	5.25E-12	0.36	2.71E-12

Problem dimension	PSO f(xi,yi) Mean.	PSO time (s) Mean.	PSO Standard Deviation	TSA f(xi,yi) Mean.	TSA time (s) Mean.	TSA Standard Deviation	DSPSO-TSA f(xi,yi) Mean.	DSPSO-TSA time (s) Mean.	DSPSO-TSA Standard Deviation
90	2.3E-11	1.59	1.81E-11	8.937E-11	3.32	3.744E-11	6.05E-12	0.40	2.52E-12
95	2.73E-11	1.70	2.12E-11	5.986E-11	3.66	4.436E-11	5.35E-12	0.43	2.33E-12
100	2.35E-11	1.83	1.7E-11	8.672E-11	4.02	4.687E-11	5.41E-12	0.46	2.54E-12



(a) (b)

Fig.3.7. (a) Mean value of function vs. problem dimension of Rastrigin's function; (b) Standard deviation of solution vs. problem dimension of six methods compared

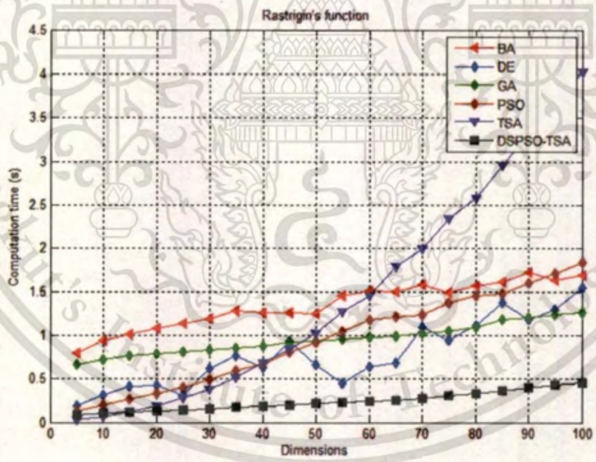


Fig. 3.8. Computational time vs. problem dimension of six methods

Table 3.3

Simulation results obtained from six methods for Griewank's function

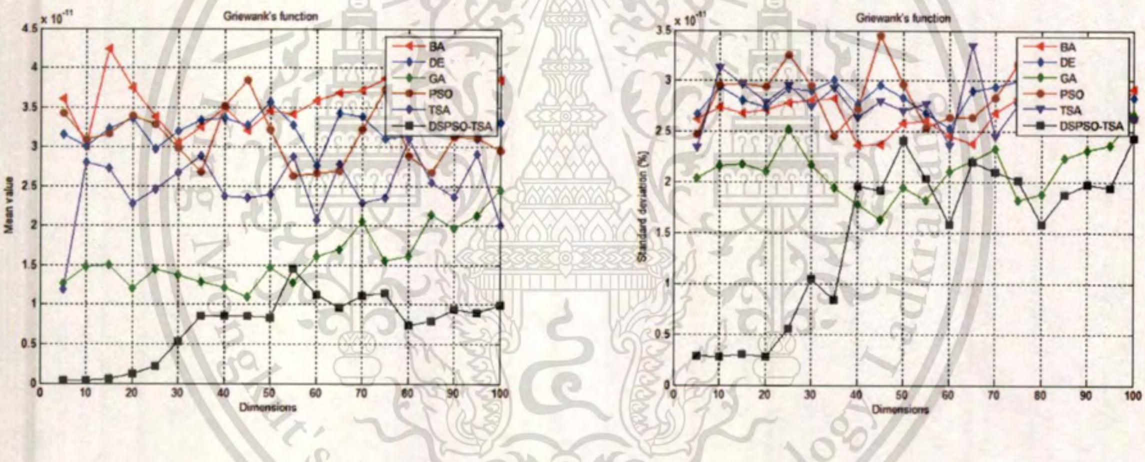
Problem dimension	BA f(xi,yi) Mean.	BA time (s) Mean.	BA Standard Deviation	DE f(xi,yi) Mean.	DE time (s) Mean.	DE Standard Deviation	GA f(xi,yi) Mean.	GA time (s) Mean.	GA Standard Deviation
5	3.606E-11	0.19	2.619E-11	3.153E-11	0.21	2.672E-11	1.257E-11	0.65	2.035E-11
10	3.012E-11	0.27	2.737E-11	2.996E-11	0.29	2.934E-11	1.477E-11	0.66	2.156E-11
15	4.232E-11	0.33	2.680E-11	3.227E-11	0.34	2.800E-11	1.495E-11	0.70	2.175E-11
20	3.745E-11	0.40	2.708E-11	3.358E-11	0.41	2.736E-11	1.195E-11	0.74	2.101E-11
25	3.373E-11	0.47	2.777E-11	2.972E-11	0.48	2.919E-11	1.441E-11	0.77	2.516E-11
30	3.042E-11	0.52	2.799E-11	3.195E-11	0.55	2.896E-11	1.372E-11	0.81	2.169E-11
35	3.247E-11	0.61	2.818E-11	3.330E-11	0.61	3.001E-11	1.284E-11	0.86	1.940E-11
40	3.494E-11	0.68	2.362E-11	3.352E-11	0.68	2.761E-11	1.206E-11	0.88	1.779E-11
45	3.201E-11	0.78	2.370E-11	3.273E-11	0.74	2.955E-11	1.086E-11	0.91	1.624E-11
50	3.457E-11	0.89	2.573E-11	3.555E-11	0.82	2.830E-11	1.466E-11	0.95	1.940E-11
55	3.407E-11	0.96	2.600E-11	3.270E-11	0.89	2.673E-11	1.278E-11	0.97	1.815E-11
60	3.577E-11	1.03	2.455E-11	2.758E-11	0.98	2.528E-11	1.608E-11	1.00	2.097E-11
65	3.680E-11	1.08	2.372E-11	3.418E-11	1.04	2.897E-11	1.692E-11	1.06	2.207E-11
70	3.711E-11	1.15	2.676E-11	3.376E-11	1.11	2.933E-11	2.041E-11	1.10	2.330E-11
75	3.857E-11	1.27	2.819E-11	3.100E-11	1.18	2.995E-11	1.552E-11	1.15	1.815E-11
80	4.071E-11	1.33	2.615E-11	3.132E-11	1.26	2.932E-11	1.606E-11	1.14	1.877E-11
85	3.399E-11	1.45	2.643E-11	3.278E-11	1.33	2.794E-11	2.137E-11	1.25	2.234E-11
90	3.405E-11	1.56	2.585E-11	3.437E-11	1.43	3.029E-11	1.961E-11	1.32	2.309E-11
95	4.246E-11	1.62	2.601E-11	3.100E-11	1.49	2.815E-11	2.124E-11	1.34	2.363E-11
100	3.840E-11	1.73	2.907E-11	3.307E-11	1.58	2.829E-11	2.445E-11	1.40	2.662E-11

Problem dimension	PSO f(xi,yi) Mean.	PSO time (s) Mean.	PSO Standard Deviation	TSA f(xi,yi) Mean.	TSA time (s) Mean.	TSA Standard Deviation	DSPSO-TSA f(xi,yi) Mean.	DSPSO-TSA time (s) Mean.	DSPSO-TSA Standard Deviation
5	3.422E-11	0.10	2.471E-11	1.185E-11	0.04	2.341E-11	4.006E-12	0.10	2.886E-12
10	3.091E-11	0.16	2.953E-11	2.794E-11	0.11	3.131E-11	3.801E-12	0.11	2.819E-12
15	3.167E-11	0.23	2.966E-11	2.725E-11	0.21	2.961E-11	3.744E-12	0.14	3.013E-12
20	3.385E-11	0.28	2.934E-11	2.272E-11	0.30	2.783E-11	3.821E-12	0.18	2.830E-12
25	3.286E-11	0.35	3.255E-11	2.450E-11	0.44	2.957E-11	4.767E-12	0.22	5.543E-12
30	2.980E-11	0.42	2.943E-11	2.673E-11	0.60	2.747E-11	6.399E-12	0.25	1.047E-11
35	2.674E-11	0.54	2.453E-11	2.882E-11	0.82	2.931E-11	6.577E-12	0.30	8.392E-12
40	3.519E-11	0.60	2.722E-11	2.371E-11	1.10	2.633E-11	1.341E-11	0.32	1.956E-11
45	3.842E-11	0.70	3.447E-11	2.351E-11	1.33	2.798E-11	1.273E-11	0.37	1.917E-11
50	3.213E-11	0.79	2.959E-11	2.387E-11	1.63	2.700E-11	1.583E-11	0.40	2.411E-11
55	2.628E-11	0.86	2.539E-11	2.871E-11	1.87	2.769E-11	1.183E-11	0.43	2.036E-11

Table 3.3

Simulation results obtained from six methods for Griewank's function (cont'.)

Problem dimension	PSO f(xi,yi) Mean.	PSO time (s) Mean.	PSO Standard Deviation	TSA f(xi,yi) Mean.	TSA time (s) Mean.	TSA Standard Deviation	DSPSO-TSA f(xi,yi) Mean.	DSPSO-TSA time (s) Mean.	DSPSO-TSA Standard Deviation
60	2.655E-11	0.94	2.636E-11	2.063E-11	2.32	2.363E-11	9.800E-12	0.45	1.580E-11
65	2.693E-11	1.02	2.637E-11	2.772E-11	2.76	3.349E-11	1.194E-11	0.50	2.203E-11
70	3.215E-11	1.11	2.830E-11	2.282E-11	3.23	2.448E-11	1.166E-11	0.51	2.099E-11
75	3.723E-11	1.23	3.162E-11	2.342E-11	3.86	2.765E-11	1.099E-11	0.56	2.014E-11
80	2.887E-11	1.30	2.870E-11	3.104E-11	4.53	3.025E-11	8.188E-12	0.60	1.587E-11
85	2.664E-11	1.43	2.673E-11	2.535E-11	5.24	2.915E-11	9.969E-12	0.66	1.876E-11
90	3.120E-11	1.49	2.902E-11	2.355E-11	5.63	2.923E-11	1.028E-11	0.70	1.975E-11
95	3.112E-11	1.61	2.858E-11	2.903E-11	6.37	3.059E-11	1.069E-11	0.71	1.942E-11
100	2.947E-11	1.72	2.622E-11	2.000E-11	7.49	2.622E-11	1.432E-11	0.77	2.433E-11



(a) (b)

Fig. 3.9. (a) Mean value of function vs. problem dimension of Griewank's function; (b) Standard deviation of solution vs. problem dimension of Griewank's function

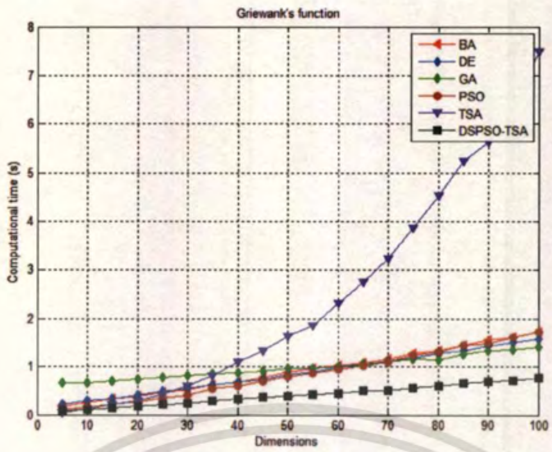


Fig. 3.10. Computational time vs. problem dimension of Griewank's function

Table 3.4

Simulation results obtained from six methods for Schwefel's function

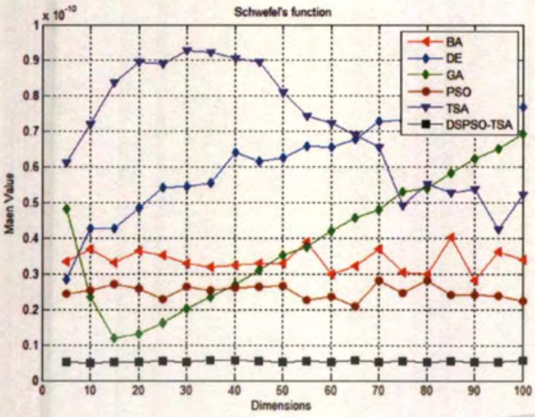
Problem dimension	BA f(xi,yi) Mean.	BA time (s) Mean.	BA Standard Deviation	DE f(xi,yi) Mean.	DE time (s) Mean.	DE Standard Deviation	GA f(xi,yi) Mean.	GA time (s) Mean.	GA Standard Deviation
5	3.341E-11	0.62	2.948E-11	2.84E-11	0.05	2.3E-11	4.83E-11	0.89	3.03E-11
10	3.687E-11	0.74	3.068E-11	4.28E-11	0.06	2.57E-11	2.35E-11	0.96	2.97E-11
15	3.321E-11	0.83	2.808E-11	4.27E-11	0.08	2.5E-11	1.19E-11	0.99	1.09E-11
20	3.652E-11	0.83	2.771E-11	4.86E-11	0.10	2.47E-11	1.32E-11	1.02	3.42E-12
25	3.509E-11	0.82	2.702E-11	5.42E-11	0.12	2.24E-11	1.61E-11	1.04	3.84E-12
30	3.294E-11	0.88	2.872E-11	5.46E-11	0.14	2.08E-11	2.01E-11	1.07	4.14E-12
35	3.199E-11	0.97	2.902E-11	5.55E-11	0.16	1.81E-11	2.35E-11	1.10	5.1E-12
40	3.252E-11	0.95	3.167E-11	6.41E-11	0.19	2.17E-11	2.7E-11	1.13	5.41E-12
45	3.292E-11	1.00	2.731E-11	6.15E-11	0.22	1.74E-11	3.09E-11	1.17	5.3E-12
50	3.293E-11	0.96	2.919E-11	6.26E-11	0.25	1.97E-11	3.53E-11	1.20	5.63E-12
55	3.888E-11	1.09	3.125E-11	6.59E-11	0.28	1.85E-11	3.74E-11	1.23	5.89E-12
60	3.007E-11	1.10	2.55E-11	6.56E-11	0.33	1.82E-11	4.19E-11	1.27	6.28E-12
65	3.214E-11	1.19	3.125E-11	6.79E-11	0.37	1.67E-11	4.57E-11	1.29	6.52E-12
70	3.709E-11	1.16	3.231E-11	7.27E-11	0.42	1.71E-11	4.79E-11	1.33	6.94E-12
75	3.055E-11	1.06	2.71E-11	7.3E-11	0.49	1.5E-11	5.3E-11	1.37	7.21E-12
80	3.002E-11	1.16	2.736E-11	7.53E-11	0.59	1.54E-11	5.39E-11	1.42	7.42E-12

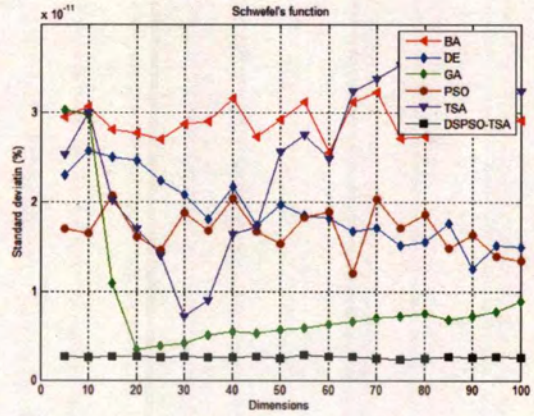
Table 3.4

Simulation results obtained from six methods for Schwefel's function (cont')

Problem dimension	BA f(xi,yi) Mean.	BA time (s) Mean.	BA Standard Deviation	DE f(xi,yi) Mean.	DE time (s) Mean.	DE Standard Deviation	GA f(xi,yi) Mean.	GA time (s) Mean.	GA Standard Deviation
85	4.012E-11	1.24	3.035E-11	7.27E-11	0.70	1.76E-11	5.83E-11	1.56	6.77E-12
90	2.824E-11	1.26	2.867E-11	7.99E-11	0.89	1.25E-11	6.22E-11	1.60	7.2E-12
95	3.617E-11	1.30	3.125E-11	7.67E-11	1.19	1.51E-11	6.5E-11	1.63	7.68E-12
100	3.399E-11	1.35	2.91E-11	7.67E-11	1.34	1.49E-11	6.92E-11	1.69	8.86E-12

Problem Dimension	PSO f(xi,yi) Mean.	PSO time (s) Mean.	PSO Standard Deviation	TSA f(xi,yi) Mean.	TSA time (s) Mean.	TSA Standard Deviation	DSPSO-TSA f(xi,yi) Mean.	DSPSO-TSA time (s) Mean.	DSPSO-TSA Standard Deviation
5	2.45E-11	0.11	1.7E-11	6.12E-11	0.04	2.53E-11	5.25E-12	0.09	2.66E-12
10	2.55E-11	0.16	1.64E-11	7.2E-11	0.09	3.01E-11	4.99E-12	0.11	2.56E-12
15	2.73E-11	0.22	2.07E-11	8.39E-11	0.21	2.02E-11	5.17E-12	0.13	2.62E-12
20	2.6E-11	0.28	1.61E-11	8.95E-11	0.33	1.7E-11	5.08E-12	0.14	2.64E-12
25	2.28E-11	0.37	1.46E-11	8.91E-11	0.51	1.39E-11	5.27E-12	0.15	2.51E-12
30	2.64E-11	0.43	1.88E-11	9.28E-11	0.69	7.14E-12	5.21E-12	0.17	2.62E-12
35	2.54E-11	0.51	1.68E-11	9.24E-11	0.92	8.98E-12	5.56E-12	0.18	2.52E-12
40	2.62E-11	0.58	2.04E-11	9.06E-11	1.18	1.64E-11	5.55E-12	0.20	2.55E-12
45	2.66E-11	0.66	1.67E-11	8.96E-11	1.41	1.72E-11	5.36E-12	0.22	2.69E-12
50	2.67E-11	0.73	1.53E-11	8.1E-11	1.72	2.56E-11	5.06E-12	0.23	2.41E-12
55	2.26E-11	0.82	1.83E-11	7.42E-11	1.98	2.75E-11	5.38E-12	0.25	2.83E-12
60	2.37E-11	0.90	1.89E-11	7.24E-11	2.25	2.48E-11	5.21E-12	0.27	2.63E-12
65	2.09E-11	0.95	1.2E-11	6.9E-11	2.59	3.24E-11	5.61E-12	0.27	2.7E-12
70	2.82E-11	1.06	2.03E-11	6.55E-11	2.95	3.38E-11	5.04E-12	0.30	2.44E-12
75	2.46E-11	1.14	1.7E-11	4.91E-11	3.31	3.54E-11	5.46E-12	0.32	2.4E-12
80	2.82E-11	1.19	1.86E-11	5.52E-11	3.69	3.31E-11	5.2E-12	0.34	2.48E-12
85	2.43E-11	1.37	1.48E-11	5.27E-11	4.16	3.27E-11	5.28E-12	0.37	2.62E-12
90	2.43E-11	1.44	1.63E-11	5.39E-11	4.61	3.25E-11	5.04E-12	0.38	2.55E-12
95	2.39E-11	1.65	1.39E-11	4.25E-11	5.38	3.11E-11	5.04E-12	0.45	2.62E-12
100	2.25E-11	1.94	1.34E-11	5.21E-11	6.00	3.24E-11	5.57E-12	0.48	2.53E-12





(a) (b)

Fig. 3.11. (a) Mean value of Schwefel's function vs. problem dimension; (b) Standard deviation of solution vs. problem dimension of Schwefel's function

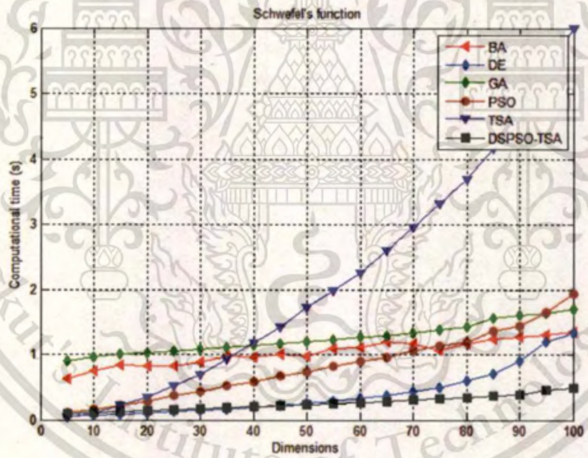


Fig. 3.12. Computational time vs. problem dimension of Schwefel's function

3.4 Conclusion

In this chapter, the results from investigation of the performance of the proposed method which compared with BA, DE, GA, PSO and TSA are shown. In the simulation results, mean value of the function, standard deviation of the solution and the computational time of the 100 running with different random seed of 5, 10, 15, …95 and 100 dimension ranges are reported on Table 3.1-Table 3.4. From the results in Table 3.1-Table 3.4, the DSPSO-TSA has outperforming than five methods for all functions and all dimension ranges. The results represented on the tables are plotted in Figures 3.1-3.12. The results both from table and figure show that the DSPSO-TSA has solution robustness for all dimensions, can be said that the performance of the proposed method less sensitive to the incremental of the problem dimension ranges than the BA, DE, GA, PSO and TSA. Another advantage issue of the proposed method is the less execution time as compared with the BA, DE, GA, PSO and TSA.

Chapter 4

DSPSO-TSA Application for Economic Dispatch and Optimal Power Flow Problem

In this thesis, a new hybrid technique based on PSO and TSA is proposed. This proposed approach is called distributed Sobol PSO and TSA (DSPSO-TSA). Three addition mechanisms i.e. Sobol sequence, distributed processing and hybrid PSO-TSA had been presented. In order to show the effectiveness, the proposed method is tested to solve ED problem with four case studies considering nonsmooth and noncontinuous fuel cost functions and optimal power flow (OPF) with two case studies.

4.1 Formulation of economic dispatch problem

The main objective of solving the ED problem is to minimize the total fuel cost operation of each generating unit in electric power system while to satisfy the total real power demand, real power balance and generating unit constraints. The formulation of ED problem can be modeled as

$$\textit{Minimize } F_T = \sum_{i=1}^{n} F_i(P_i) \tag{4.1}$$

Generally, the fuel cost of thermal generating unit can be represented as a quadratic function.

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{4.2}$$

subjected to

$$\sum_{i=1}^{n} P_i = D + P_L \tag{4.3}$$

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{4.4}$$

where F_T is total fuel cost, $F_i(P_i)$ is fuel cost of generating unit i, n is the number of generating units, a_i, b_i and c_i are cost coefficients of unit i, P_i is the real power of generating unit i, D is total load demand, P_L is total transmission line loss, $P_{i,\min}$ is the minimum generation limit of unit i and $P_{i,\max}$ is the maximum generation limit of unit i.

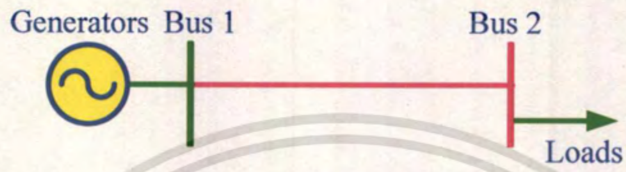


Figure 4.1 A simplified schematic diagram of power system

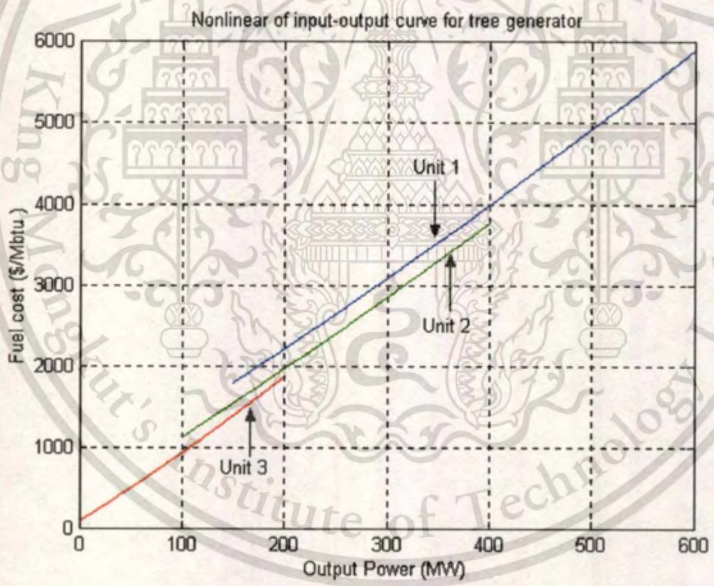


Figure 4.2 Quadratic fuel cost function of the three generating units

In practical, the fuel cost function and generating unit constraints have been modified in many forms due to operation limitation characteristics, e.g. multiple fuels for generating the electricity [6-8], effect of steam valve starting [9-12] and prohibited operating zones [13-16]. In this thesis, three types of ED problems are considered.

4.1.1 Economic dispatch problem with valve-point loading effect

The generating unit considering the effect of valve-point loading has different input-output curve compared with the smooth cost function. To consider the valve-point effects, sinusoidal functions are added to the quadratic cost functions as follows [1-5]:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + \left| e_i \times \sin(f_i \times (P_i^{\min} - P_i)) \right| \tag{4.5}$$

where e_i and f_i are the coefficients of the generating unit reflecting valve-point effects.

4.1.2 ED problem with multiple fuels and valve-point loading effect

The piecewise quadratic cost function with the valve point loading effect can be given as:

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1}P_{i1} + c_{i1}P_i^2 + \left|e_{i1}\sin\left(f_{i1}\left(P_{i,\min} - P_i\right)\right)\right| & \text{fuel type 1, } \left(P_{i,\min} \le P_i < P_{i1}\right) \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2 + \left|e_{i2}\sin\left(f_{i2}\left(P_{i,\min} - P_i\right)\right)\right| & \text{fuel type 2, } \left(P_{i1} \le P_i \le P_{i2}\right) \\ \vdots & \vdots \\ a_{ik} + b_{ik}P_i + c_{ik}P_i^2 + \left|e_{ik}\sin\left(f_{ik}\left(P_{i,\min} - P_i\right)\right)\right| & \text{fuel type } k, \left(P_{ik-1} \le P_i < P_{i,\max}\right) \end{cases} \tag{4.6}$$

where a_{ik}, b_{ik}, c_{ik}, e_{ik} and f_{ik} are cost coefficient of generating unit i.

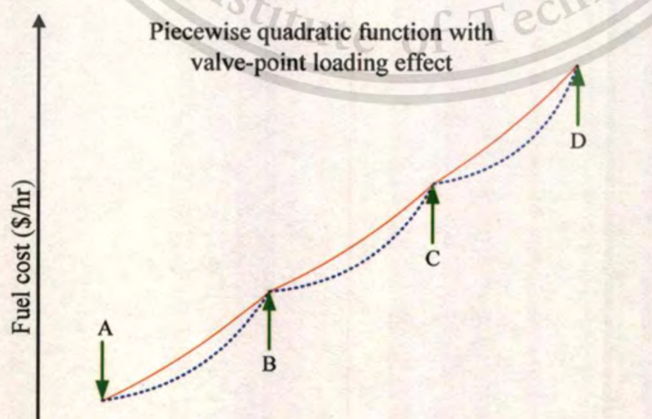


Figure 4.3 Multiples fuel with valve-point loading effect function cost characteristic

4.1.3 Economic dispatch problem with prohibited operating zones

The ED problem considering the prohibited operating zones, the ramp rate limit constraints and transmission line losses can be expressed as follows:

A) Prohibited operating zones (POZ):

$$
\begin{aligned}
& P_{i,\min} \le P_i \le P_{i,1}^{L} \\
& P_{i,k-1}^{U} \le P_i \le P_{i,k}^{L} \qquad k = 2,3,\ldots,n_i\,,\, n_i = l,\ldots,m \\
& P_{i,n_i}^{U} \le P_i \le P_{i,\max}
\end{aligned}
\tag{4.7}
$$

where k is the number of prohibited operating zones of generating unit i, $P_{i,k}^{L}$ and $P_{i,k}^{U}$ are lower and upper limits of the k^{th} prohibited zone of generating unit i, respectively.

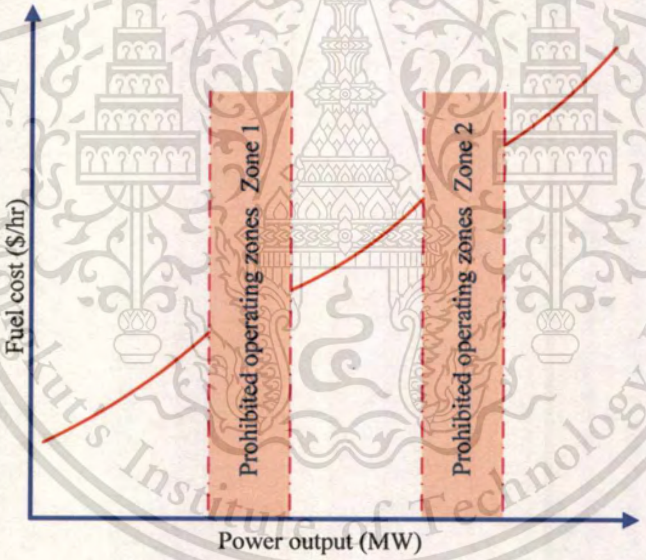


Figure 4.4 Two prohibited operating zones in quadratic function cost curve

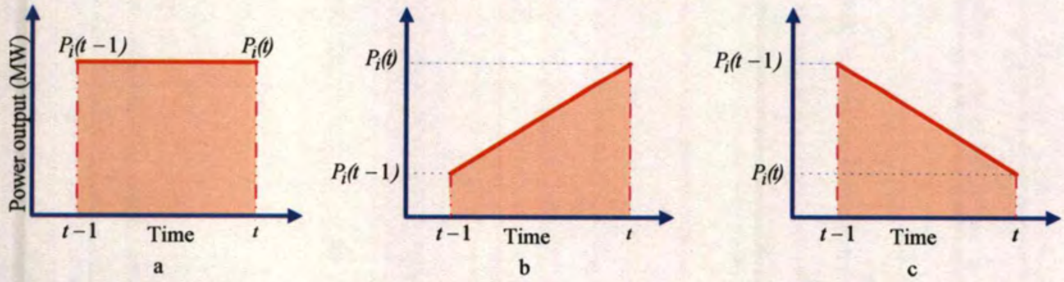


Figure 4.5 Three feasible conditions of generating unit i

B) Ramp rate limit:

The increasing and decreasing of generating outputs can not be adjusted instantaneously.

1) Generation increasing:

$$P_{i(t)} + P_{i(t-1)} \leq UR_i \tag{4.8}$$

2) Generation decreasing:

$$P_{i(t-1)} - P_{i(t)} \leq DR_i \tag{4.9}$$

where $P_{i(t)}$ is output power of generating unit i at current, $P_{i(t-1)}$ is output power at previous, UR_i is upramp limit of generating unit i (MW/time − period) and DR_i is downrampt limit of generating unit i (MW/time − period)

C) Generator operating constraint:

$$\max(P_{i,\min}, P_{i(t-1)} - DR_i) \leq P_{i(t)} \leq \min(P_{i,\max}, P_{i(t-1)} + UR_i) \tag{4.10}$$

D) Transmission line losses:

$$P_L = \sum_{i=1}^{n}\sum_{i=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} P_i B_{oi} + B_{oo} \tag{4.11}$$

where B_{ij}, B_{oi}, B_{oo} transmission line loss coefficients.

4.2. Implementation of the DSPSO-TSA for solving ED problem

The proposed DSPSO is applied to solve the ED problem considering generator constraints, nonsmooth cost function, noncontinuos cost function and transmission loss. The DSPSO process splits the group of particles to many subgroups. Each subgroup has a memory which stores its current particle in the subgroup p^{th} $(x_{i,p})$, the best particle in the subgroup p^{th} ($pbest_{i,p}$) and the best particle of among all the particles in the subgroup p^{th} ($gbest_{i,p}$). All subgroups will use these memories together in order to look into the best particle among all groups ($Gbest$). The inertia factor is created by the Sobol random generator and compared with the conventional inertia factor, as presented in Figure 4.6

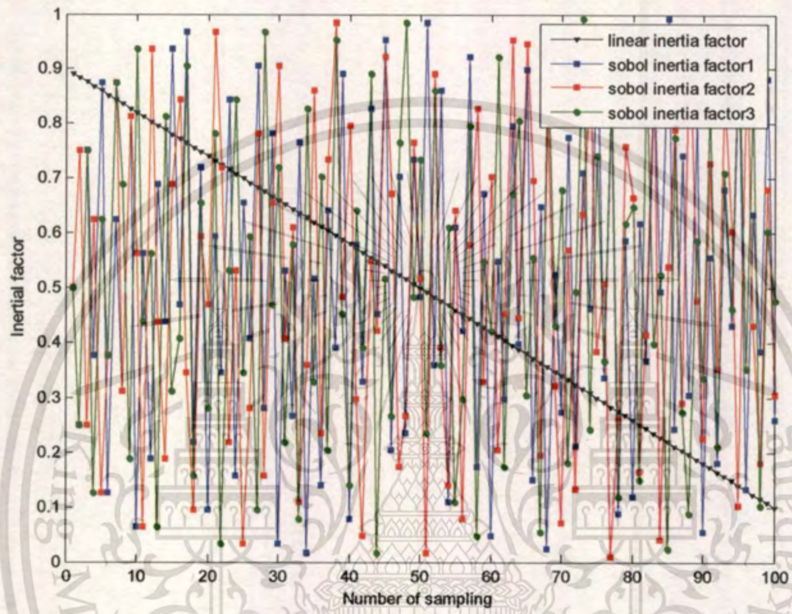


Figure 4.6 100 samples of Sobol inertia factor for DSPSO-TSA

The detail of the DSPSO which is applied to solve the ED problem is described in the following steps.

Step 1. Set these parameters c_1, c_2, c_3, c_4. Generate the inertia factor using Sobol random generator. Set the first iteration $t = 0$ and determine the number of subgroup (p).

Step 2. Initialize the positions of the particles of each subgroup. Represent the position of each particle in subgroup as $x_i = (x_{i1}, x_{i2}, ..., x_{in})$ with following equation.

$$P_i^{\mathrm{U}} = \min(P_i^{\max}, P_i^0 + UR_i),\ P_i^{\mathrm{L}} = \max(P_i^{\max}, P_i^0 - DR_i)$$

$$x_i = P_i^{\mathrm{L}} + rand() \times (P_i^{\mathrm{U}} - P_i^{\mathrm{L}}), \quad i = 1,2,...m, \tag{4.12}$$

The particles in the p^{th} subgroup have n decision variables (number of the generating units) and m dimension (number of particles), represent as following matrix.

$$X_p^{th} = \begin{bmatrix} x_{11} & \cdots & x_{1d} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{id} & \cdots & x_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{mi} & \cdots & x_{md} & \cdots & x_{mn} \end{bmatrix} \tag{4.13}$$

These particles (real power output) must satisfy the constraints i.e. power balance (4.3), power limit of generating units (4.4), prohibited zones (4.7), ramp-rate limit (4.8-4.9) and generator operation (4.10). The initial velocities of particles can be created by using below equation.

$$V_i = P_i^L \times rand() + rand() \times (P_i^U - P_i^L) \tag{4.14}$$

Step 3. Calculate the transmission line losses. The objective function of each particle x_{id} in the matrix X_p^{th} using below equation.

$$F_T = \sum_{i=1}^{m}\sum_{d=1}^{n} F_i(P_{id}) + \varphi \cdot (\sum_{i=1}^{m}\sum_{d=1}^{n} P_{id} - D - P_L)^2 + \gamma \cdot \sum_{i=1}^{m}\sum_{d=1}^{n} Z_{id} \tag{4.15}$$

where φ is a penalty factor for penalizing real power balance constraint, γ is a penalty factor for penalizing prohibited operating zones constraint. These parameters are verified by empirical testes. Observantly, if the real power balance constraint has been satisfied, the second term will become to zero. Otherwise, this term will affect to the objective function. The Z_{id} is the prohibited operating zones index which express as follow.

$$Z_{id} = \begin{cases} 1 & \text{if } x_{id} \text{ violates the prohibited operating zones} \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

Thus, if the real power fallen in the prohibited zones, the objective function will be penalized with γ. Otherwise the last term will become to zero.

Step 4. The best particle of the each subgroup will be represented as $pbest_{i,p}$. The best particle among all particles of each subgroup will be set as $gbest_p$. Finally, the best particle among all groups will be set as $Gbest$.

Step 5. Apply an elimination high cost technique for replacing the worst particle with the best particle. Ascending order the particles based on their objective values, the two worst particles will be replaced with the two best particles.

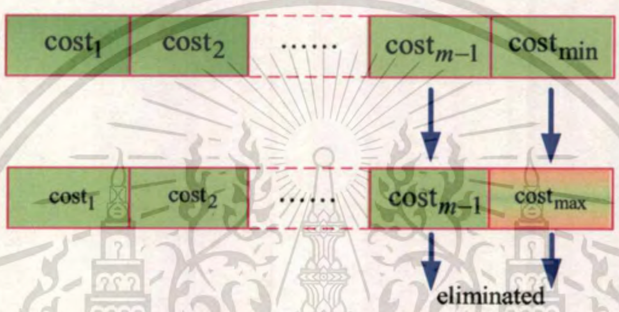


Figure 4.7 Depicts elimination high cost technique

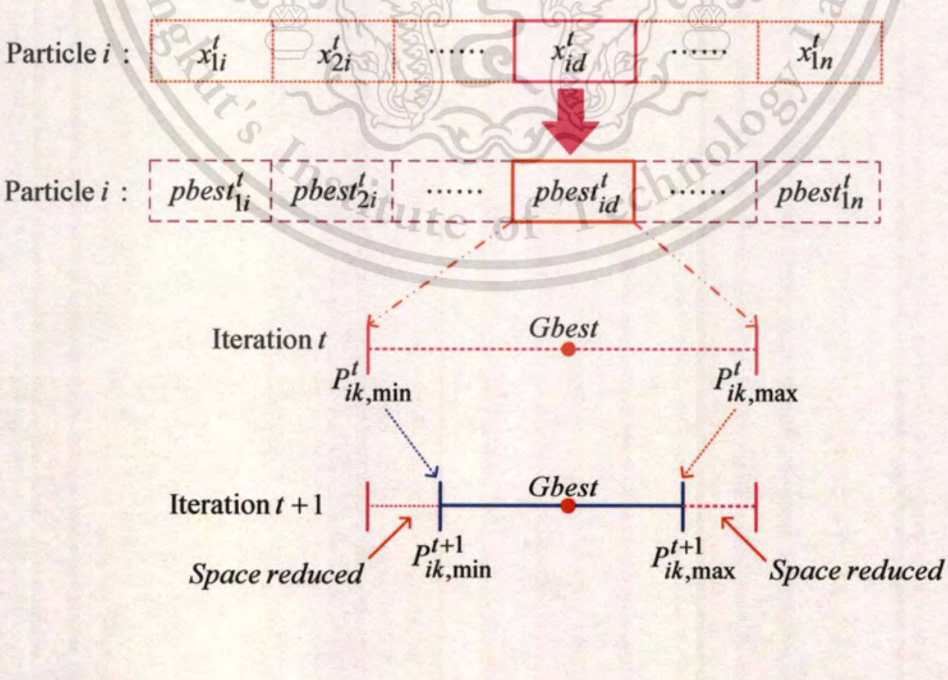


Figure 4.8 Illustration of reduce the search space technique

Step 6. Apply a reduced search space technique. The search space of each particle must be reduced dynamically depend on the distance between the $gbest_p^t$ and the $gbest_p^{t-\Delta}$ where Δ is previous iteration. The reduced search spaces must satisfy the equations (4.7-4.10). Therefore, the maximum and minimum boundaries will be calculated with equation as below.

$$P_i^U = \max(gbest_p^t,\ gbest_p^{t-\Delta}) \text{ and } P_i^L = \min(gbest_p^t,\ gbest_p^{t-\Delta}) \quad (4.17)$$

Step 7. Calculate the velocity of all the particles is given in equation (4.18).

$$V_i^{(t+1)} = K \times \begin{pmatrix} \omega_{sb} \times V_i^{(t)} + c_1 \times r_1 \times (pbest_i - x_i^{(t)}) + c_2 \times r_2 \times (gbest - x_i^{(t)}) \\ + c_4 \times r_4 \times (Gbest - x_i^{(t)}) \end{pmatrix} \quad (4.18)$$

Step 8. Check the velocity limits, if a particle exceed the velocity limits that is $V_i^{t+1} > V_i^{\max}$ then set $V_i^{t+1} = 0.5 \times P_i^U$, otherwise set $V_i^{t+1} = -0.5 \times P_i^L$.

Step 9. Update the position of all particles using equation (4.19), the updating position must be satisfied all the constraints.

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)},\ \text{i} = 1,2,...,\text{n}, \quad (4.19)$$

Step 10. Adjust the position of the particle which violated the constraint. The new velocity must be adjusted by using equation (4.20), and modified the new position by using equation (4.19)

$$V_{i,p}^{(t+1)} = K \times \begin{pmatrix} \omega_{sb} \times V_{i,p}^{(t)} + c_1 \times r_1 \times (pbest_{i,p} - x_{i,p}^{(t)}) + c_2 \times r_2 \times (gbest_{i,p} - x_{i,p}^{(t)}) \\ + c_3 \times r_3 \times (pbest_{i,j} - x_{i,p}^{(t)}) + c_4 \times r_4 \times (Gbest - x_{i,p}^{(t)}) \end{pmatrix} \quad (4.20)$$

where $pbest_{i,j}$ is the best position of the particle in the subgroup j, r_3 is random number between 0 and 1, c_3 is accelerator factor.

Step 11. Check stopping criterion. In this thesis, the stopping criterion use two conditions i.e. the maximum iterations and sticking of solution. If satisfy the stopping criterion, increase iteration $t = t + 1$ and go back to step 3. Otherwise go to step 12.

Step 12. Apply the TSA, the particles of the *Gbest* which obtained by DSPSO in the latest iteration will be defined as the optimal real power output of the generating units. This value will be used as an initial solution of TSA which described in the chapter 2.

4.3 Test systems and numerical solutions

The parameters of the DSPSO-TSA such as c_1, c_2, c_3, c_4, and p must be determined. In this thesis, these parameters had been empirically tested. All case studies set c_1, c_2 equal to 2.05. For case study 1 and 2 set $c_3 = 1.20$, $c_4 = 0.85$. But in case study 3 and 4 set $c_3 = 1.35$, $c_4 = 0.7$. The number of the subgroup is defined as 3, number of particles for each subgroup as 10 and maximum iteration of each subgroup as 100. The Sobol inertia factor for balancing the exploration is generated cover 0 to 1 ranges. The TSA parameters such as maximum iteration set as 100, neighborhood sizes set as $2\times$(problem dimensional), recency set as 20 and frequency set as 15.

In addition, the case study 1 and 2 are neglected the transmission line losses and prohibited operating zones constraint. The penalty factors i.e. γ set as 0 and φ set as 1000. For case study 3 and 4 which are considering the prohibited operating zones, γ set as 100 and φ set as 5000. All methods (DSPSO-TSA, PSO, TSA and GA) are implemented by MatLab programming language and executed in an AMD Athlon (tm) XP 1600+, 256 MB RAM personal computer. The DSPSO-TSA, GA, PSO and TSA are implemented in MATLAB language and executed on an Intel(R) Core2 Duo 3.0 GHz personal computer with a 4.0 GB of RAM.

The parameters of each algorithm have to be defined before solving the non-linear mathematic function. The results of each method are compared with the proposed method. The parameters of each method describe as following:

Genetic Algorithm's parameters

- Number of populations = 100
- Number of precision bit = 14
- Generation gap = 0.9
- Crossover rate = 0.7
- Mutation rate = 0.5

Particle Swarm Optimization's parameters

- Number of particles = 100
- Maximum inertia factor = 0.9
- Minimum inertia factor = 0.1
- Accelerating factors = 2.00
- Maximum velocity is *0.5 × search space dimension*
- Minimum velocity are *-0.5 × search space dimension*

Tabu Search Algorithm's parameters

- Number of neighborhoods size = *2 × problem dimension*
- Tabu list size = *neighborhood size × 3*
- Restriction period = 100
- Frequently period = 50

DSPSO-TSA's parameters

- Number of groups = 3
- Number of particles for each group = 10
- The Sobol inertia factor had been generated between 0 and 1
- Accelerating factors = $c_1=2.05$, $c_2=2.05$ *for all case study*
- Accelerating factors $c_3=1.20$, $c_4=0.85$ *for case study 1 and 2*
- Accelerating factors $c_3=1.35$, $c_4=0.7$ *for case study 3 and 4*
- Maximum velocity is 0.5 × *search space dimension*
- Minimum velocity is -0.5 × *search space dimension*
- The parameters of the TSA are same as above

4.3.1. Case study 1: Thirteen-unit system with valve-point effect

The effectiveness of the DSPSO-TSA is tested with the thirteen thermal generating units with valve-point loading effect system, which neglect the transmission line losses. The information of generating units of test system is given in [8-10] and Table

A.1 in appendix. The best numerical solution is compared against three classical optimization methods i.e. TSA, GA, and PSO, shown in Table 4.1. The efficiency of the proposed approach is discovered with 100 independent trials, the simulation results are presented in Table 4.2.

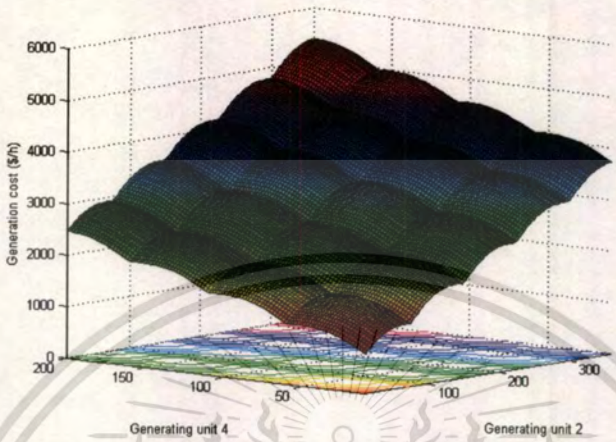


Figure 4.9 Characteristics of fuel cost function of case study 1

Table 4.1 The best results comparison of case study 1

Unit power output (MW)	TSA	GA	PSO	DSPSO-TSA
P_1	628.32	628.3	628.32	628.32
P_2	299.08	299.12	299.2	299.2
P_3	299.18	299.07	299.21	299.2
P_4	159.69	159.6	159.73	159.73
P_5	159.71	159.71	159.73	159.73
P_6	159.73	159.69	159.73	159.73
P_7	159.74	159.73	159.73	159.73
P_8	159.65	159.73	159.73	159.73
P_9	159.7	159.71	159.73	159.73
P_{10}	76.96	77.34	77.4	77.4
P_{11}	73.76	77.32	77.4	77.4
P_{12}	92.2	88.36	92.4	87.69
P_{13}	92.29	92.32	87.68	92.4
Total power output (MW)	2520	2520	2520	2520
Total generation cost ($/h)	24171.21	24170.8	24170.17	**24169.923**

Table 4.2 Comparison results of case study 1

Methods	Generation cost ($/h)			Standard deviation	Computational time (s)
	Min.	Average	Max.		
TSA	24171.211	24184.055	24392.203	41	10.52
GA	24170.804	24188.394	24567.974	59.53	19.39
PSO	24170.167	24184.849	24377.890	38.86	10.75
DSPSO-TSA	**24169.923**	**24173.137**	**24230.803**	**7.72**	**2.92**

Table 4.3 Selection of c_3 and c_4 by empirically for case study 1

c3	c4	Generation cost ($/h)			CPU time (s)			Standard deviation
		Min.	Mean.	Max.	Min.	Mean.	Max.	
1.03	1.03	24169.9309	24176.3721	24297.8600	2.55	2.94	4.05	20.7116
1.05	1.00	24170.0190	24176.3248	24274.5116	2.44	2.93	3.47	16.3497
1.08	0.98	24169.9853	24177.6028	24308.1614	2.36	2.92	3.03	21.5581
1.10	0.95	24170.2344	24180.5038	24358.5923	2.44	2.91	3.05	29.7695
1.13	0.93	24169.9355	24176.8280	24304.7143	2.42	2.93	3.06	20.6721
1.15	0.90	24169.9672	24182.0502	24425.3873	2.45	2.93	3.05	41.5854
1.18	0.88	24169.9464	24179.9959	24370.7560	2.48	2.93	3.03	32.1777
1.20	***0.85***	***24169.9234***	***24173.1369***	***24230.8026***	***2.48***	***2.92***	***3.03***	***7.71504***
1.23	0.83	24170.3573	24178.8259	24364.5630	2.41	2.89	3.03	27.5285
1.25	0.80	24170.0247	24176.8681	24360.2288	2.39	2.91	3.03	27.5172
1.28	0.78	24169.9828	24178.2546	24390.7958	2.44	2.92	3.03	31.8010
1.30	0.75	24170.0178	24178.0374	24369.1673	2.34	2.88	3.03	25.8201
1.33	0.73	24169.9749	24175.6961	24346.4727	2.47	2.89	3.03	20.0033
1.35	0.70	24169.9770	24175.7732	24308.6894	2.36	2.90	3.05	17.5568
1.38	0.68	24170.0870	24174.6522	24301.5676	2.44	2.97	3.86	14.9392
1.40	0.65	24170.1623	24177.4128	24337.4279	2.33	2.89	3.06	23.5144
1.43	0.63	24170.0109	24178.2459	24300.1294	2.36	2.87	3.06	23.7249
1.45	0.60	24169.9441	24175.4512	24362.0118	2.34	2.87	3.05	20.8224
1.48	0.58	24170.2956	24174.4073	24245.5905	2.31	2.97	3.67	9.1315
1.50	0.55	24170.2167	24177.9972	24380.0694	2.33	2.89	4.02	24.5039

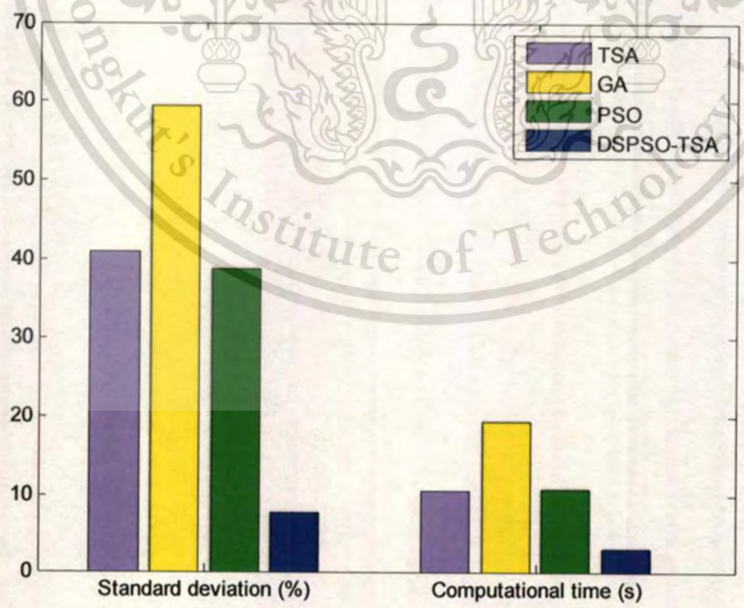


Figure 4.10 Comparison of standard deviation and computational time of case study 1

The results show that the DSPSO-TSA can converge to the global optimum faster than other methods. Figure 4.11 shows convergence characteristic curve of the DSPSO-TSA, TSA, GA and PSO. Observantly, the DSPSO-TSA converges faster. Figure 4.12 depicts the process of the DSPSO-TSA to obtain the *Gbest* .

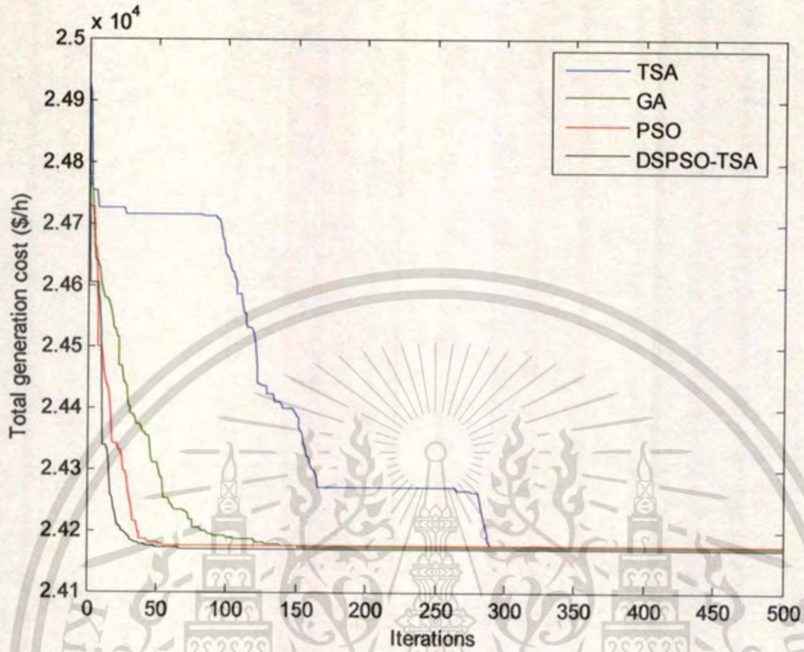


Figure 4.11 Comparison of convergences of the DSPSO-TSA with TSA, GA and PSO of case study 1

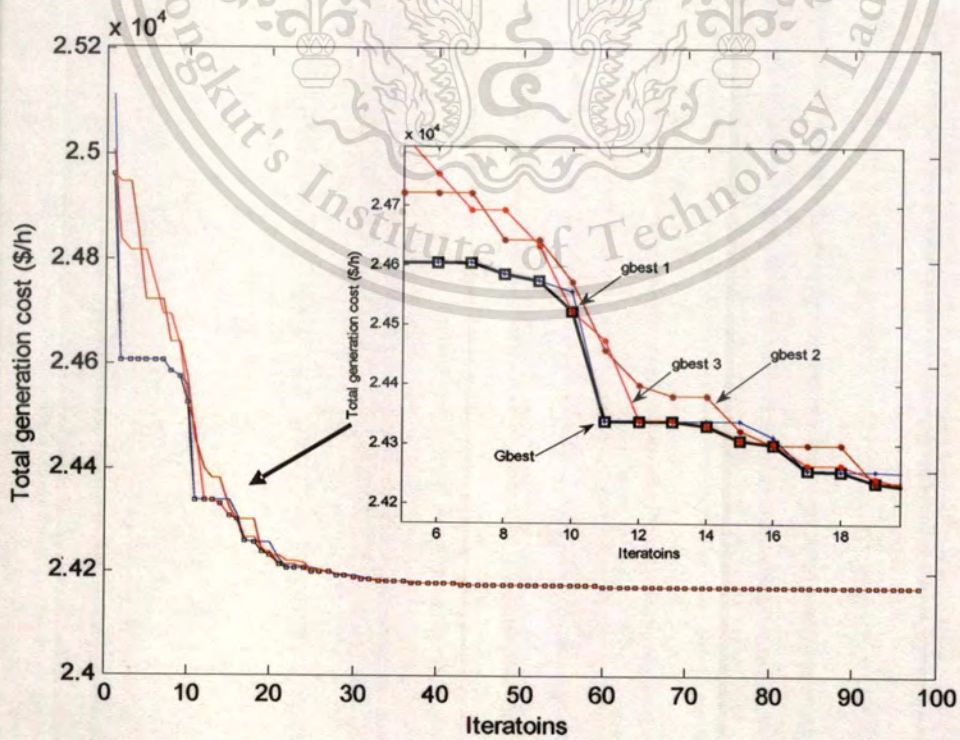


Figure 4.12 The process to obtain the *Gbest* for each iteration of case study 1

4.3.2 Case study 2: Ten-unit system with multiple fuels and valve-point loading effect

This case study contains ten generating units with both multiple fuels and valve-point loading effect [11] and shown Table A.2 in appendix. The total load demand is 2700 MW and not considering transmission line loss. Table 4.3 depicts comparison the minimum cost, average cost and maximum generation cost. The fuel function cost describes in equation (4.6). Comparison the best simulation results of the DSPSO-TSA with CGA_MU [11], IGA_MU [11], TSA, GA and PSO are shown in Table 4.4.

Table 4.4 Results comparison of case study 2

Methods	Generation cost ($/h)			Standard	Average
	Min.	Average.	Max.	deviation	CPU time (s)
CGA_MU[11]	624.7193	627.6087	633.8652	-	26.64
IGAMU[11]	624.517	625.8692	630.8705	-	7.32
TSA	624.3078	624.8285	624.0582	0.0407	9.71
GA	624.505	624.7419	624.4207	0.1281	18.37
PSO	624.3045	624.5054	624.3156	0.0704	11.04
DSPSO-TSA	**623.8375**	**623.8625**	**623.9001**	**0.0106**	**3.44**

Table 4.5 The best cost comparison of case study 2

Output (MW)	Methods											
	CGA_MU [11]		IGA_MU [11]		TSA		GA		PSO		DSPSO-TSA	
	PG(MW)	Fuel type	PG(MW)	Fuel type	PG(MW)	Fuel type	PG(MW)	Fuel type	PG(MW)	Fuel type	PG(MW)	Fuel type
P_1	222.0108	2	219.1261	2	219.4959	2	216.2974	2	225.5729	2	217.5568	2
P_2	211.6352	1	211.1645	1	206.7093	1	210.4431	1	208.2240	1	211.2163	1
P_3	283.9455	1	280.6572	1	291.3532	1	281.2672	1	278.8078	1	279.6495	1
P_4	237.8052	3	238.477	3	237.6731	3	241.5130	3	238.0062	3	240.0422	3
P_5	280.448	1	276.4179	1	279.2478	1	264.4552	1	282.4136	1	279.9441	1
P6	236.033	3	240.4672	3	237.3793	3	242.3118	3	239.6464	3	239.7737	3
P_7	292.0499	1	287.7399	1	277.9598	1	284.1604	1	285.4269	1	287.7353	1
P_8	241.9708	3	240.7614	3	238.9435	3	240.5808	3	239.1045	3	239.5054	3
P_9	424.2011	3	429.337	3	429.9256	3	438.6755	3	425.5856	3	428.7090	3
P_{10}	269.9005	3	275.8518	3	281.3126	3	280.2956	3	277.2121	3	275.8677	3
Total power (MW)	2700		2700		2700		2700		2700		2700	
Total cost ($/h)	624.7193		624.5178		624.3078		624.5050		624.3046		**623.8375**	

The results show that the generation cost obtained from the DSPSO-TSA is better than other methods. Furthermore, the average computation time is lower than compared methods. The results obtained from the DSPSO-TSA had minimum cost, average cost and maximum generation cost are better than other methods. The 100 independent trials had converged to optimum solution illustrated in Figure 4.13.

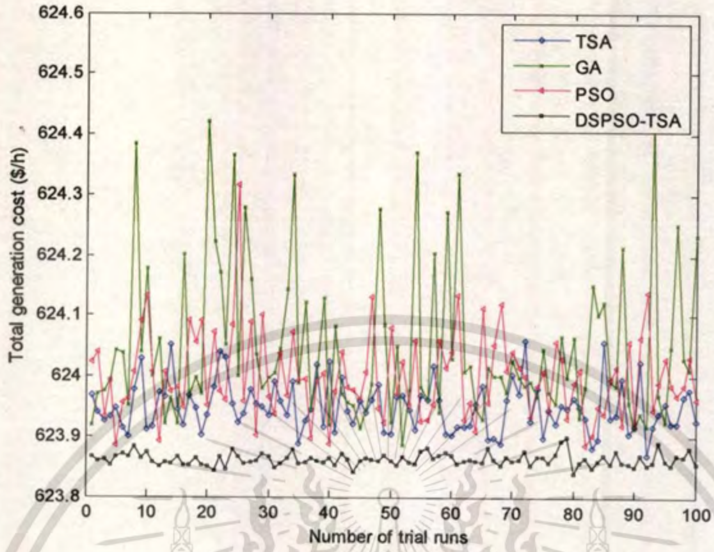


Figure 4.13 Convergences of 100 independent trials of case study 2

4.3.3 Case study 3: Six-unit system with prohibited operating zones

This case study is based on a six thermal generator units system, 26 buses and 46 transmission lines, the data of this system are shown in Table A.3-A.4 in appendix and references [13-14]. Table 4.5 reports the best dispatch results of the proposed method and other methods from literatures. Evidently, the DSPSO-TSA provides less the total generation cost when compared with GA [13], PSO [13], APSO [15], CPSO [16], MTS [19], AIS [20] and TSA.

Table 4.6 The best results comparison of the six generators

Power output (MW)	PSO [13]	GA [13]	APSO [15]	CPSO [16]	MTS [19]	AIS [20]	TSA	DSPSO-TSA
P_1	447.497	474.8066	446.6686	434.4295	448.1277	458.2904	449.3651	439.2935
P_2	173.3221	178.6363	173.1556	173.3231	172.8082	168.0518	182.252	187.7876
P_3	263.4745	262.2089	262.826	274.4735	262.5932	262.5175	254.2904	261.026
P_4	139.0594	134.2826	143.4686	128.0598	136.9605	139.0604	143.4506	129.4973
P_5	165.4761	151.9039	163.914	179.4759	168.2031	178.3936	161.9682	171.7101
P_6	87.128	74.1812	85.3437	85.9281	87.3304	69.3416	86.0185	86.1648
Total power (MW)	1276.01	1276.03	1275.376	1276	1276.023	1275.655	1277.345	1275.514
P_{loss} (MW)	12.9584	13.0217	12.4216	12.9583	13.0205	12.655	14.3449	13.0421
Total cost ($/h)	15,450	15,459	15,444	15,446	15450.06	15,448	15,451.63	**15,441.57**

Table 4.6 shows the effectiveness of the DSPSO-TSA in term of the solution quality among 100 independent trials. Obviously, the proposed method yields higher solution quality than remaining methods. Figure 4.14 shows the profiles of the simulation solutions obtained from running the DSPSO-TSA with 100 independent trials.

Table 4.7 Selection of c_3 and c_4 by empirically for case study 3

c3	c4	Generation cost ($/h)			CPU time (s)			Standard deviation
		Min.	Mean.	Max.	Min.	Mean.	Max.	
1.03	1.03	15443.525	15443.955	15448.073	0.16	1.38	2.42	0.5159
1.05	1.00	15443.480	15443.900	15444.682	0.17	1.22	2.34	0.2408
1.08	0.98	15443.468	15443.866	15444.995	0.17	1.20	2.34	0.2211
1.10	0.95	15443.541	15443.927	15445.366	0.16	1.29	2.34	0.2633
1.13	0.93	15443.449	15443.894	15445.302	0.16	1.34	2.34	0.2678
1.15	0.90	15443.477	15445.154	15570.829	0.16	1.31	2.34	12.6978
1.18	0.88	15443.473	15443.930	15446.387	0.16	1.22	2.34	0.3500
1.20	0.85	15443.579	15443.881	15444.486	0.16	1.26	2.34	0.1713
1.23	0.83	15443.489	15445.069	15558.792	0.16	1.43	2.34	11.4911
1.25	0.80	15443.465	15443.914	15445.850	0.16	1.36	2.33	0.3382
1.28	0.78	15443.467	15443.905	15445.844	0.16	1.27	2.33	0.3124
1.30	***0.75***	***15441.571***	***15443.836***	***15446.216***	***0.16***	***1.07***	***2.34***	***0.3714***
1.33	0.73	15443.468	15444.106	15467.220	0.16	1.26	2.33	2.3438
1.35	0.70	15443.487	15443.892	15444.926	0.16	1.35	2.34	0.2118
1.38	0.68	15443.523	15443.922	15445.898	0.16	1.32	2.33	0.3259
1.40	0.65	15443.492	15443.992	15451.091	0.16	1.33	2.33	0.7905
1.43	0.63	15443.468	15443.891	15445.152	0.16	1.31	2.33	0.2416
1.45	0.60	15443.494	15444.048	15448.124	0.16	1.40	2.66	0.6713
1.48	0.58	15443.460	15443.925	15445.222	0.16	1.30	2.45	0.2873
1.50	0.55	15443.522	15443.960	15447.803	0.17	1.54	2.36	0.5044

Table 4.8 Simulation results comparison of the six generators

Methods	Generation cost ($/h)			Average CPU time (s)	Standard deviation
	Min.	Average.	Max.		
PSO [13]	15,450.00	15,454.00	15,492.00	14.86	-
GA [13]	15,459.00	15,469.00	15,469.00	41.58	-
APSO [15]	15,443.58	15,449.99	-	-	6.77
CPSO [16]	15,446.00	15,449.00	15,490.00	8.13	-
MTS [19]	15,450.06	15,451.17	15,453.64	1.29	0.93
AIS [20]	15,448.00	15,459.70	15,472.00	-	6.25
TSA	15451.631	15462.263	15506.451	5.98	18.09
DSPSO-TSA	**15,441.57**	**15,443.84**	**15,446.22**	**1.07**	**0.37**

4.3.4 Case study 4: Fifteen-unit system with prohibited operating zones

This case study comprises the 15 thermal units systems are shown in Table A.5-A.6 in appendix [13-14]. Clearly, the solution quality of the DSPSO-TSA is higher than GA [14], PSO [14], APSO [15], CPSO [16], MTS [19], AIS [20] and TSA.

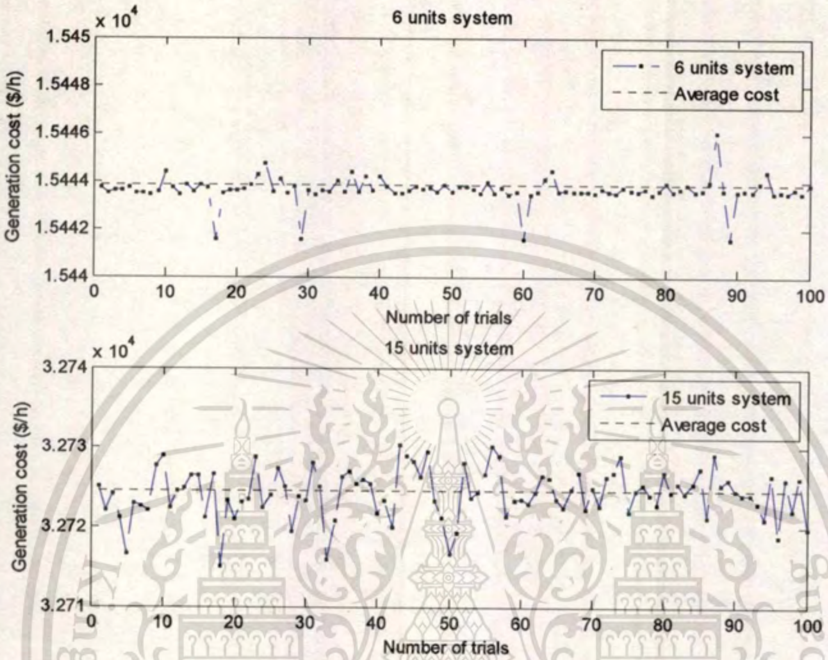


Figure 4.14 Solution profiles of the 6 and 15 generator units systems

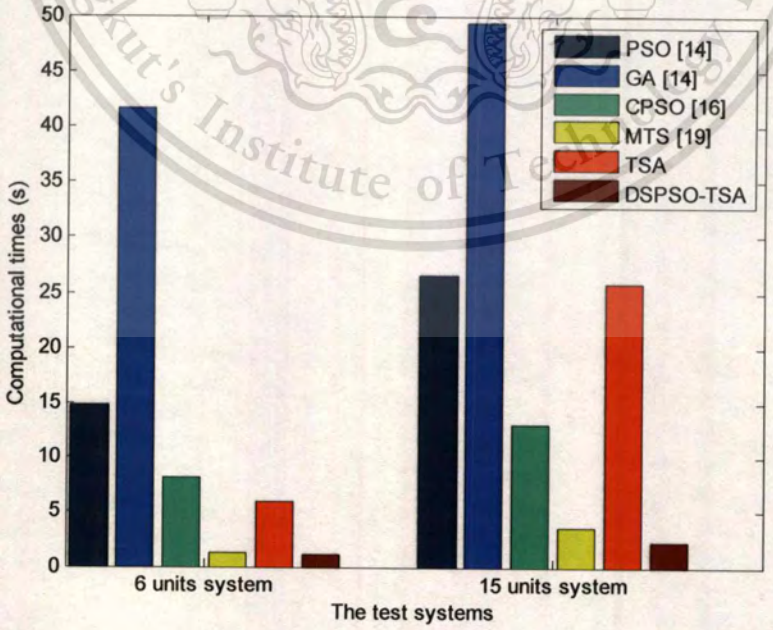


Figure 4.15 Computational time comparison of six and fifteen generator units system

Furthermore, the computational time has lower than other methods. The best simulation results are shown in Table 4.9. Table 4.10 depicts the simulation results of the DSPSO-TSA in term of the minimum cost, average cost and maximum cost among 100 independent trials and compared with methods from literatures. It is clear that the best result obtained by the proposed method less than the rest methods. Frequencies of convergence of independent initial solution among 100 independent trials are shown in Figure 4.14. Figure 4.15 shows convergence time of case study 3 and 4.

Table 4.9 The best results comparison of the DSPSO-TSA with seven methods of case study 4

Power output (MW)	PSO [14]	GA [14]	APSO[15]	CPSO[16]	MTS [19]	AIS[20]	TSA	DSPSO-TSA
P_1	450.298	455	455	450.02	453.992	441.159	440.5	453.627
P_2	440	440	380.01	454.06	379.743	409.587	346.8	379.895
P_3	118.118	119.572	130	124.81	130	117.298	110.88	129.482
P_4	122.484	117.984	126.52	124.81	129.923	131.258	122.46	129.923
P_5	270	270	170.01	151.06	168.088	151.011	177.74	168.956
P_6	284.04	324.896	460	460	460	466.258	459.11	459.907
P_7	430	314.152	428.28	434.57	429.225	423.368	406.41	429.971
P_8	151.274	140.381	60	148.46	104.31	99.948	107.55	103.673
P_9	111.394	113.275	25	63.59	35.036	110.684	107.27	34.909
P_{10}	75.112	128.625	159.79	101.12	155.883	100.229	140.56	154.593
P_{11}	50.456	63.23	80	28.66	79.899	32.057	78.47	79.559
P_{12}	44.658	44.156	80	20.91	79.904	78.815	74.17	79.388
P_{13}	47.317	77.28	33.7	25	25.022	23.568	31.95	25.487
P_{14}	37.184	25.714	55	54.41	15.259	40.258	37.38	15.952
P_{15}	35.09	34.025	15	20.62	15.08	36.906	22.47	15.64
Total power (MW)	2667.4	2668.3	2658.32	2662.1	2661.36	2662.04	2663.7	2660.96
Power loss (MW)	37.3329	38.2499	28.37	32.13	31.3523	32.4075	33.811	30.952
Total cost ($/h)	33,020.00	33,149.00	32,724.78	32,834.00	32,716.87	32,854.00	32,918.00	**32,715.06**

Table 4.10 Results comparison of the DSPSO-TSA for the fifteen generators system

Methods	Generation cost ($/h)			Average CPU time (s)	Standard deviation
	Min.	Average.	Max.		
PSO [14]	32,858.00	33,105.00	33,331.00	26.59	-
GA [14]	33,113.00	33,228.00	33,337.00	49.31	-
APSO [15]	32,742.78	32,976.68	-	-	133.93
CPSO [16]	32,834.00	33,021.00	33,318.00	13.13	-
MTS [19]	32,716.87	32,767.21	32,796.15	3.65	17.51
AIS [20]	32,854.00	32,873.25	32,892.00	-	10.81
TSA	32,917.87	33,066.76	33,245.54	25.75	66.82
DSPSO-TSA	**32,715.06**	**32,724.63**	**32,730.39**	**2.3**	**8.4**

4.4 Optimal Power Flow Problem

The optimal power flow (OPF) problem has been well studied over the past few decades [72-75]. The OPF problem could be treated as a nonlinear optimization problem with nonlinear objective function and subject to several equality and inequality constraints. Figure 4.16 shows the mostly used objective function in OPF problem [35]. Many optimization techniques have been applied to solve this problem. Conventional techniques such as Newton method, gradient methods, linear programming, dynamic programming and interior point methods often have problems of convergence and difficulties in locating the global optima. As indicated in [77], these methods rely on convexity to obtain the global optimum solution and forced to simplify relationships in order to ensure convexity.

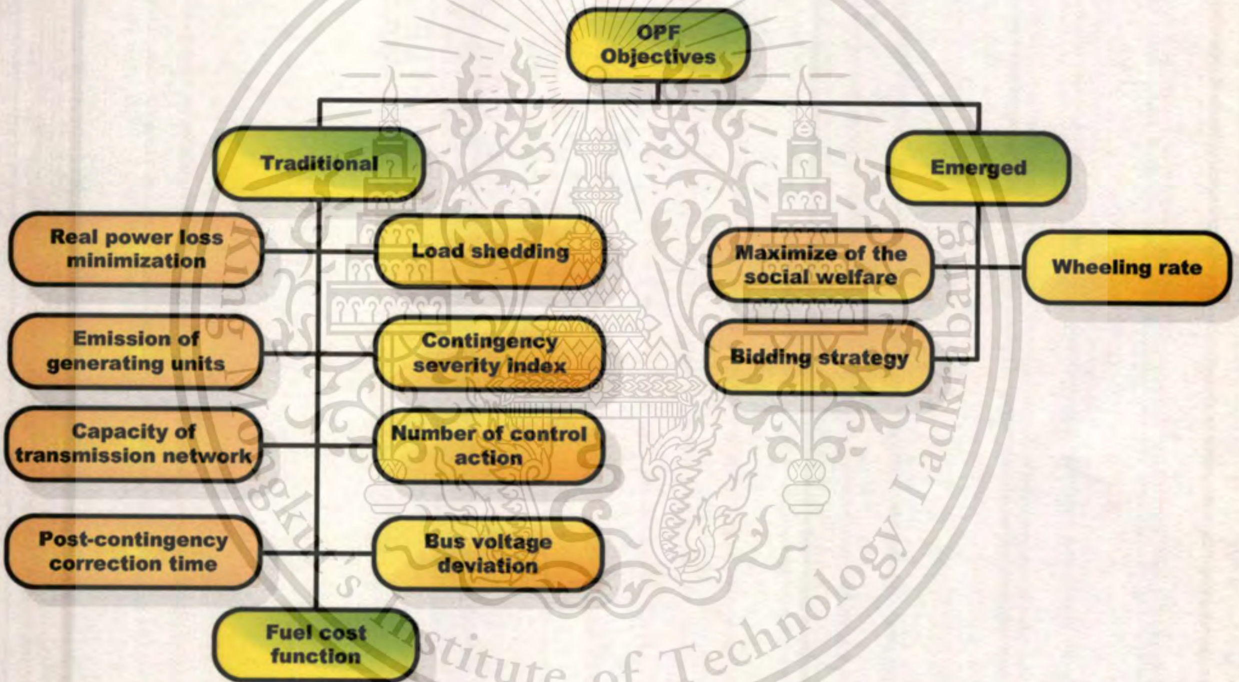


Figure 4.16 Most commonly used objective functions in the OPF problem

A major difficulty of the OPF problem is the nature of the control variables since some of them are continuous (real power outputs and voltages) and others are discrete (transformer tap setting, phase shifters, and reactive injections). Optimal values are computed in order to achieve a certain goal subjected to number of equality and inequality constraints. In general, the OPF problem can be presented as:

$$\text{Min } f(x,u) \tag{4.21}$$

Subjected to

$$g(x,u)=0 \tag{4.22}$$

$$h(x,u)\le 0 \tag{4.23}$$

where Min $f(x,u)$ is the objective function. Generally, $g(x,u)=0$ represents the real and reactive power balance equations and $h(x,u)\le 0$ represents security limits. u is a set of control variables, which includes active power and voltage magnitude of generator buses, voltage angle and magnitude of the swing bus, and tap position of LTCs. x is a set of dependent variables, which includes active and reactive power of the swing bus, voltage angle and reactive power of generator buses, and voltage angle and magnitude of load buses. The equality constraints (4.22) are the nonlinear power flow equations which are formulated as follows:

$$0=P_{Gi}-P_{Di}-V_i\sum_{j=1}^{n}V_j\left(G_{ij}\cos\theta_{ij}+B_{ij}\sin\theta_{ij}\right) \qquad i=1,\ldots,NB \tag{4.24}$$

$$0=Q_{Gi}-Q_{Di}-V_i\sum_{j=1}^{n}V_j\left(G_{ij}\sin\theta_{ij}-B_{ij}\cos\theta_{ij}\right) \qquad i=1,\ldots,NB \tag{4.25}$$

where NB is the total number of system buses, P_{Gi} and Q_{Gi} are the active and reactive power generations of bus i, P_{Di} and Q_{Di} are the active and reactive power loads of bus i; V_i is the voltage magnitude of bus i, θ_{ij} is the voltage angle difference between bus i and j; G_{ij} and B_{ij} are transfer admittance between bus i and j.

The inequality constraints (4.23) are the system operating constraints, including

(a) Generator constraints:

$$V_{G_i}^{\min} \le V_{G_i} \le V_{G_i}^{\max}, \quad i \in NG \tag{4.26}$$

$$P_{G_i}^{\min} \le P_{G_i} \le P_{G_i}^{\max}, \quad i \in NG \tag{4.27}$$

$$Q_{G_i}^{\min} \le Q_{G_i} \le Q_{G_i}^{\max}, \quad i \in NG \tag{4.28}$$

where NG is the number of generator buses.

(b) Transformer Constraints:

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i \in NT \tag{4.29}$$

where NT is number of transformers.

(c) Security constraints: For secure operation, the transmission line loading S_{l_i} is restricted by its upper limits $S_{l_i}^{\max}$ as:

$$V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max}, \quad i \in NB \tag{4.30}$$

$$S_{l_i} \le S_{l_i}^{\max}, \quad i \in nl \tag{4.31}$$

where nl is the number of transmission lines.

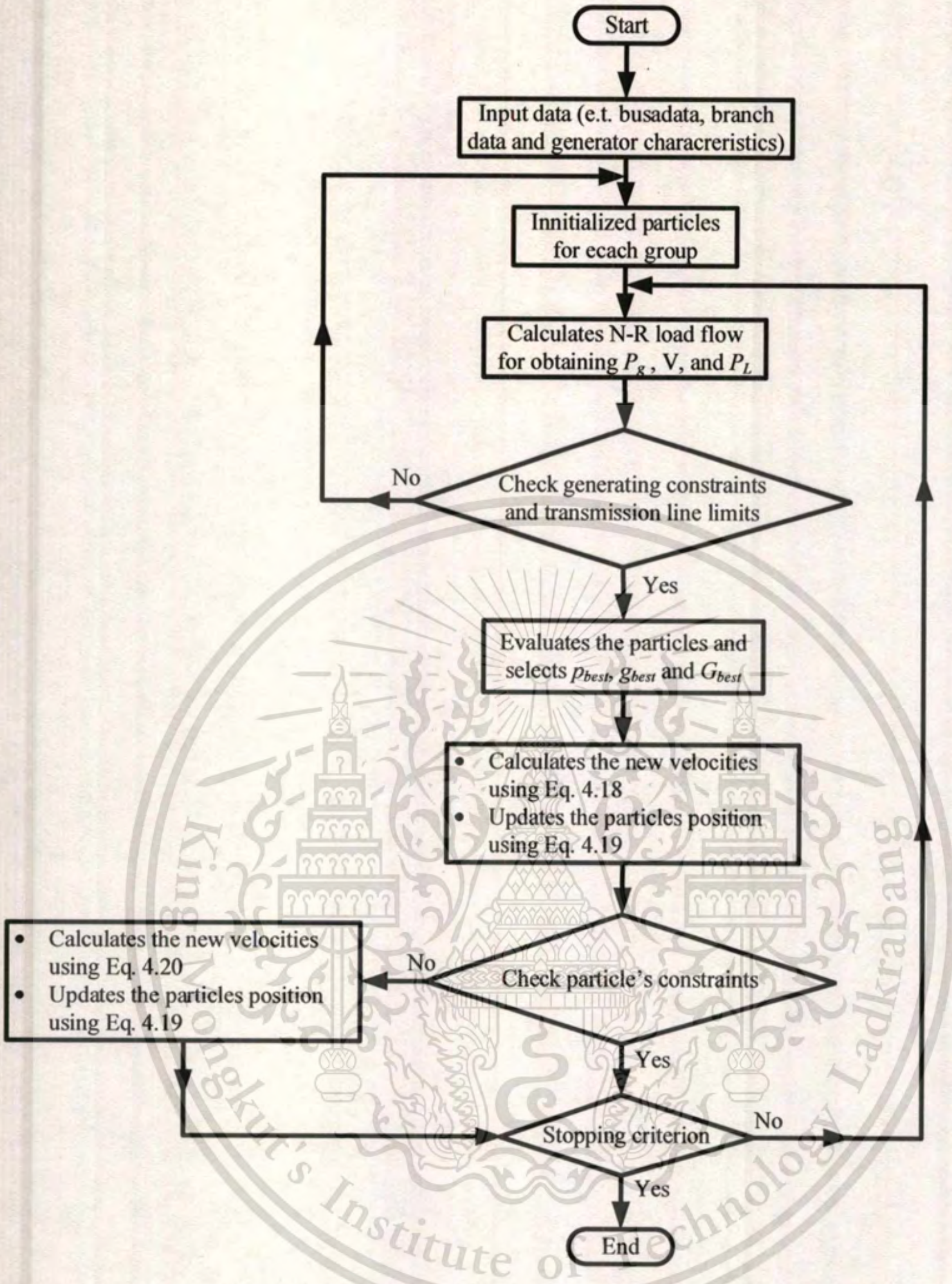


Figure 4.17 Flow chart of DSPSO applied to the OPF problem

Generally, the most common method for solving nonlinear constrained optimization problems is to transfer a constrained optimization problem into an unconstrained one using penalty functions. Thus, the objective function (4.21) is generalized as follows:

$$f(x,u) = \sum_{i=1}^{NG} f_i + \gamma_P (P_{G_1} - P_{G_1}^{\lim})^2 + \gamma_V \sum_{i=1}^{NL} (V_{L_i} - V_{L_i}^{\lim})^2$$

$$+ \gamma_Q \sum_{i=1}^{NG} (Q_{G_i} - Q_{G_i}^{\lim})^2 + \gamma_S \sum_{i=1}^{nl} (S_{l_i} - S_{l_i}^{\max})^2 \qquad (4.32)$$

where $\gamma_P, \gamma_V, \gamma_Q$ and γ_S are penalty factors and $x^{\lim}$ is upper and lower bound of control variable x , can be expressed as:

$$x^{\lim} = \begin{cases} x^{\max}; & x > x^{\max} \\ x^{\min}; & x < x^{\min} \end{cases} \qquad (4.33)$$

This section shows the results and effectiveness of the DSPSO-TSA for solving the OPF problem. Flow chart of applying the proposed method for solving this problem is shown in Figure 4.17. In order to illustrate the effectiveness and robustness of the proposed method, two test systems have been examined and considered. The first is the 6-bus test system [77] and the second one is the IEEE 30-bus test system [78-79]. The effectiveness of the proposed method is compared with several methods.

The parameters of each algorithm have to be defined before solving the non-linear mathematic function. The results of each method are compared with the proposed method. The parameters of each method describe as following:

Bee Algorithm's parameters

- Number of the initialized bees = 100
- Number of selected sites = 3
- Number of elite sites = 5
- Number of bees for selected site = 5
- Number of bees for elite site = 10

Differential Evolution algorithm's parameters

- Number of populations = 30
- Scaling factor = 0.45
- Crossover rate = 0.85

4.4.1 Six-bus test system

Figure 4.18 shows the single-line diagram of six-bus test system with 4 generating units, 7 transmission lines and total system load demand is 600 MW. Bus data and line data of this test system are shown in Table 4.17 and Table 4.18, respectively. The objective function of this case study is quadratic function like as equation (4.2). The cost coefficients are shown in Table 4.19.

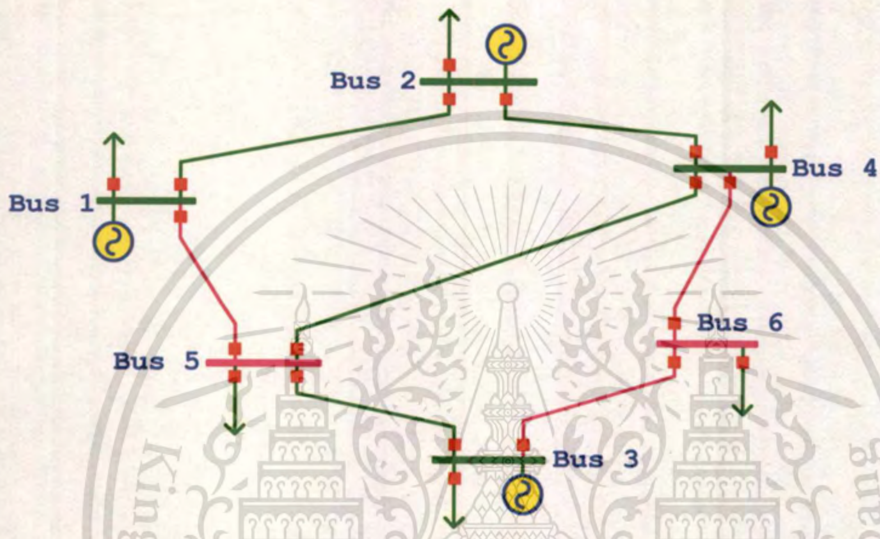


Figure 4.18 Schematic diagram of 6-bus test system

Table 4.11 Bus data of 6-bus test system

Bus	Voltage at bus			Loads		Generators			
	Type	|V|	Degree	Pload (MW)	Qload (Mvar)	Pgen (MW)	Qgen (Mvar)	Qmin (Mvar)	Qmax (Mvar)
1	1	1	0	100	20	0	0	-50	120
2	2	1	0	100	20	190	0	-50	120
3	2	1	0	100	20	110	0	-50	120
4	2	1	0	100	20	178	0	-50	120
5	0	0.95	0	100	50	0	0	0	0
6	0	0.95	0	100	10	0	0	0	0

For applying the proposed method, a particle size of each group is 3 and maximum iteration is taken as 100. Simulation results obtained from DSPSO-TSA compared with BA, DE, GA, PSO and TSA and shown in Table 4.14 and Table 4.15, respectively. Table 4.14 shows the results of proposed method and other methods after performing 100 independent runs, which include the minimum, average, maximum generation cost, computational time and standard deviation.

Table 4.12 Line data of 6-bus test system

From bus	To bus	Rs	Xs	Ysh/2	Tap size
1	2	0.04	0.08	0.02	1
1	5	0.04	0.08	0.02	1
2	4	0.04	0.08	0.02	1
3	5	0.04	0.08	0.02	1
3	6	0.04	0.08	0.02	1
4	5	0.04	0.08	0.02	1
4	6	0.04	0.08	0.02	1

Table 4.13 Cost coefficients and power output limits of generators

Units	Cost coefficient			Generators limit	
	a_i	b_i	c_i	P_{min} (MW)	P_{min} (MW)
1	105	12	0.012	50	250
2	96	9.6	0.0096	50	250
3	105	13	0.013	50	250
4	94	9.4	0.0094	50	250

It can be observed that the results obtained from DSPSO-TSA had outperformed than the rest methods in term of average generation cost, average computational time and standard deviation of solutions. The best solution is founded by each method are slightly different and reported in Table 4.15. Figure 4.19 illustrates the convergence characteristic of the proposed method compared with five methods, seen that the DSPSO-TSA has converge to near global optimum faster than BA, DE, GA, PSO and TSA.

The solution profiles of 100 different trial runs of the proposed method compared with the rest methods shown in Figure 4.20 and Figure 4.21. Evident that the results of other methods had probably obtained the local optimum, thus the standard deviation value of generation cost of the proposed method has lower than other methods. Selecting new parameter in the proposed method, namely c_3 and c_4 are investigated. Figure 4.22 and Figure 4.23 demonstrates the example and some value of c_3 and c_4. The figure shows that, c_4 is fixed while c_3 varied from 1 to 2. The best value of these parameter obtained from empirically tests are $c_3 = 1.40$ and $c_4 = 0.85$.

Table 4.14 Simulation results of 6-bus test system

Methods	Generation cost ($/h)			CPU	Standard
	Min	Mean	Max	times (s)	deviation (%)
BA	7823.535	7823.624	7823.699	3.18	0.040911
DE	7823.537	7823.631	7823.698	1.42	0.044527
GA	7823.539	7823.638	7823.700	1.70	0.044880
PSO	7823.537	7823.626	7823.697	1.23	0.043208
TSA	7823.551	7823.626	7823.699	1.71	0.044562
DSPSO-TSA	***7823.535***	***7823.614***	***7823.698***	***0.50***	***0.045463***

Table 4.15 Best results comparison of the DSPSO-TSA with five methods of 6-bus test system

Power output (MW)	BA	DE	GA	PSO	TSA	DSPSO-TSA
P_1	91.201	91.032	91.293	91.246	90.651	91.020
P_2	196.345	196.367	196.721	196.503	196.116	196.638
P_3	86.476	86.236	86.203	86.103	87.096	85.469
P_4	236.791	237.207	236.636	237.003	236.893	237.801
Total power (MW)	610.813	610.842	610.852	610.855	610.757	610.928
Power loss (MW)	10.814	10.843	10.852	10.856	10.757	10.837
Total cost ($/h)	7823.535	7823.537	7823.539	7823.537	7823.551	***7823.535***

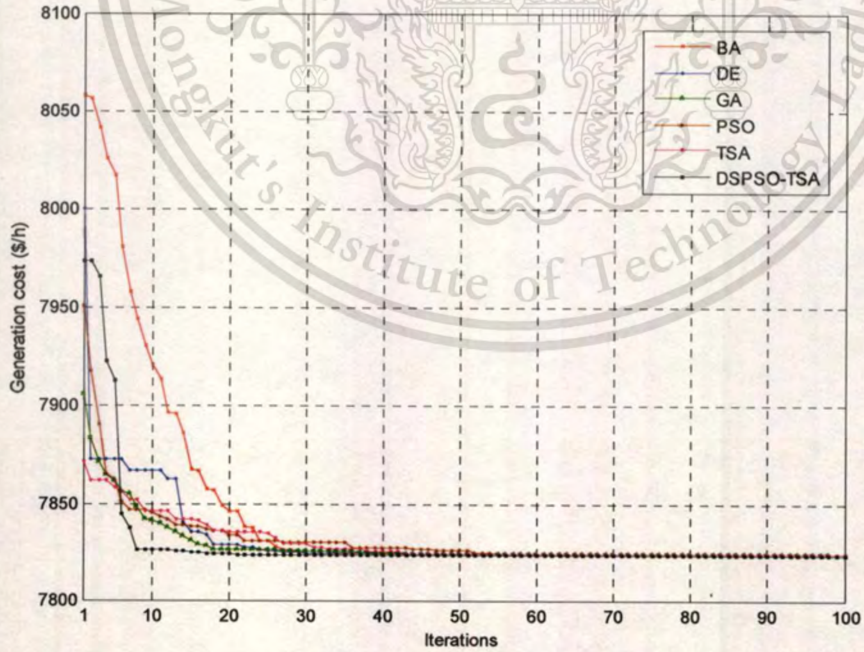


Figure 4.19 Convergence characteristics of proposed method compared with five methods of six-bus test system

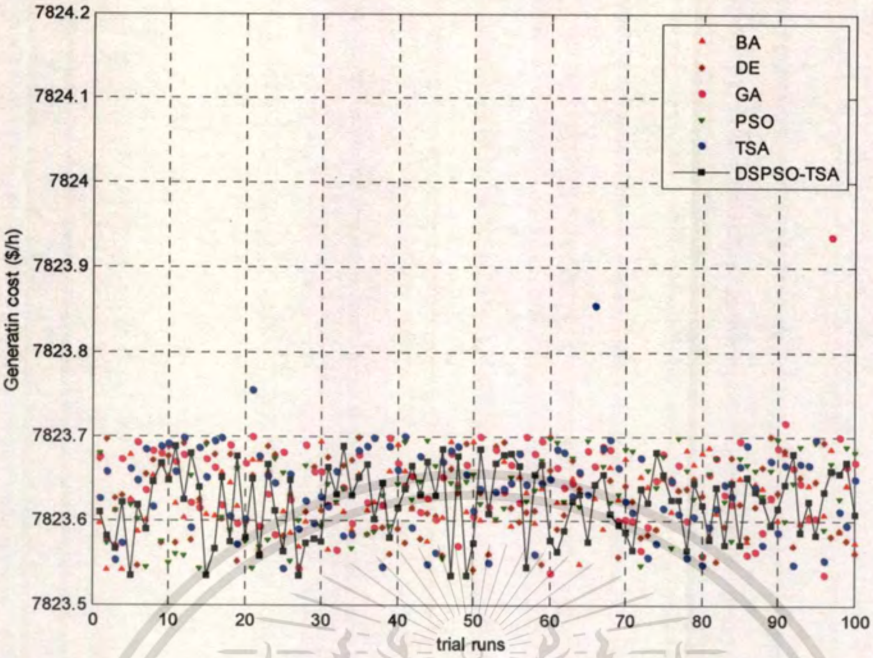


Figure 4.20 Comparison of solution profiles of the 6-bus OPF problem

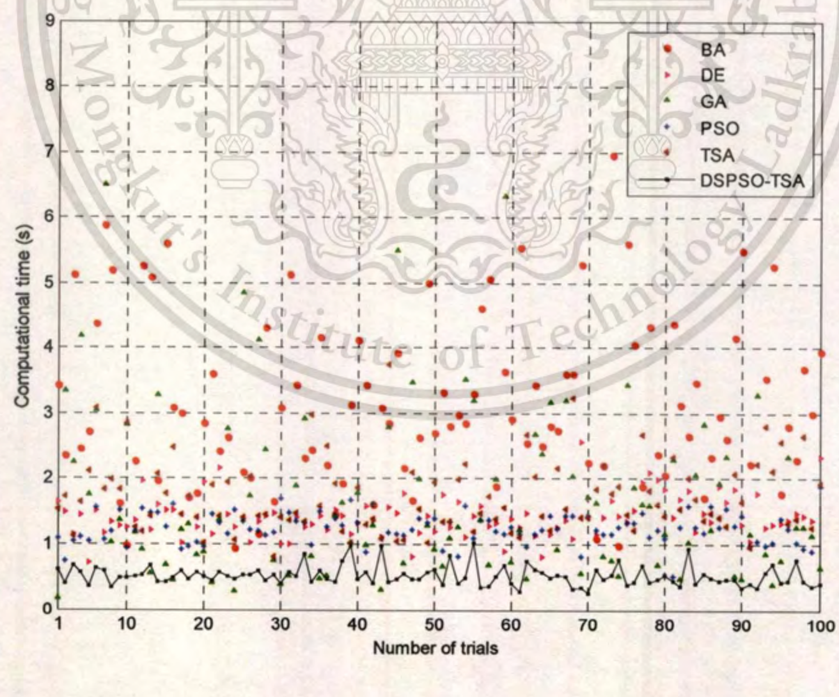


Figure 4.21 Comparison of computational time profiles of 6 methods of 6-bus OPF problem

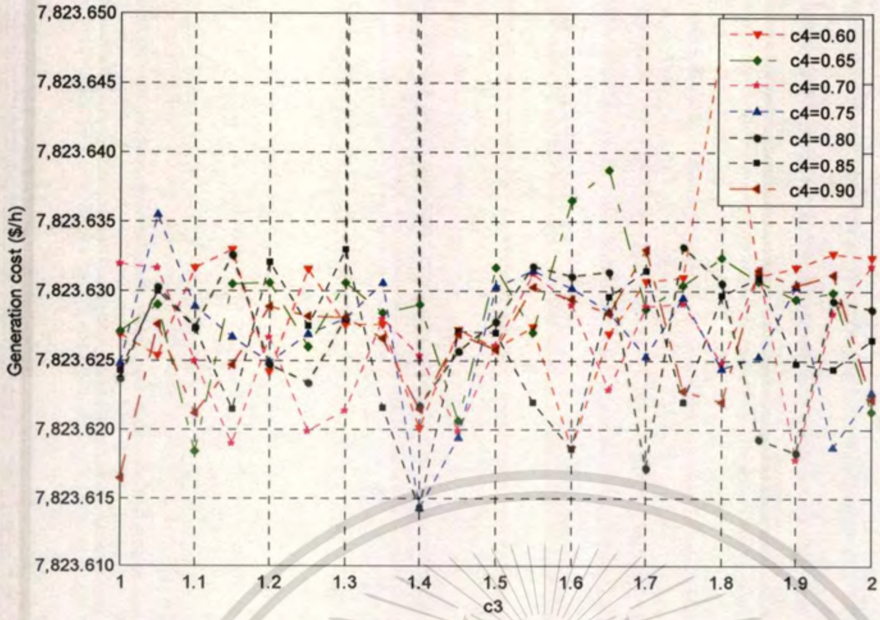


Figure 4.22 Solutions obtained from empirical testing c_3 and c_4 of 6-bus OPF problem

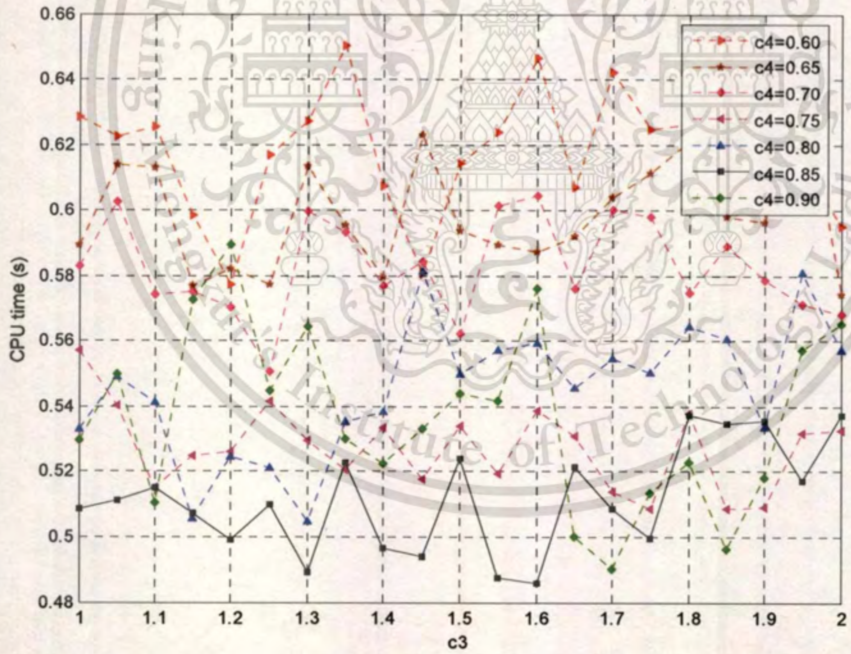


Figure 4.23 Computation time obtained from testing c_3 and c_4 of 6-bus OPF problem

4.4.2 IEEE 30-bus test system

In this section performance of proposed approach for solving the optimal power flow is tested on IEEE 30 bus test system. The system consist of six generating units committed at bus 1, 2, 5, 8, 11 and 13, forty-one transmission lines, four tap-changing transformers installed in lines 4-12, 6-9, 6-10 and 27-28 and two capacitors installed at bus five and twenty-four. Single-line diagram of test system is depicted in Figure 4.24. Bus data of this test system are given in Table 4.16 and Branch data depicted in Table 4.17. Table 4.18 given the generator characteristics.

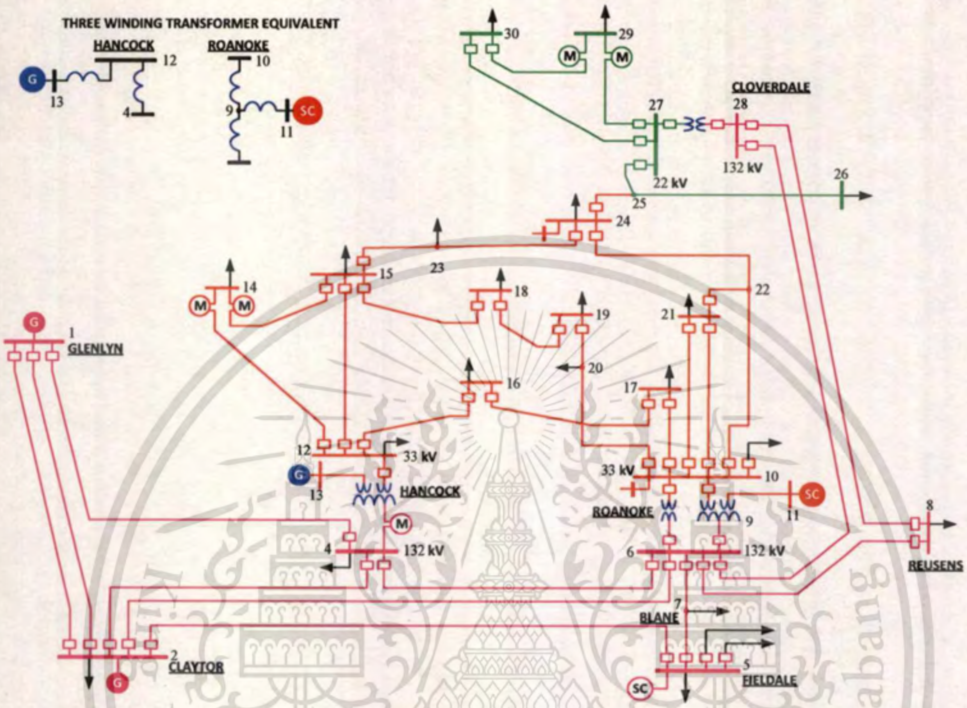


Figure 4.24 Single-line diagram of the IEEE 30-bus system

Table 4.16 Bus data for IEEE 30-bus system

Bus	Bus	Voltage	Angle	Load		Generator				Static
No.	code	(p.u.)	(degree)	PL (MW)	QL (Mvar)	Pg (MW)	Qg (Mvar)	Qmin	Qmax	Mvar
1	1	1.06	0.0	0	0	0	0	0	0	0
2	2	1.04	0.0	21.70	12.70	40	0	-40	50	0
3	0	1.00	0.0	2.400	1.200	0	0	0	0	0
4	0	1.06	0.0	7.600	1.600	0	0	0	0	0
5	2	1.01	0.0	94.20	19.00	0	0	-40	40	0
6	0	1.00	0.0	0	0	0	0	0	0	0
7	0	1.00	0.0	22.80	10.90	0	0	0	0	0
8	2	1.01	0.0	30.00	30.00	0	0	-10	60	0
9	0	1.00	0.0	0	0	0	0	0	0	0
10	0	1.00	0.0	5.800	2.000	0	0	0	0	19
11	2	1.08	0.0	0	0	0	0	-6	24	0
12	0	1.00	0.0	11.20	7.500	0	0	0	0	0
13	2	1.07	0.0	0	0	0	0	-6	24	0
14	0	1.00	0.0	6.200	1.600	0	0	0	0	0
15	0	1.00	0.0	8.200	2.500	0	0	0	0	0

Table 4.16 Bus data for IEEE 30-bus system (cont')

Bus No.	Bus code	Voltage (p.u.)	Angle (degree)	Load PL (MW)	Load QL (Mvar)	Generator Pg (MW)	Generator Qg (Mvar)	Qmin	Qmax	Static Mvar
16	0	1.00	0.0	3.500	1.800	0	0	0	0	0
17	0	1.00	0.0	9.000	5.800	0	0	0	0	0
18	0	1.00	0.0	3.200	0.900	0	0	0	0	0
19	0	1.00	0.0	9.500	3.400	0	0	0	0	0
20	0	1.00	0.0	2.200	0.700	0	0	0	0	0
21	0	1.00	0.0	17.50	11.20	0	0	0	0	0
22	0	1.00	0.0	0	0	0	0	0	0	0
23	0	1.00	0.0	3.200	1.600	0	0	0	0	0
24	0	1.00	0.0	8.700	6.700	0	0	0	0	4
25	0	1.00	0.0	0	0	0	0	0	0	0
26	0	1.00	0.0	3.500	2.300	0	0	0	0	0
27	0	1.00	0.0	0	0	0	0	0	0	0
28	0	1.00	0.0	0	0	0	0	0	0	0
29	0	1.00	0.0	2.400	0.900	0	0	0	0	0
30	0	1.00	0.0	10.60	1.900	0	0	0	0	0

Table 4.17 Branch data for IEEE 30-bus system

From	To	R (p.u)	X (p.u)	0.5xB (p.u)	Tr
1	2	0.0192	0.0575	0.0264	1
1	3	0.0452	0.1852	0.0204	1
2	4	0.0570	0.1737	0.0184	1
3	4	0.0132	0.0379	0.0042	1
2	5	0.0472	0.1983	0.0209	1
2	6	0.0581	0.1763	0.0187	1
4	6	0.0119	0.0414	0.0045	1
5	7	0.0460	0.1160	0.0102	1
6	7	0.0267	0.0820	0.0085	1
6	8	0.0120	0.0420	0.0045	1
6	9	0	0.2080	0	0.978
6	10	0	0.5560	0	0.969
9	11	0	0.2080	0	1
9	10	0	0.1100	0	1
4	12	0	0.2560	0	0.932
12	13	0	0.1400	0	1
12	14	0.1231	0.2559	0	1
12	15	0.0662	0.1304	0	1
12	16	0.0945	0.1987	0	1
14	15	0.2210	0.1997	0	1
16	17	0.0824	0.1923	0	1
15	18	0.1073	0.2185	0	1
18	19	0.0639	0.1292	0	1

Table 4.17 Branch data for IEEE 30-bus system (cont')

From	To	R (p.u)	X (p.u)	0.5xB (p.u)	Tr
19	20	0.0340	0.0680	0	1
10	20	0.0936	0.2090	0	1
10	17	0.0324	0.0845	0	1
10	21	0.0348	0.0749	0	1
10	22	0.0727	0.1499	0	1
21	22	0.0116	0.0236	0	1
15	23	0.1000	0.2020	0	1
22	24	0.1150	0.1790	0	1
23	24	0.1320	0.2700	0	1
24	25	0.1885	0.3292	0	1
25	26	0.2544	0.3800	0	1
25	27	0.1093	0.2087	0	1
28	27	0	0.3960	0	0.968
27	29	0.2198	0.4153	0	1
27	30	0.3202	0.6027	0	1
29	30	0.2399	0.4533	0	1
8	28	0.0636	0.2000	0.0214	1
6	28	0.0169	0.0599	0.065	1

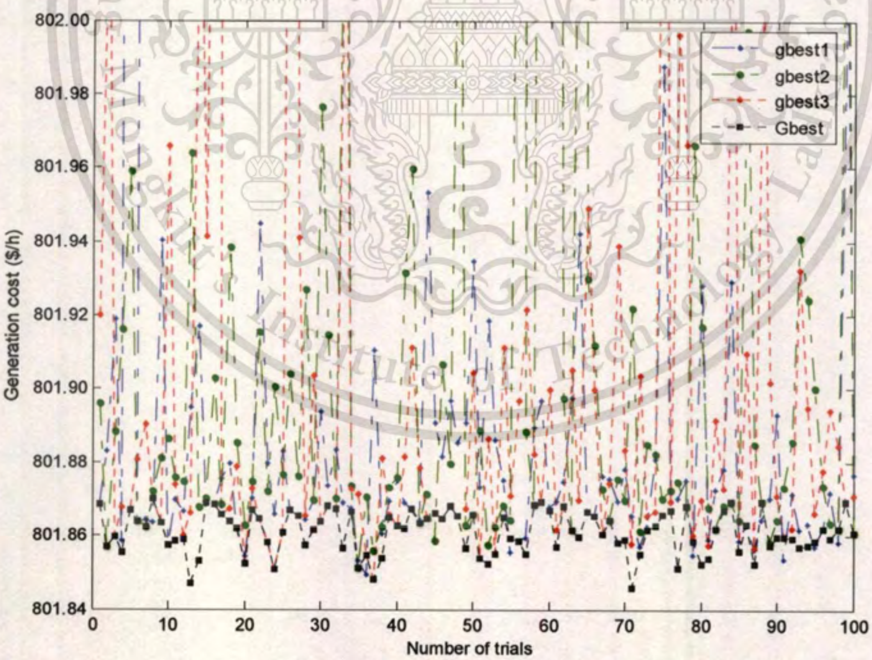


Figure 4.25 Illustrates an obtaining the G_{best} from g_{best} of 100 trails of IEEE 30-bus system

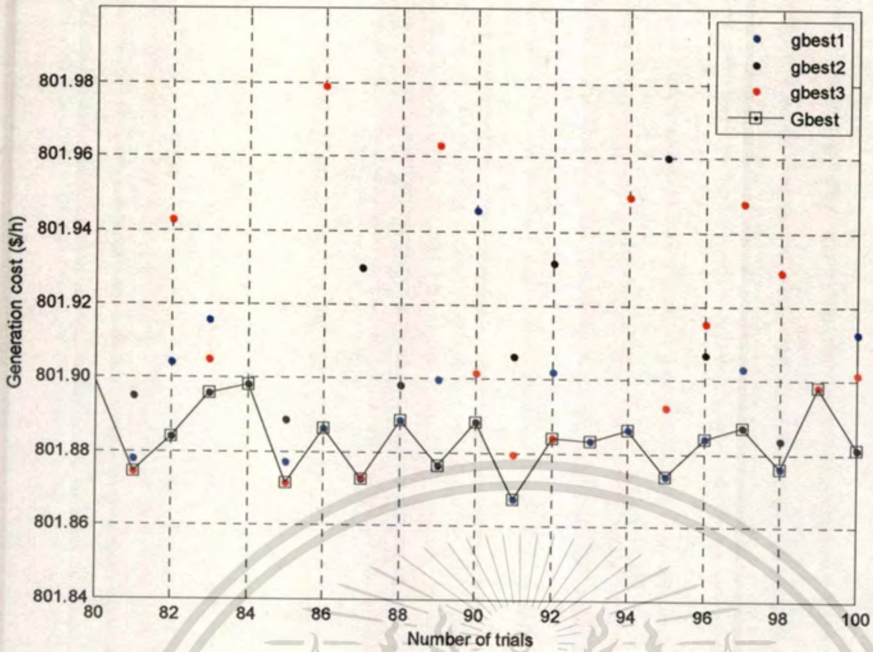


Figure 4.26 Illustrates an obtaining the G_{best} from g_{best} of 100 trails (zoomed) of IEEE 30-bus system

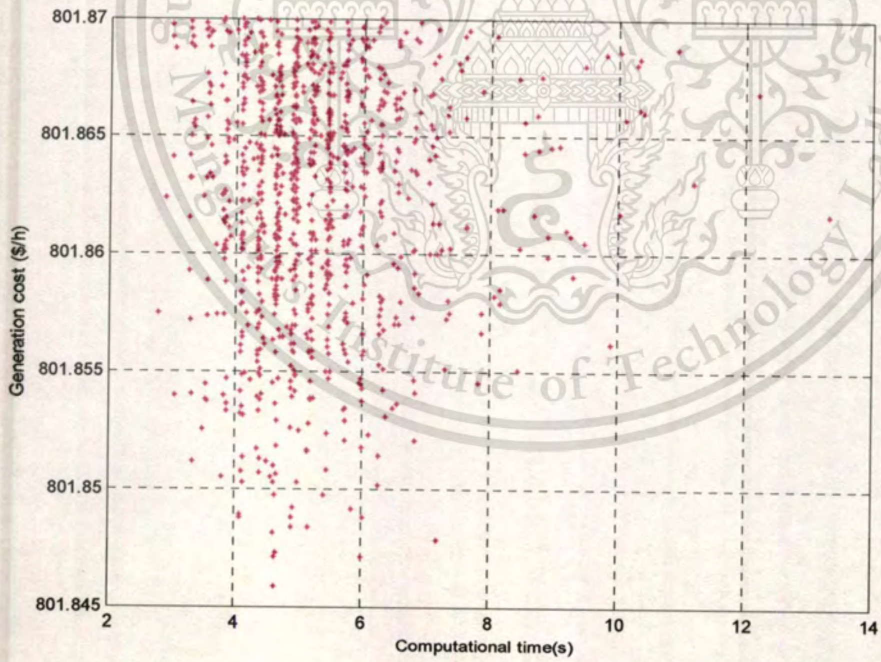


Figure 4.27 Solution vs. computational time profiles of 1000 trial runs of IEEE 30-bus system

Figure 4.25 give the performance of the proposed method in 100 different trial runs, this figure shows the best solution profiles of each group (g_{bests}) and the best solution among all groups (G_{best}). It can be seen that even though the trajectory of convergence is hardness the proposed method attempts to get g_{best} several values, thus the G_{best} should be chosen from the best g_{best} of each group and can aid it escapes from several local optimums lead to global optimum. Figure 4.26 depicts the zooming of solution profiles of Figure 4.25 for trial ranges between 80 to 100 trials. This process makes the DSPSO-TSA obtain generation cost, the computational time and standard deviation lower than other methods as depicted in Table 4.19 and Table 4.20.

Table 4.18 Characteristics of the generation units for IEEE 30-bus system

Bus	a	b	c	Pmin (MW)	Pmax (MW)
1	0.00375	2.00	0	50	200
2	0.01750	1.75	0	20	80
5	0.06250	1.00	0	15	50
8	0.00834	3.25	0	10	35
11	0.02500	3.00	0	10	30
13	0.02500	3.00	0	12	40

Table 4.19 shows the robustness of the proposed methodology is carried out for 100 trial runs with difference initial starting points (initial populations or particles), it can be observed that the simulation results of the proposed method had better than other method namely, BA, DE, GA, PSO and TSA in term of minimum generation cost, average generation cost and average executed time.

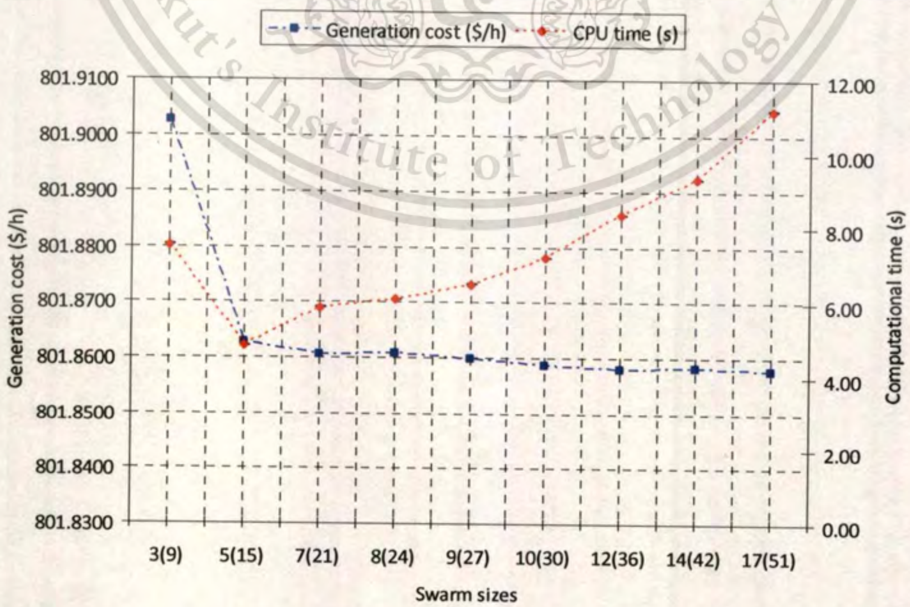


Figure 4.28 Shows results of empirical test of swarm size influences for IEEE 30-bus system

Table 4.20 depicts the best generation cost of several methods which compared with DSPSO-TSA. The results show that the proposed method gives minimum solution than the rest methods.

Table 4.19 Comparison of results of six methods of IEEE 30-bus system

Methods	Generation cost ($/h)			CPU	Standard
	Min.	Mean	Max.	times (s)	deviation (%)
BA	801.8484	801.8696	801.8770	10.15	0.00615
DE	801.8490	801.8633	801.8699	11.44	0.00522
GA	801.8539	801.8727	801.9050	17.88	0.00685
PSO	801.8701	801.8861	802.1843	8.13	0.03519
TSA	801.8517	801.8633	801.8700	24.78	0.00447
DSPSO-TSA	***801.8462***	***801.8616***	***801.8699***	***5.68***	***0.00520***

Table 4.20 Best result of six methods of IEEE 30-bus system

Power output (MW)	BA	DE	GA	PSO	TSA	DSPSO-TSA
P_1	177.138	177.032	176.075	176.680	177.240	177.588
P_2	48.959	48.821	50.010	48.347	48.913	48.791
P_3	21.397	21.254	21.490	21.790	21.651	21.325
P^5	21.192	21.474	21.659	21.497	21.490	21.779
P_8	12.222	12.361	12.226	12.167	11.658	12.079
P_{13}	12.037	12.005	11.305	12.313	12.008	12.024
Total power (MW)	292.946	292.947	292.764	292.795	292.959	293.585
Power loss (MW)	9.632	9.621	9.364	9.565	9.632	9.668
Total cost ($/h)	801.8484	801.8490	801.8539	801.8701	801.8517	***801.8462***

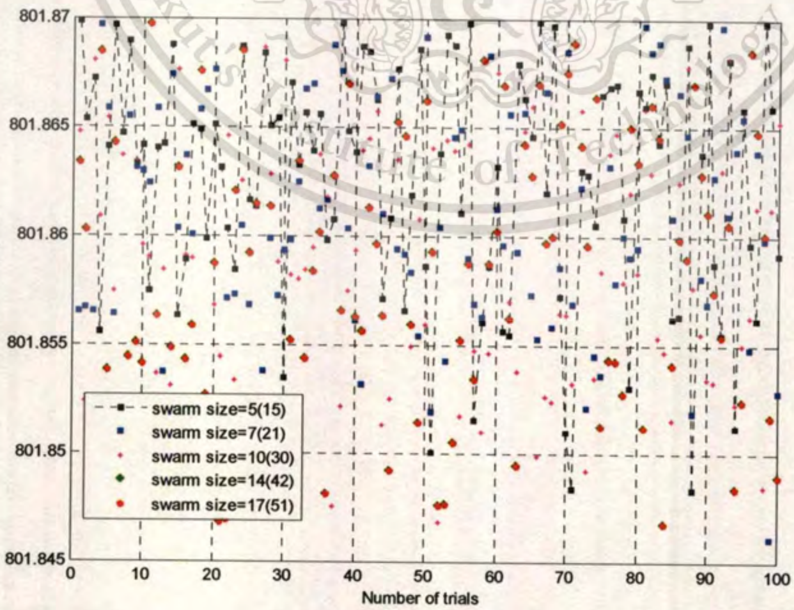


Figure 4.29 Illustrates the solution profiles vs. some selected swarm sizes of 100 trial runs for IEEE 30-bus system

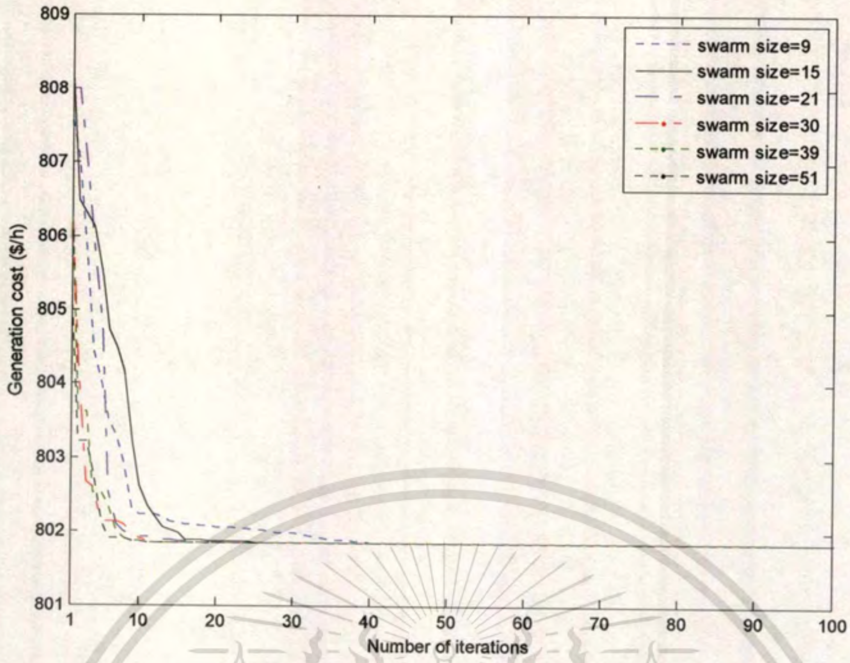


Figure 4.30 Convergences curve of some selected swarm sizes for IEEE 30-bus system

4.4.3 Voltage improvement of IEEE 30-bus system

The solution of OPF problem considers only fuel cost function may lead to infeasible solution that has unattractive voltage profile [80-81]. To improve voltage profile at load bus, it is treated at 1.0 per unit. Thus, the objective function is modified as Eq. (4.34) for amend voltage profile at load bus to near 1.0 per unit.

$$J = \sum_{i=1}^{NG} f_i + \alpha \sum_{i=1}^{NL} |V_i - 1.0| + \gamma_P (P_{G_1} - P_{G_1}^{\lim})^2 + \gamma_V \sum_{i=1}^{NL} (V_{L_i} - V_{L_i}^{\lim})^2$$
$$+ \gamma_Q \sum_{i=1}^{NG} (Q_{G_i} - Q_{G_i}^{\lim})^2 + \gamma_S \sum_{i=1}^{nl} (S_{l_i} - S_{l_i}^{\max})^2 \qquad (4.34)$$

Where α is penalty factor and set as 10, if voltage at load bus has digested from 1.0 p.u. penalty factor added to objective function. This effect lead objective function has little increased.

Table 4.21 Comparison result of six methods for voltage improvement of IEEE 30-bus system

Methods	Cost($/h) Min	Mean	Max	CPU time (s)	Standard deviation (%)
BA	806.443	806.756	809.655	72.53	0.31
DE	806.376	806.642	809.160	44.30	0.61
GA	806.654	811.183	819.408	92.68	2.39
PSO	806.379	807.396	811.535	40.23	0.89
TSA	806.367	806.746	806.865	62.61	0.11
DSPSO-TSA	806.357	806.614	809.275	27.93	0.46

Table 4.22 Best result of six methods for voltage improvement of IEEE 30-bus system

Power output (MW)	BA	DE	GA	PSO	TSA	DSPSO-TSA
P_1	174.92	174.92	176.47	175.91	175.52	174.55
P_2	48.66	49.27	48.65	49.13	48.13	48.68
P_3	21.71	20.89	21.39	22.03	21.90	21.64
P_5	22.76	21.73	21.75	20.11	22.32	21.75
P_8	13.09	12.68	11.97	14.10	11.69	13.08
P_{13}	12.03	13.58	12.95	12.00	13.80	13.31
C_{10}	11	8	18	12	34	13
C_{24}	22	18	17	19	15	15
tx_{4-12}	0.94	0.93	1.04	0.98	1.04	1.01
tx_{6-9}	1.07	1.07	1.01	1.00	0.97	0.95
tx_{6-10}	1.00	1.00	1.00	0.97	0.95	0.99
tx_{27-28}	1.02	1.02	1.01	1.01	0.97	1.05
V_1	1.04	1.05	1.05	1.04	1.04	1.04
V_2	1.01	1.01	0.99	0.99	1.05	1.04
V_5	1.04	1.04	1.00	1.02	1.05	0.98
V_8	1.00	1.01	1.00	1.02	1.01	1.00
V_{11}	1.00	1.00	1.06	1.00	1.00	0.99
V_{13}	1.03	1.03	1.04	1.02	0.98	1.02
Total power (MW)	293.17	293.06	293.18	293.28	293.36	293.00
Power loss (MW)	9.77	9.66	9.78	9.88	9.96	9.60
Total cost ($/h)	806.443	806.376	806.654	806.379	806.367	806.357

Table 4.22 depicts the best results obtained from 100 trial runs with different initial solutions. The results of the proposed method is compared with other optimization methods, obviously, the least generation cost is the DSPSO-TSA. Table 4.21 shows the robustness of the DSPSO-TSA compared to BA, DE, GA, PSO and TSA. Table 4.23 and Figure 4.31 show the voltage profile and voltage deviation of each method, voltage deviation obtain from the proposed method has lower than the rest methods. The results show that the DSPSO-TSA has good generation cost and least computational time than other methods. Figure 4.32 demonstrates convergence characteristic of the proposed method and zoomed to look the process of TSA for helping the DSPSO to find the global solution in the last searching process.

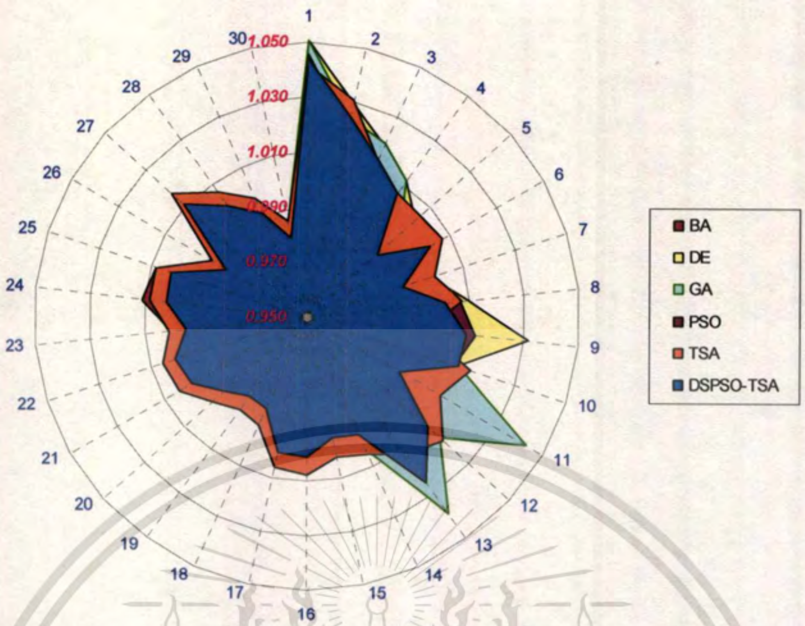


Figure 4.31 Characteristics of voltage profile obtained from six methods

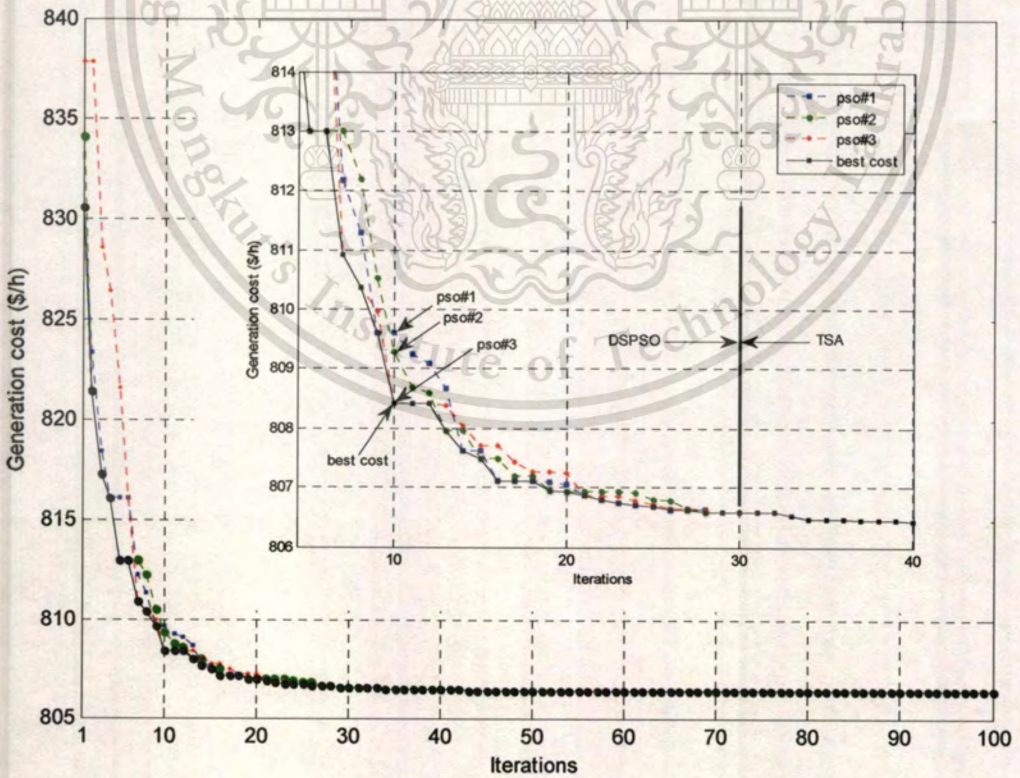


Figure 4.32 Convergence characteristics of the DSPSO-TSA for voltage improvement of IEEE 30-bus system

Table 4.23 Voltage profile of six methods

Buses	BA	DE	GA	PSO	TSA	DSPSO -TSA
1	1.038	1.050	1.050	1.045	1.041	1.044
2	1.020	1.029	1.023	1.020	1.030	1.024
3	1.011	1.019	1.019	1.012	1.013	1.013
4	1.005	1.011	1.011	1.005	1.006	1.006
5	1.000	1.002	0.995	1.002	1.005	0.984
6	1.000	1.006	1.007	1.003	1.007	1.001
7	0.992	0.997	0.994	0.995	0.998	0.986
8	1.000	1.010	1.000	1.006	1.007	1.000
9	1.026	1.031	1.008	1.012	0.992	1.009
10	1.008	1.007	1.008	1.006	1.013	1.008
11	1.000	1.001	1.042	1.002	1.007	0.989
12	1.011	1.011	1.015	1.010	1.017	1.009
13	1.034	1.027	1.038	1.017	1.011	1.025
14	0.999	0.998	1.002	0.997	1.004	0.997
15	0.998	0.997	1.000	0.996	1.002	0.995
16	1.002	1.002	1.005	1.001	1.008	1.001
17	1.001	1.000	1.001	0.999	1.006	1.000
18	0.989	0.988	0.990	0.987	0.993	0.987
19	0.987	0.986	0.988	0.985	0.991	0.985
20	0.991	0.990	0.992	0.989	0.996	0.990
21	1.000	0.999	0.999	0.998	1.004	0.999
22	1.002	1.001	1.001	1.000	1.005	1.000
23	1.000	0.997	0.999	0.996	1.001	0.994
24	1.011	1.006	1.007	1.006	1.008	1.002
25	1.008	1.007	1.007	1.006	1.008	1.002
26	0.990	0.989	0.989	0.988	0.990	0.984
27	1.015	1.016	1.016	1.015	1.017	1.011
28	0.999	1.006	1.004	1.003	1.006	1.000
29	0.995	0.996	0.996	0.995	0.997	0.991
30	0.983	0.985	0.985	0.983	0.986	0.979
Voltage Deviation (%)	0.0097	0.0105	0.0119	0.0084	0.0082	0.0088

4.5. Conclusion

This chapter shows the effectiveness and robustness of proposed method. Parameters tuning for obtains the best parameters of the new method, i.e. c_3 and c_4, particle size and procedure to obtain the G_{best} are demonstrated also. The effectiveness and robustness of the DSPSO-TSA are tested with the famous two optimization problems, i.e. economic dispatch problem and optimal power flow with various case studies. The results obtained from bee colony algorithm, differential evolution, genetic algorithm, particle swarm optimization and tabu search are compared with the proposed method.

Numerical simulation results shows that the DSPSO-TSA has high solution quality than the rest methods in term of average generation cost and computational time of convergence to the optimal solution.

Chapter 5
Conclusions

At present, a personal microcomputer has high speed computation power and low cost. Therefore, the concept of multiple computations can be implemented by using only a personal microcomputer for solving the large scale problems. Therefore, this thesis has proposed a new technique based on distributed searches of particle swarm optimization method. In addition, the new inertia factor, the elimination high cost and the search space reducing techniques are applied to improve the conventional search process.

This thesis proposes a new approach based on particle swarm optimization (PSO) and tabu search algorithm (TSA). This proposed approach is called distributed Sobol PSO and TSA (DSPSO-TSA). The DSPSO-TSA is applied to solve the multi-dimensional and multimodal numeric problems, four case studies of the economic dispatch problem (ED) with various kinds of the cost characteristic and optimal power flow (OPF) with two case studies. To enhance the global exploration, the conventional PSO is distributed and integrated with TSA. Moreover, the Sobol sequence is applied for generating the inertia factor of the PSO. Additionally, two techniques, i.e. the elimination high cost and the search space reducing techniques are implemented in the DSPSO-TSA also. The proposed method has compared with five optimization methods, i.e. bee colony algorithm (BA), differential evolution (DE), genetic algorithm (GA), particle swarm optimization (PSO) and tabu search algorithm (TSA).

Firstly, the performance of the DSPSO-TSA is tested by applying to solve the benchmark numeric function consist of multi-dimensional and multimodal functions as shown in chapter 3. In this chapter, the investigation performance of the proposed method and the simulation result comparing with BA, DE, GA, PSO and TSA are shown. In the simulations, mean value of function, standard deviation of solution and computational time of 100 runs with different random seed dimensions are reported. The results show that the DSPSO-TSA has outperformed than five methods on all functions and all dimensions.

Secondly, the effectiveness and robustness of the proposed method are tested with the ED problem and OPF problem. For the ED problem, the performance of the DSPSO-TSA is tested with four case studies three different types of cost function. Based on the simulation results which obtained from 100 different initial trials, the solution quality, reliability and computation efficiency shows the superiority of the DSPSO-TSA over other methods. The proposed method converges to the global solution, regardless of the shape of the cost function, i.e. discontinuous and non-smooth characteristic.

For the OPF problem, the performance of the DSPSO-TSA is tested with two test systems, six bus test system and IEEE 30-bus test system respectively. For the first test system, the new parameters, i.e. c_3 and c_4 are tuned for exploring the appropriate values. The numerical results of the proposed method are compared with five methods, i.e. BA, DE, GA, PSO and TSA. For the second test system, the performance testing is done with two case studies. The first case study considers only minimize the generation cost. The simulation figure shows the process to get the G_{best} and the influence of the particle swarm size relative to computational time and generation cost. The second case study is performed by considering both minimize generation cost and regulating voltage deviation at load bus. The voltages at load bus are treated at 1.0 p.u. for voltage security of the equipments in the substation. The proposed method yields the least generation cost and least standard deviation of voltage at load bus than the rest methods.

Finally, future works can be suggested to continue this research:

1. Implement the proposed method for other types of economic dispatch problem such as:

   - The ED problem with considers other types of cost function such as hydro-thermal ED problem, combine cycle co-generation plant or combined with generation cost of wind power plant.
   - The ED problem with considers power exchanged to another power system networks, i.e. multi-area ED problem.

2. Apply the proposed method to solve other types of objective function in the optimal power flow problem such as minimize power loss in transmission line or dispatching the reactive power in power system network.

3. Apply the DSPSO-TSA to solve power system optimization problems such as the unit commitment, the transmission line expansion planning and allocation of distributed generator (DG) problem.

References

[1] A. J. Wood and B. F Wollenberg, **Power generation, operation and control**. New York: John Wiley & Sons; 1984.

[2] J. Arrillaga and C. P. Arnold. **Computer analysis of power systems**. Great Britain: John Wiley & Sons; 1990.

[3] B. H. Chowdhury and S. Rahman, **"A review of recent advances in economic dispatch,"** IEEE Trans Power Syst; vol. 5, no. 4, pp 1248-59, 1990.

[4] F. N. Lee and A. M. Breiphol, **"Reserve constrained economic dispatch with prohibited operating zones,"** IEEE Trans Power Syst, vol. 8, pp 246-54, 1993.

[5] C. E. Lin and G. L. Viviani, **"Hierarchical economic dispatch for piecewise quadratic cost functions,"** IEEE Trans Power App Syst, vol. 103, pp 1170-5, 1984.

[6] J. B. Park, Y. S. Kim, I. K. Eom and K. Y. Lee, **"Economic load dispatch for piecewise quadratic cost function using hopfield neural network,"** IEEE Trans Power Syst, vol. 8, no. 3, pp 1030-8, 1993.

[7] K. Y. Lee, A.S. Yome and J. H. Park, **"Adaptive hopfield networks for economic load dispatch,"** IEEE Trans Power Syst, vol. 13, no. 2, pp 519-26, 1998.

[8] K. P. Wong and C.C. Fung, **"Simulate annealing base economic dispatch algorithm"** IEE Proc C, vol. 140, no. 6, pp 507-13, 1993.

[9] D. C. Walters and G. B. Sheble, **"Genetic algorithm solution of economic dispatch with valve point loading,"** IEEE Trans Power Syst, vol. 8, no. 3, pp 1325-32, 1993.

[10] K. P. Wong and Y. W. Wong, **"Genetic and Genetic/Simulated - Annealing approaches to economic dispatch,"** IEE Proc.Gener Transm. Distrib, vol. 141, no. 5, pp 507-13, 1994.

[11] C. L. Chiang, **"Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels,"** IEEE Trans Power Syst, vol. 20, no. 4, pp 1690-9, 2005.

[12] A. Bakirtzis, V. Petridis and S. Kazarlis, **"Genetic algorithm solution to the economic dispatch problem,"** Proc Inst Elect Eng-Gen Transm Dist., vol. 141, no. 4, pp 377-82, 1994.

[13] Z. L. Giang, **"Particle swarm optimization to solving the economic dispatch considering the generator constraints,"** IEEE Trans Power Syst, vol. 18, no. 3,

pp 1187-95, 2003.

[14] Z. L. Giang, **"Closure to Discussion of Particle swarm optimization to solving the economic dispatch considering the generator constraints,"** IEEE Trans Power Syst, vol. 19, no. 4, pp 2122-3, 2004.

[15] B. K. Panigrahi, V. R. Pandi and S. Das, **"Adaptive particle swarm optimization approach for static and dynamic economic load dispatch,"** Energy Convers Manage 2008. doi:10.1016/j.enconman.2007.12.023.

[16] C. Jiejin, M. Xiaoqian, L. Lixiang and P. Haipeng, **"Chaotic particle swarm optimization for economic dispatch considering the generator constraints,"** Energy Convers Manage, vol. 48, pp 645-53, 2007.

[17] S. Khamsawang, C. Boonseng and S. Pothiya, **"Solving the economic dispatch problem with tabu search algorithm,"** In: Proceeding of the IEEE International Conference on Industrial Technology 2002, Bangkok, Thialand, pp. 274-8, 2002.

[18] S. Khamsawang, C. Boonseng and S. Pothiya, **"Solving the economic dispatch problem with multiple tabu search algorithm,"** IEEE TENCON04 2004, Chiangmai, Thialand, pp. 274-8, 2004.

[19] S. Pothiya, I. Ngamroo and W. Kongprawechnon., **"Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints,"** Energy Convers Manage 2007. doi:10.1016/j.enconman.2007.08.012.

[20] B. K. Panigrah, S. R. Yadav, S. Agrawal and M. K. Tiwari, **"A clonal algorithm to solve economic load dispatch,"** Elect Power Syst Res 2006. doi:10.1016/j.epsr.2006.10.007.

[21] J. F. Kenedy and R. C. Eberhart, **"Particle swarm optimization,"** IEEE international conference on neural networks, vol.4, pp 1942-8, 1995.

[22] Y. Shi, **"Particle Swarm Optimization,"** IEEE Neural Networks Society February 2004

[23] M. Clerc and J. F. Kenedy, **"The particle swarm: Explosion, stability and convergence in a multi - dimensional complex space,"** IEEE Trans Evol Comput; vol 2, no.3, pp 91-6, 1998.

[24] S. Naka, T. Genti, T. Yura and Y. Fukuyama, **"Practical distribution state estimation using hybrid particle swarm optimization,"** IEEE Power Engineering Society Winter Meeting, vol. 2, pp 815-20, 2001.

[25] S. Naka, T.Genti, T. Yura and Y. Fukuyama, **"Hybrid particle swarm optimization based distribution state estimation using constriction factor approach,"** Proc Conference SCIS ISIS 2002; vol. 2, pp 1083-8, 2002.

[26] I. M. Sobol, **"On the distribution of points in a cube and the approximate evaluation of integrals,"** USSR computation mathematics and Mathematical physics, vol. 7, no. 4, pp 86-112, 1967.

[27] I. A. Antonov and V. M. Saleev, **"An economic method of computing LP-sequences,"** USSR computation mathematics and Mathematical physics, vol.19, no. 1, pp 252-6, 1979.

[28] P. Bradley and B. L. Fox, **"Algorithm 659 implementing Sobol's quasirandom sequence generator,"** ACM Trans Mathemativcal Software, vol. 14, no. 1, pp 88-100, 1988.

[29] S. Jiriwibhakorn and A. H. Coonick, B. J. Cory, **"Fast Critical Clearing Time Function Approximation Using Neural Networks and Sobol Sequences,"** IEEE Power Engineering Review 2000, vol. 20, no. 1, pp 51-3, 2000.

[30] S. Jiriwibhakorn and A. H. Coonick, **"Fast Critical Clearing Time Estimation of A Large Power System Using Neural Networks and Sobol Sequences,"** IEEE Power Engineering Society Summer Meeting 2000; vol. 1, pp 522-7, 2000.

[31] F. Glover, **"Tabu Search Part I,"** ORSA J. Comput, vol. 1 no. 3, pp 190-06, 1989.

[32] F. Glover, **"Tabu Search Part II,"** ORSA J. Comput; vol. l2, no. 1, pp 4-32, 1990.

[33] J. A. Bland and G. P. Dawson, **"Tabu Search and Design Optimization. Computer - aided design,"** vol. 23, no. 3, pp 195-01, 1991.

[34] B. Yang, Y. Chen and Z. Zhao, **"Survey on Applications of particle swarm optimization in electric power systems,"** 2007 IEEE International Conference on Control and Automation WeD3-3, Guangzhou, CHINA, 2007

[35] M. R. AlRashidia and M. E. El-Hawary, **"Applications of computational intelligence techniques for solving the revived optimal power flow problem,"** Electric Power Systems Research, vol. 79, pp 694–702, 2009.

[36] J. A. Momoh, R. Adapa and M. E. El-Hawary, **"A review of selected optimal power flow literature to 1993. I. Nonlinear and quadratic programming approaches,"** IEEE Transactions on Power Systems, vol. 14, no. 1, pp 96–104, 1999.

[37] J. A. Momoh, M. E. El-Hawary and R. Adapa, **"A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior**

point methods," IEEE Transactions on Power Systems, vol. 14, no. 1, pp 105–111, 1999.

[38] Z. L. Gaing, **"Constrained optimal power flow by mixed-integer particle swarm optimization,"** Proceedings of 2005 IEEE Power Engineering Society General Meeting, vol. 1, pp. 243-250, 2005.

[39] M. Huneault and F. D. Galiana, **"A survey of the optimal power flow literature,"** IEEE Transactions on Power Systems, vol. 6, no. 2, pp 762–770, 1991.

[40] B. Chao, C. X. Guo and Y. J. Cao, **"Improved particle swarm optimization algorithm for OPF problems,"** Proceedings of 2004 IEEE PES Power Systems Conference and Exposition, vol. 1, pp. 233-238, 2004.

[41] S. He, J. Y. Wen, E. Prempaint, Q. H. Wu, J. Fitch and S. Mann, **"An improved particle swarm optimization for optimal power flow,"** Proceedings of 2004 International Conference on Power System Technology, vol. 2, pp. 1633-1637, 2004.

[42] C. R. Wang, H. J. Yuan, Z. Q. Huang, J. W. Zhang and C. J. Sun, **"A modified particle swarm optimization algorithm and its application in optimal power flow problem,"** Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 2885-2889, 2005.

[43] A. Ahmed Esmin, G. L. Torres and C. Zambroni de Souza, **"A hybrid particle swarm optimization applied to loss power minimization,"** IEEE Transactions on Power Systems, vol. 20, no. 2, pp. 859-866, 2005.

[44] Z. L. Gaing and R. F. Chang, **"Security-constrained optimal power flow by Mixed-Integer,"** IEEE Power Engineering Society General Meeting, 2006.

[45] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas and.V. Petridis, **"Optimal power flow by enhanced genetic algorithm,"** IEEE Power Engineering Review, 2002.

[46] T. S. Chung and Y. Z. Li, **"A hybrid GA approach for OPF with consideration of FACTS devices,"** IEEE Power Engineering Review, pp. 47-50, Feb. 2001.

[47] M. Varadarajan and K.S. Swarup, **"Differential evolutionary algorithm for optimal reactive power dispatch,"** Electrical Power and Energy Systems, vol. 30, pp 435–441, 2008.

[48] S. Rahnamayan, H. R. Tizhoosh, and Magdy M. A. Salama, **"Opposition-based differential evolution,"** IEEE Transaction on Evol. Comp., vol. 12, no. 1, Feb. 2008.

[49] J. H. Holland, **Adaptation in natural and artificial systems: An introductory**

analysis with applications to biology, control, and artificial intelligence. University of Michigan, 1975.

[50] W. M. Spears, **Evolutionary algorithms: The role of mutation and recombination.** Berlin: Springer, 2000.

[51] W. S. Mohammed Elshamy, **Using artificial intelligence models in system identification.** A Thesis Submitted to the Faculty of Engineering at Cairo University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Power and Machines

[52] A. Georgieva and I. Jordanov, **"Global optimization based on novel heuristics, low-discrepancy sequences and genetic algorithms,"** European Journal of Operational Research, vol. 196, pp 413–422, 2009.

[53] C. Darwin, On the Origin of Species. London: John Murray, 1859, http: //embryology.med.unsw.edu.au/pdf/Origin of Species.pdf.

[54] C. Christober and R. Asir, **"Genetic algorithm based tabu search method for solving unit commitment problem with cooling-banking constraints"** Journal of Electrical Engineering, vol. 60, no. 2, pp 69-78, 2009

[55] D. E. Goldberg, Genetic **Algorithms in search optimization and machine learning.** Eastern Press, Addison-Wesley, 1999.

[56] M. Mitchell, **An introduction to genetic algorithms.** MIT Press, U.S.A., 2002.

[57] S. Pothiya, I. Ngamroo and W. Kongprawechnon, **Application of multiple tabu search algorithm to solve dynamic economic dispatch and load frequency control.** Ph.D. Thesis, SIIT 2009.

[58] R. Storn and K. V. Price, **"Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces,"** Technical Report TR-95-012, ICSI, http://http.icsi.berkeley.edu/~storn/litera.html, 1995.

[59] ____, **"Differential Evolution – a simple and efficient heuristic for global optimization over continuous spaces,"** Journal of Global Optimization, vol. 11, no. 4, pp 341–359, 1997.

[60] R. Storn, K. V. Price, and J. Lampinen, **"Differential Evolution - A practical approach to global optimization,"** Springer, Berlin: 2005.

[61] S. Das, A. Abraham, U. K. Chakraborty and A. Konar, **"Differential evolution using a neighborhood-based mutation operator,"** IEEE Transactions on Evolutionary Computation, Vol. 3, no. 3, June 2009.

[62] V. Tereshko and A. Loengarov, **"Collective decision-making in honey bee**

foraging dynamics," Computing and Information Systems Journal, vol. 9, no. 3, 2005.

[63] T. D. Seeley, **The wisdom of the hive: the social physiology of honey bee colonies.** Cambridge: Mass., Harvard University Press, 1996.

[64] S. Camazine, J. L Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz and E. Bonabeau, **Self-organization in biological systems.** Princeton: Princeton University Press, 2003.

[65] V. Tereshko and T. Lee, **"How information mapping patterns determine foraging behaviour of a honey bee colony,"** Open Systems & Information Dynamics, vol. 9, pp. 181-193, 2002.

[66] D. T. Pham, et al. **"The Bees Algorithm: A novel tool for complex optimisation problems."** 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006), Oxford: Elsevier, 2006.

[67] C. S. Chong and M. Y. H. Low, **"A Bee Colony Optimization Algorithnm to Job Shop Scheduling,"** 2006 Winter Simulation Conference, 2006.

[68] D. Karaboga nad B. Akay, **"Artificial Bee Colony (ABC), Harmony Search and Bees Algorithms on Numerical Optimization,"** IPROMS 2009 Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.

[69] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, **"Opposition-base differential evolution,"** IEEE Trans. On Evol. Comp., vol.12, no.1, Febuary 2008.

[70] D. Karaboga and B. Akay, **"Effect of region scaling on the initialization of particle swarm optimization, differential evolution and artificial bee colony algorithms on multimodal high dimensional problems,"** International Conference on Multivariate Statistical Modelling & High Dimensional Data Mining, Erciyes University, Kayseri: Turkey, 19-23 June 2008.

[71] M. G. H. Omran, A. P. Engelbrecht and A. Salman, **"Bare bone differential evolution,"** European journal of operation research, vol. 196, pp. 128-139, 2009.

[72] J. Carpentier, **"Contribution to the economic dispatch problem,"** Bull.Soc. Franc. Elect., vol. 8, no. 3, pp. 431–447, 1962.

[73] J. A. Momoh, R. J. Koessler, and M. S. Bond, **"Challenges to optimal power flow,"** IEEE Trans. Power Syst., vol. 12, no. 2, pp. 444–455, Feb.1997.

[74] K. Xie and Y. H. Song, **"Dynamic optimal power flow by interior point methods, "** Proc. IEEE, vol.148, no. 1, pp. 76–84, Jan. 2001.

[75] Q. H.Wu and Y. J. Cao, **"Dispatching"**, in Encyclopaedia of Electrical and Electronics Engineering, J. G. Webster, Ed. New York: Wiley, 1999.

[76] H. R. Cai, C. Y. Chung, and K. P. Wong, **"Application of differential evolution algorithm for transient stability constrained optimal power flow,"** IEEE Transaction on power System, vol. 23, no. 2, May 2008.

[77] J. D. Weber, **Implementation of a newton-based optimal power flow into a power system simulation environment**. M.Sc. Thesis, University of Illinois at Urbana-Champaign, 1997.

[78] O. Alsac and B. Scott, **"Optimal load flow with steady-state security,"** IEEE T-PAS, vol. 93, pp 745-751, 1974.

[79] L. Lakshminarasimman and S. Subramanian, **"Short-term scheduling of hydrothermal power system with cascaded reservoirs by using modified differential evolution,"** IEE Proc.-Gener. Transm. Distrib, vol. 153, no. 6, November 2006.

[80] M. A. Abido, **"Optimal power flow using particle swarm optimization,"** Electrical Power and Energy Systems, vol. 24, pp 563-571, 2002.

[81] C. A. Roa-Sepulveda and B. J. Pavez-Lazo, **"A solution to the optimal power flow using simulated annealing,"** Electrical Power and Energy Systems, vol. 25, pp 47-57, 2003.

Appendix

Appendix A

Table A.1 Generating unit data of 13 units system with valve-point effect

Unit	P_i^{max} (MW)	P_i^{min} (MW)	a_i (\$/h)	b_i (\$/MW)	c_i (\$/MW^2)	e_i (\$)	f_i (\$/MW)
1	0	680	550	8.1	0.00028	300	0.035
2	0	360	309	8.1	0.00056	200	0.042
3	0	360	307	8.1	0.00056	200	0.042
4	60	200	240	7.74	0.00324	150	0.063
5	60	200	240	7.74	0.00324	150	0.063
6	60	200	240	7.74	0.00324	150	0.063
7	60	200	240	7.74	0.00324	150	0.063
8	60	200	240	7.74	0.00324	150	0.063
9	60	200	240	7.74	0.00324	150	0.063
10	40	120	126	8.6	0.00284	100	0.084
11	40	120	126	8.6	0.00284	100	0.084
12	55	120	126	8.6	0.00284	100	0.084
13	55	120	126	8.6	0.00284	100	0.084

Table A.2 The generating unit capacity and cost coefficients of the 10 units system

Unit	Fuel types	P_i^{min} (MW)	P_i^{max} (MW)	a_i (\$/h)	b_i (\$/MW)	c_i (\$/MW^2)	e_i (\$)	f_i (\$/MW)
1	1	100	190	26.970	-0.397500	0.0021760	26.970	-0.397500
	2	196	250	21.130	-0.305900	0.0018610	21.130	-0.305900
2	1	50	144	1.865	-0.039880	0.0011380	1.865	-0.039880
	2	114	157	13.650	-0.198000	0.0016200	13.650	-0.198000
	3	157	230	118.400	-1.269000	0.0041940	118.400	-1.269000
3	1	200	332	39.7900	-0.311600	0.0014570	39.790	-0.311600
	2	332	388	-2.876	0.033890	0.0008035	-2.876	0.033890
	3	388	500	-59.140	0.486400	0.0000118	-59.140	0.486400
4	1	99	138	1.983	-0.031140	0.0010490	1.983	-0.031140
	2	138	200	52.850	-0.634800	0.0027580	52.850	-0.634800
	3	200	265	266.800	-2.338000	0.0059350	266.800	-2.338000
5	1	190	338	13.920	-0.087330	0.0010660	13.920	-0.087330
	2	338	407	99.760	-0.520600	0.0015970	99.760	-0.520600
	3	407	490	-53.990	0.446200	0.0001498	-53.990	0.446200
6	1	85	138	1.983	-0.031140	0.0010490	1.983	-0.031140
	2	138	200	52.850	-0.634800	0.0027580	52.850	-0.634800
	3	200	265	266.800	-2.338000	0.0059350	266.800	-2.338000
7	1	200	331	18.930	-0.132500	0.0011070	18.930	-0.132500
	2	331	391	43.770	-0.226700	0.0011650	43.770	-0.226700
	3	391	500	-43.350	0.355900	0.0002454	-43.350	0.355900
8	1	99	138	1.983	-0.031140	0.0010490	1.983	-0.031140
	2	138	200	52.850	-0.634800	0.0027580	52.850	-0.634800
	3	200	265	266.800	-2.338000	0.0059350	266.800	-2.338000
9	1	130	213	14.230	-0.018170	0.0006121	14.230	-0.018170
	2	213	370	88.530	-0.567500	0.0015540	88.530	-0.567500
	3	370	440	14.230	-0.018170	0.0006121	14.230	-0.018170
10	1	200	362	13.970	-0.099380	0.0011020	13.970	-0.099380
	2	362	407	46.710	-0.202400	0.0011370	46.710	-0.202400
	3	407	490	-61.130	0.508400	0.0000416	-61.130	0.508400

Table A.3 The generating unit capacity and cost coefficients of the six generator system

Units	P_i^{max} (MW)	P_i^{min} (MW)	a_i ($)	b_i ($/MW)	c_i ($/MW^2)
1	500	100	240	7.00	0.0070
2	200	50	200	10.0	0.0095
3	300	80	220	8.50	0.0090
4	150	50	200	11.0	0.0090
5	200	50	220	10.5	0.0080
6	120	50	190	12.0	0.0075

Table A.4 The ramp rate limits and prohibited operating zones of the six generator system

Unit	P_i^0 (MW)	UR_i (MW/h)	DR_i (MW/h)	Prohibited zones (MW)	
1	440	80	120	[210-240]	[350-380]
2	170	50	90	[90-110]	[140-160]
3	200	65	100	[150-170]	[210-240]
4	150	50	90	[80-90]	[110-120]
5	190	50	90	[90-110]	[140-150]
6	150	50	90	[75-85]	[100-105]

$$B_{ij} = \begin{bmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.0010 & -0.0006 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.0010 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.0150 \end{bmatrix} \tag{B1.1}$$

$$B_{oi} = 1.0\times10^{-3} \times [-0.3908 \quad -0.1297 \quad 0.7047 \quad 0.0591 \quad 0.2161 \quad -0.6635] \tag{B1.2}$$

$$B_{oo} = 0.056 \tag{B1.3}$$

Table A.5 The ramp rate limits and prohibited operating zones of the 15 generators system

Unit	P_i^{min} (MW)	P_i^{max} (MW)	P_i^0 (MW)	UR_i (MW/h)	DR_i (MW/h)	Prohibited zones (MW)		
1	150	455	400	80	120			
2	150	455	360	80	120	[185-225]	[305-335]	[420-450]
3	20	130	105	130	130			
4	20	130	100	130	130			
5	150	470	190	80	120	[180-200]	[305-335]	[390-420]
6	135	460	400	80	120	[230-255]	[365-395]	[430-455]
7	135	465	350	80	120			

Table A.5 The ramp rate limits and prohibited operating zones of the 15 generators system (cont')

Unit	$P_i^{\min}$ (MW)	$P_i^{\max}$ (MW)	P_i^0 (MW)	UR_i (MW/h)	DR_i (MW/h)	Prohibited zones (MW)		
8	60	300	95	65	100			
9	25	162	105	60	100			
10	25	160	110	60	100			
11	20	80	60	80	80			
12	20	80	40	80	80	[30-40]	[55-65]	-
13	25	85	30	80	80			
14	15	55	20	55	55			
15	15	55	20	55	55			

Table A.6 The generating unit capacity and cost coefficients of the 15 generator system

Unit	$P_i^{\min}$ (MW)	$P_i^{\max}$ (MW)	a_i (\$)	b_i (\$/MW)	c_i (\$/MW^2)
1	150	455	671	10.1	0.000299
2	150	455	574	10.2	0.000183
3	20	130	374	8.8	0.001126
4	20	130	374	8.8	0.001126
5	150	470	461	10.4	0.000205
6	135	460	630	10.1	0.000301
7	135	465	548	9.8	0.000364
8	60	300	227	11.2	0.000338
9	25	162	173	11.2	0.000807
10	25	160	175	10.7	0.001203
11	20	80	186	10.2	0.003586
12	20	80	230	9.9	0.005513
13	25	85	225	13.1	0.000371
14	15	55	309	12.1	0.001929
15	15	55	323	12.4	0.004447

$$B_{ij} = \begin{bmatrix} 0.0014 & 0.0012 & 0.0007 & -0.0001 & -0.0003 & -0.0001 & -0.0001 & -0.0001 & -0.0003 & 0.0005 & -0.0003 & -0.0002 & 0.0004 & 0.0003 & -0.0001 \\ 0.0012 & 0.0015 & 0.0013 & 0.0000 & -0.0005 & -0.0002 & 0.0000 & 0.0001 & -0.0002 & -0.0004 & -0.0004 & -0.0000 & 0.0004 & 0.0010 & -0.0002 \\ 0.0007 & 0.0013 & 0.0076 & -0.0001 & -0.0013 & -0.0009 & -0.0001 & 0.0000 & -0.0008 & -0.0012 & -0.0017 & -0.0000 & -0.0026 & 0.0111 & -0.0028 \\ -0.0001 & 0.0000 & -0.0001 & 0.0034 & -0.0007 & -0.0004 & 0.0011 & 0.0050 & 0.0029 & 0.0032 & -0.0011 & -0.0000 & 0.0001 & 0.0001 & -0.0026 \\ -0.0003 & -0.0005 & -0.0013 & -0.0007 & 0.0090 & 0.0014 & -0.0003 & -0.0012 & -0.0010 & -0.0013 & 0.0007 & -0.0002 & -0.0002 & -0.0024 & -0.0003 \\ -0.0001 & -0.0002 & -0.0009 & -0.0004 & 0.0014 & 0.0016 & -0.0000 & -0.0006 & -0.0005 & -0.0008 & 0.0011 & -0.0001 & -0.0002 & -0.0017 & 0.0003 \\ -0.0001 & 0.0000 & -0.0001 & 0.0011 & -0.0003 & -0.0000 & 0.0015 & 0.0017 & 0.0015 & 0.0009 & -0.0005 & 0.0007 & -0.0000 & -0.0002 & -0.0008 \\ -0.0001 & 0.0001 & 0.0000 & 0.0050 & -0.0012 & -0.0006 & 0.0017 & 0.0168 & 0.0082 & 0.0079 & -0.0023 & -0.0036 & 0.0001 & 0.0005 & -0.0078 \\ -0.0003 & -0.0002 & -0.0008 & 0.0029 & -0.0010 & -0.0005 & 0.0015 & 0.0082 & 0.0129 & 0.0116 & -0.0021 & -0.0025 & 0.0007 & -0.0012 & -0.0072 \\ -0.0005 & -0.0004 & -0.0012 & 0.0032 & -0.0013 & -0.0008 & 0.0009 & 0.0079 & 0.0116 & 0.0200 & -0.0027 & -0.0034 & 0.0009 & -0.0011 & -0.0088 \\ -0.0003 & -0.0004 & -0.0017 & -0.0011 & 0.0007 & 0.0011 & -0.0005 & -0.0023 & -0.0021 & -0.0027 & 0.0140 & 0.0001 & 0.0004 & -0.0038 & 0.0168 \\ -0.0002 & -0.0000 & -0.0000 & -0.0000 & -0.0002 & -0.0001 & 0.0007 & -0.0036 & -0.0025 & -0.0034 & 0.0001 & 0.0054 & -0.0001 & -0.0004 & 0.0028 \\ 0.0004 & 0.0004 & -0.0026 & 0.0001 & -0.0002 & -0.0002 & -0.0000 & 0.0001 & 0.0007 & 0.0009 & 0.0004 & -0.0001 & 0.0103 & -0.0101 & 0.0028 \\ 0.0003 & 0.0010 & 0.0111 & 0.0001 & -0.0024 & -0.0017 & -0.0002 & 0.0005 & -0.0012 & -0.0011 & -0.0038 & -0.0004 & -0.0101 & 0.0578 & -0.0094 \\ -0.0001 & -0.0002 & -0.0028 & -0.0026 & -0.0003 & 0.0003 & -0.0008 & -0.0078 & -0.0072 & -0.0088 & 0.0168 & 0.0028 & 0.0028 & -0.0094 & 0.1283 \end{bmatrix}$$

(B1.4)

$$B_{0i} = [-0.0001\ -0.0002\ 0.0028\ -0.0001\ 0.0001\ -0.0003\ -0.0002\ -0.0002\ 0.0006\ 0.0039\ -0.0017\ 0.000\ -0.0032\ 0.0067\ -0.0064]$$

(B1.5)

$$B_{00} = 0.0055$$

(B1.6)

Appendix B

List of Publications

International journal

[1] S. Khamsawang and S. Jiriwibhakorn. **"DSPSO–TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions,"** Energy Conversion and Management, vol. 51, pp. 365–375, 2010.

International conferences

[1] S. Khamsawang, P. wannakarn, S. Pothiya, and S. Jiriwibhakorn, **"Solving the economic dispatch problem by using differential evolution,"** 6th International Conference on Electrical Engineering / Electronics,Computer, Telecommunications and Information Technology (ECTI-CON 2009), Pattaya, Thailand, 6-9 May 2009.

[2] S. Khamsawang, P. wannakarn and S. Jiriwibhakorn, **"Hybrid PSO-DE for solving the economic dispatch problem with generator constraints,"** 2010 International Conference on Electrical and Energy Systems (ICEES 2010), Suntec Singapore International Convention & Exhibition Centre, Singapore, 26-28 February 2010.