Particle Swarm Optimisation with Applications in Power System Generation

A thesis Submitted for the Degree of Doctor of Philosophy

by

Pichet Sriyanyong

School of Engineering and Design
Brunel University

July 2007

ABSTRACT

Today the modern power system is more dynamic and its operation is a subject to a number of constraints that are reflected in various management and planning tools used by system operators. In the case of hourly generation planning, Economic Dispatch (ED) allocates the outputs of all committed generating units, which are previously identified by the solution of the Unit Commitment (UC) problem. Thus, the accurate solutions of the ED and UC problems are essential in order to operate the power system in an economic and efficient manner. A number of computation techniques have progressively been proposed to solve these critical issues. One of them is a Particle Swarm Optimisation (PSO), which belongs to the evolutionary computation techniques, and it has attracted a great attention of the research community since it has been found to be extremely effective in solving a wide range of engineering problems. The attractive characteristics of PSO include: ease of implementation, fast convergence compared with the traditional evolutionary computation techniques and stable convergence characteristic. Although the PSO algorithms can converge very quickly towards the optimal solutions for many optimisation problems, it has been observed that in problems with a large number of suboptimal areas (i.e. multi-modal problems), PSO could get trapped in those local minima, including ED and UC problems.

Aiming at enhancing the diversity of the traditional PSO algorithms, this thesis proposes a method of combining the PSO algorithms with a real-valued natural mutation (RVM) operator to enhance the global search capability and investigate the performance of the proposed algorithm compared with the standard PSO algorithms and other

algorithms. Prior to applying to ED and UC problems, the proposed method is tested with some selected mathematical functions where the results show that it can avoid being trapped in local minima. The proposed methodology is then applied to ED and UC problems, and the obtained results show that it can provide solutions with good accuracy and stable convergence characteristic with simple implementation and satisfactory calculation time.

Furthermore, the sensitivity analysis of PSO parameters has been studied so as to investigate the response of the proposed method to the parameter variations, especially in both ED and UC problems. The outcome of this research shows that the proposed method succeeds in dealing with the PSO's drawbacks and also shows the superiority over the traditional PSO algorithms and other methods in terms of high quality solutions, stable convergence characteristic, and robustness.

ACKNOWLEDGEMENTS

I am extremely grateful to my supervisor, Professor Yong-Hua Song, for his continuous guidance and encouragement throughout the duration of my research. I am also grateful to Professor Malcolm Irving, Dr. Hai Yan Lu, Dr. Ivana Kockar, Dr. Stephen McArthur, Dr. Gareth Taylor, Assistant Professor Pathom Attaviriyanupap, and Associate Professor Chao-Lung Chiang for their valuable comments and suggestions throughout my research.

I would like to thank Dr. Jeremy Daniel for his technical support on hardware and programming. I would also like to extend my thanks to Mr. Thanawat Nakawiro who has been helping me with the useful advice on my thesis.

I would like to express my heartfelt appreciation to the one I love ...the one I can always talk to...the one who always understands.

I gratefully acknowledge financial support from the Royal Thai Government and King Mongkut's Institute of Technology North Bangkok.

Last but not least, I would like to especially express my deepest gratitude to my parents for their love, support, and long-suffering listening to my difficulty throughout my Ph.D study.

DECLARATION

The work described in this thesis has not been previously submitted for a degree in this or any other university and unless otherwise referenced it is the author's own work.

STATEMENT OF COPYRIGHT

Table of Contents

List of Nomenclature

ARPSO	Attractive and Repulsive Particle Swarm Optimisation
BPSO	Basic Particle Swarm Optimisation Algorithm
BPSO-RVM	Basic Particle Swarm Optimisation with Real-Valued Natural Mutation
CBPSO	Original Particle Swarm Optimisation Including Inertia Weight and Constriction Factor
CBPSO-RVM	Original Particle Swarm Optimisation Including Inertia Weight and Constriction Factor combined with Real-Valued Natural Mutation
CEP	Classical Evolutionary Programming
CGA_MU	Conventional Genetic Algorithm with Multiplier Updating
CPSO	Constriction Factor Particle Swarm Optimisation Algorithm
CPSO-RVM	Advanced Particle Swarm Optimisation with Real-Valued Natural Mutation
DED	Dynamic Economic Dispatch
ED	Economic Dispatch
EP	Evolutionary Programming
FDP	Forward Dynamic Programming

FEP	Fast Evolutionary Programming
GA	Genetic Algorithm
GENCOs	Generation Companies
HM	Hierarchical Numerical Method
HPSO	Hybrid Particle Swarm Optimisation
IEP	Improved Evolutionary Programming
IFEP	Improved Fast Evolutionary Programming
IGA_MU	Improved Genetic Algorithm with Multiplier Updating
IPSO	Improved Particle Swarm Optimisation
ITS	Improved Tabu Search
LR	Lagrange Relaxation Method
LRS	Local Random Search Algorithm
LR-EP	Hybrid Method between Lagrange Relaxation and Evolutionary Programming
LR-CBPSO_RVM	Hybrid Method between Lagrange Relaxation and Particle Swarm Optimisation
MFEP	Modified Fast Evolutionary Programming
MHNN	Modified Hopfield Neural Network
MPSO	Modified Particle Swarm Optimisation
NM	Numerical Method
OPSO	Original Particle Swarm Optimisation Algorithm
PSO-CEP	Particle Swarm Optimisation with Classical Evolutionary Programming

PSO-GM	Particle Swarm Optimisation with Gaussian Mutation
PSO-LRS	Particle Swarm Optimisation with Local Random Search
NPSO	New Particle Swarm Optimisation
NPSO-LRS	New Particle Swarm Optimisation with Local Random Search
RVM	Real-Valued Natural Mutation
SQP	Sequential Quadratic Programming
TS	Tabu Search
TM	Taguchi Method
UC	Unit Commitment

List of Symbols

v_{id}^t	velocity of particle i at iteration t in d-dimensional space
$V_{d,\min}$	minimum velocity of particle
$V_{d,\max}$	maximum velocity of particle
x_{id}^t	current position of particle i at iteration t
$x_{i,mutate}^t$	mutated position of particle i at iteration t
w	inertia weight factor
t	number of iterations
n	number of particles in a group
m	number of members in a particle
k	constriction factor
c_1, c_2	acceleration constant
φ	summation of c_1 and c_2
$rand_1, rand_2$	uniformly distributed random number between 0 and 1
P_m	mutation probability
R_m	mutation rate
$gaussian(\sigma)$	a random number drawn from a Gaussian distribution
N_{run}	number of runs
N_{m}	number of particles that participate in mutation
$Iter$	total number of iterations

Pop	total number of population
TC	total production cost
$F_i(P_{it})$	fuel cost of generator i
ST_i	start-up cost of unit i
U_{it}	on/off status of unit i at hour t
P_{it}	generation output of unit i at hour t
R_{it}	spinning reserve of unit i at hour t
ρ_{SP_t}	forecasted spot price at hour t
ρ_{RP_t}	forecasted spinning reserve price at hour t
λ_t, μ_t	Lagrange multipliers
r	estimated probability that spinning reserve is called and generated
P_{Dt}	load demand at hour t
SR_t	spinning reserve at hour t
$P_{i,\min}$	minimum power output of unit i
$P_{i,\max}$	maximum power output of unit i
$T_{i,\text{up}}$	minimum up time of unit i
$T_{i,\text{down}}$	minimum down time of unit i
N	total number of generators
T	total number of hours
$T_{i,\text{off}}$	unit's off time

HSC_i	unit's hot start-up cost
CSC_i	unit's cold start-up cost
CSH_i	cold start hour
L	Lagrange function
J	primal value
q	dual value
ε	duality gap

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background of the Research

Unit Commitment (UC) is a problem in power system operation that determines the schedule of generating units to meet electricity demand and operating constraints over a time horizon. Basically, Economic Dispatch (ED), as a sub-problem of UC, determines the optimal scheduling of generation for a particular time that minimises the total production cost and satisfies equality and inequality constraints. Recently, a number of computation techniques, for example Simulated Annealing (SA), Genetic Algorithm (GA), Evolutionary Programming (EP), Tabu Search (TS) and Particle Swarm Optimisation (PSO) have been applied to solve these problems. Compared to other methods, PSO can solve the problems quickly with high quality solutions and stable convergence characteristics, while it is easily implemented. However, as other techniques PSO also face the problem associated with the lack of diversity in global search, as well as and the problem concerning the sensitivity of the fine tuning of its parameters. Up to now, a significant proportion of research still deals with developing of the PSO performance in order to solve complex optimisation problems. In addition, PSO algorithm has been commonly applied to various areas of engineering problem. Until now , only a few papers have focused on the application of PSO in power system operation, especially in ED and UC problems. Therefore, the aim of this thesis is to investigate application of the PSO algorithm to solve these two power system operation problems.

1.2 Aims of the Research

The main aims of this research are listed as follows:

- To enhance the performance of the traditional PSO algorithm, a hybrid method between the standard PSO and a real-valued natural mutation operator (RVM) is proposed.
- To illustrate its efficiency, the proposed method is applied to solve the ED problem with various types of cost functions characteristic, as well as to solve the traditional UC and a profit-based UC problems.

To validate the effectiveness of the proposed method, it is tested on a suite of mathematical benchmark functions. Moreover, it is compared to the traditional PSO methods and the existing hybrid PSO method (PSO with Gaussian Mutation). In this research, the proposed algorithm is aimed at increasing the diversity of particles in order to prevent being trapped in suboptimal points during search. The developed algorithm is expected to maintain the stability and reliability of its solutions, whereas the parameters are varied. In other words, it should be less sensitive to the change of the input parameters. In addition, it is expected that the proposed method is more powerful in power system generation applications than traditional PSO methods and some other methods, such as GA or EP techniques.

1.3 Contributions of the Research

The main contribution of this thesis is in the improvement of the standard PSO method, the application of the improved PSO in ED and UC problems as well as sensitivity analysis of various PSO techniques used for the solution of these two problems. The main original contributions developed in this thesis are outlined below:

- First, this thesis proposes the methodology where the traditional PSO algorithm is combined with a real-valued natural mutation operator (CBPSO-RVM). In order to validate its searching capability, the proposed methodology is tested with some selected mathematical benchmark functions, i.e. Sphere Function, Quadric Function, Griewank Function, Rastrigrin Function, and Ackley Function, respectively. It is found that the proposed method can generate better results compared with the traditional PSO algorithms and a hybrid method between PSO and Gaussian Mutation (PSO-GM).
- The proposed methodology is then applied to solve ED problem while considering four different characteristics of cost function, e.g. ED problem with smooth cost function, ED problem with multiple fuels, ED problem with valve-point loading, and ED problem with both multiple fuels and valve-point loading, respectively. In addition, a heuristic search method is adopted and modified so as to deal with the operating constraints. Aiming at the enhancement of the original method capability, the modified version has a better chance of generating feasible initial solutions within shorter computation time, while avoiding a repetition of calculation procedure. The simulation results clearly confirm that the proposed method is more powerful than other methods under consideration.
- Furthermore, this research utilises the proposed methodology to deal with both

the traditional UC problem and a profit-based UC problem considering various operating constraints, such as power balance, spinning reserve, operating limit, and minimum up/down time.

- For the application of the traditional UC, the proposed algorithm (CBPSO-RVM) is combined with Lagrange Relaxation (LR) method to improve the performance of LR in which CBPSO-RVM is applied for updating the Lagrange multipliers (λ_t, μ_t). In addition, a heuristic search method called the *Unit Decommitment* is then applied to improve solution obtained by this combined LR_CBPSO-RVM to eliminate excessive spinning reserve that will result in expensive total production cost. It can be concluded that the proposed method provides a satisfactory performance in terms of solution quality.
- Concerning the PSO application in the profit-based UC, the updating of Lagrange multipliers differs from the traditional UC in which CBPSO-RVM is only employed as updating the Lagrange multiplier (λ_t), whereas the Gradient method updates μ_t for enhancing the performance of the proposed method. It can be concluded from the simulation results that the proposed method with Gradient method can achieve the optimum solution.
- Finally, the thesis investigates the influences of parameter variations on both ED and UC problems. The simulation results clearly show that the proposed method provides higher robustness in both applications compared to the standard PSO algorithms.

1.4 Thesis Layout

The organisation of this thesis is as follows:

- Chapter 2 provides a brief introduction to evolutionary computation techniques in power systems. It also presents a brief review of recent works concerning the main categories of PSO and the applications of PSO in power systems, particularly in ED and UC problems.
- Chapter 3 presents the details of a hybrid algorithm between the PSO and a real-valued natural mutation (PSO-RVM). The hybrid PSO algorithms are then tested using a suite of five benchmark functions compared with the traditional PSO algorithms and an existing hybrid algorithm between PSO and Gaussian mutation.
- Chapter 4 applies the proposed method to ED problem with smooth and non-smooth cost functions and it is also compared with other methods for validating its ability.
- Chapter 5 presents the application of PSO algorithm in UC problem with various operating constraints: power balance, spinning reserve, operating limit, and minimum up/down time. Concerning the UC problem, both the traditional UC and the profit-based UC are investigated in this section.
- Chapter 6 studies the sensitivity analysis of PSO parameters in ED and UC problems. Aiming at investigating the influence of different parameters setting on the PSO algorithms, the simulations with various parameters settings are carried out.
- Chapter 7 states the conclusions and some suggestions for future work.

Chapter 2: Literature Review of PSO in Power Systems

2.1 Numerical Methods Applied in Power Systems

Security of supply is a critical issue in the operational planning of a modern power systems, considering the importance of the electricity in modern everyday life and economy. While electricity demand changes instantaneously during the course of a day, at the same time each generating unit itself has operating limits that need to be respected by system operators when deciding when and how much it needs to produce. Moreover, large-scale storage of electric energy is still difficult and uneconomical. Thus, it is a crucial role of electric utilities to maintain the reliability and continuity of electricity supply whilst providing least cost operation. To meet these contradictory objectives, a power system operator must deal with a number of dynamic issues. In the case of hourly generation planning, Economic Dispatch (ED) schedules the outputs of all committed generating units, which are previously identified by the Unit Commitment (UC) problem. Thus, the accurate solutions to the ED and UC problems are essential in order to operate the power system in an economic and efficient manner. Over time, a number of computation techniques have been proposed to solve these critical issues. These approaches can be classified into two main categories of classical and evolutionary computation techniques.

Concerning the UC problem, a number of classical methods have been adopted for solving this problem i.e. Extensive enumeration method, Priority list method, Dynamic programming method, Lagrange relaxation method, Mixed integer programming method, etc. The brief details of these approaches can be summarised as follows [1]:

- *Extensive enumeration method* takes all possible combinations or states of the schedule generating units into account. Although, this method can find the optimal solution, it will take a significant amount of calculation time[1].
- *Priority list method* is a simple and fast method for solving UC problem, based on ordering available generation according to its full-load average generation cost. It is very easy to implement and the transition among the states is very clear, which results in a decrease of the number of possible states. Because it is based on full-load average generation cost, the priority list method can get the optimal cost if the units are fully committed [1, 2].
- *Dynamic programming method* can achieve the optimal solution by building the decision tree. Namely, each possible path will be evaluated, compared, and stored to get the minimum cost. This method performs very well for small and medium systems; however, it suffer form the curse of dimensionality problem when solving large-scale problems. To overcome this, priority order can occasionally be adopted to reduce the possible combinations [1, 3].
- *Lagrange relaxation method* employs the dual optimisation approach, which is to maximise the dual function for comparing with the primal function. The processes will carry out until the termination criterion

(predefined duality gap) is satisfied. The main advantages of this method are fast and easy to solve, since it relaxes or ignores the coupling constraints of UC problem. Moreover, it decomposes the main problem into sub-problems that are easier to solve separately. On the other hand, the major drawback to LR is the lacking in high-quality solution as well as unsatisfactory convergence characteristic [1, 4].

- *Mixed integer programming method* solves the UC problem by adopting the principle of linearisation of cost curves and various constraints. Although this method shows a satisfactory performance especially in small problem size, it suffers from the problem of calculation time when it is applied to large-scale system [4, 5].

Regarding the ED problem, there are a number of traditional methods that have been applied to handle this problem such as: Lambda iteration method, Gradient method, Newton's method, etc. The overviews of these approaches can be summarised as follows [3]:

- *Lambda iteration method* is aimed at exploring the optimal lambda by using interpolation and extrapolation, whilst satisfying power demand constraint. In some cases, it can solve the problem very fast [3].
- *Gradient method* is superior to lambda iteration method in terms of limitation of cost curve characteristic. Its basic concept is to minimise total production cost by utilising the Lagrange function. However it still has problem with the violation of power balance constraint. To overcome this, the reduced gradient method is proposed with the concept of eliminating a variable [3].
- *Newton's method* improves the performance of the gradient method by

driving the gradient to zero. Namely, it can solve the ED problem within one step, but it is based on the quadratic cost function only [3].

Although these classical approaches perform very well, they are only intended for cases when the cost function is quadratic function or the incremental cost function is monotonically increasing. However, the cost function of realistic unit is more complicated [6, 7]. And there were some attempts to find the new methodology for dealing with this shortcoming.

In recent years, evolutionary computation techniques have been developed and proposed so as to solve a wide range of power system problems including ED and UC problems (i.e. Simulated Annealing, Genetic Algorithms, Evolutionary programming, Particle Swarm Optimisation, etc) [8]. In comparison with the classical methods, characteristics of evolutionary computation techniques that make them more attractive over the traditional ones are as follows:

- They are more likely to find a global solution, while the traditional methods may become trapped in a local optimum;
- There is no mathematical limitation of the problem formulation, while classical techniques may require approximations or specific cost function forms;
- Their calculation is based on random processes; therefore, they can generate many feasible solutions. This is in contrast to the conventional approaches that may yield only one solution [9].

Overviews of some of these computation methodologies are presented below.

2.2 Evolutionary Computation Techniques in Power Systems

Simulated Annealing

In the Iron Age, blacksmiths discovered that the formation of crystals in a solid is dependent on its cooling time; the slower the cooling, the more perfect the crystals form [10]. Simulated Annealing (SA) applies this idea in its computational algorithm. The basic principle of SA is that the control parameter of optimisation process is analogous to the "*Temperature*" of the metal in an annealing process. A change in the control parameter of optimisation process or the "*Temperature*" is then basically measured throughout an iterative computation. The transition at each iteration will automatically be accepted if the change in the objective function or the cost function is negative. The transition at iteration when the change in the objective function is positive will also be accepted, however if the rate of change of such the objective function remains within the *Boltzmann* based probability distribution. In this case, the additional procedure called "*Cooling Schedule*" is required to lower the "*Temperature*" and the computation will continue iteratively until it reaches the "*Freezing Temperature*", a condition where no further change in the "*Temperature*" occurs [10, 11].

Generally, the SA algorithm is able to deal with arbitrary systems. It is based on a local search technique and is regarded as a powerful method in terms of its ability to find a near global optimal solution. When combined with a probabilistic approach, SA is also able to find a solution outside a local optimum [12, 13]. However setting the parameters for SA is difficult and the computation speed will be slow when the method is applied to complicated power systems [13].

Genetic Algorithms

The Genetic Algorithm (GA) technique is based on a stochastic global search method which mimics some of the processes of natural evolution and selection [12]. The principal idea of this algorithm comes from the natural world where each species is required to adapt to a complicated changing environment so that it can maximise the likelihood of its survival. The characteristics of each species are encoded in its chromosomes, which continually transform when reproduction occurs. Over a period of time, the changes in these chromosomes give rise to species that are more likely to survive, and thus have a greater chance of passing their improved characteristic onto future generations [10]. The GA is analogous to the idea of chromosomes in nature where the computation method identifies candidate solutions which are encoded by a finite bit string [12]. Each chromosome exchanges information through a naturally random process so that solutions can evolve to be close to the optimum. The sequence of calculations will continue and repeat until termination conditions are satisfied. The strength of a GA is that it only requires information of the objective function. Thus, a GA can deal with a non-smoothing, discontinuous and non-differentiable function [12]. Since the computation of GA requires encoding and decoding schemes, it takes a longer time to reach an optimal solution. Sometimes, it is found that a GA can have a problem with its computation efficiency and convergence [10].

Evolutionary Programming

The fundamental concept of Evolutionary Programming (EP) similar to GAs in that it maintains populations of potential solutions and uses a mechanism to select the optimum from a set of those populations [8]. Rather than using generic specific operators as observed in nature as a GA does, EP sets its control parameter from real values of the problem that will be investigated. In addition, EP primarily bases its algorithm on mutation and selection while GAs traditionally use crossover [14].

Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is one of the modern algorithms used to solve global optimisation problems [15], and it is based on similar principles as the previous methods. Thus, to solve an optimisation problem, PSO applies a simplified social model, which for instance Zoologists might use to explain the movement of individuals within a group [16]. To begin with, PSO initialises a population of random solutions each of which is defined as a "*particle*". Initially, every particle flies into a problem hyperspace at a random velocity. Thereafter, each particle adjusts its travelling speed dynamically corresponding to the flying experiences of itself and its colleagues [8, 13]. The PSO computation will keep updating the position of the particles until it finds a global optimal solution.

Compared to other methods, application of the PSO is simple to implement, it can quickly find a number of high quality solutions, and has stable convergence characteristics [8, 17]. In addition, PSO is robust in solving continuous non-linear optimisation problems, and contrary to other evolutionary algorithms it has a flexible and well-balanced mechanism for improving and adjusting the global and local search capabilities [18].

However, PSO does have some drawbacks in that the algorithm seems sensitive to the tuning of some of its weights or parameters. In addition, PSO can sometimes suffer from the lack of the diversity amongst the particles, which can lead to a stagnation stage [19]. Therefore, although PSO has been a subject of an extensive research, there is a number of issues that need to be addressed in order to exploit the full potential of PSO in solving complex power system problems [17]. One of the objectives of this thesis is to contribute to this research area and developed a new improved hybrid PSO algorithm.

2.3 Main Categories of PSO Research Areas

Recently, a number of studies have been conducted to develop suitable PSO algorithms that can be used to solve complex problems in various applications. These studies looked at different aspects of PSO improvements, and according to Eberhart and Shi [20], they can be classified into the following five main categories :

(1) ***Algorithm development*** - the original PSO algorithm was principally developed in order to solve non-linear continuous optimisation problems [8, 21], however, a discrete binary version of PSO [22] was introduced subsequently to solve non-linear discrete optimisation problems. Moreover, PSO algorithms can be divided into the global version (Gbest model) and the local version (Lbest model) types, with the ability of the Lbest model to prevent a solution being trapped in local minima. The Gbest model, on the other hand, has more chance to get trapped into a local optimum. However, the global version is superior to the local version in terms of the speed of convergence to the optimum solution and the computation time [8, 20, 23]. The global version will therefore be taken into account in this thesis.

(2) ***Configuration of topology*** - the aspect of a neighborhood topology [24-27] examines the effect of different configurations or structures on PSO algorithm, i.e. circle topology, wheel topology, star topology, etc.

(3) ***Parameters*** - as mentioned before, PSO is sensitive to the tuning of its parameters; therefore, proper setting of the parameters can significantly

improve the searching capabilities of PSO methods [28]. Shi and Eberhart [29, 30] primarily introduced the incorporation of inertia weight factor (w) into the original PSO in order to balance the global and local explorations. Further, Clerc [31, 32] proposed an application of a constriction factor (k) to guarantee the convergence of the PSO.

(4) ***Hybrid PSO*** - often, PSO methodologies utilise the operators in the same manners as they are used in evolutionary computation techniques (i.e. selection, crossover and mutation) so as to avoid the stagnation problem that is a result of plunging into the suboptimal areas [20].

Angeline [33] applied a standard selection operator, which was generally used in the evolutionary computations, to enhance the performance of PSO. Its basic concept can be summarised in the principle that the worse particles will be replaced with the copy of better particles for the next generation. From the above concept, this method performs indubitably well in the uncomplicated problem (i.e. unimodal function), since it is aimed at improving the performance throughout the whole calculation procedure. On the other hand, it will perform disappointingly in complicated problems, such as multimodal functions, since this technique has a problem in the absence of searching diversity [34].

In [35], Lovbjerg *et al.* also incorporated a crossover operator into PSO in which the new particle will be generated by a pair of particles. However, this work still has a drawback with respect to its capacity to produce a number of diversify solutions [34]. Generally, in GAs, crossover is used to guide the population to the global solution, while mutation has an ability to explore the new undiscovered areas [36]. Like GA, PSO is normally guided by the cognitive part and the social part, yet the need for enhancing the diversity of swarm has to be improved [36]. Thus, applying mutation to PSO will lead to increase in its searching capability [37]. To overcome the deficiency of diversity, mutation operators have been extensively integrated into the traditional PSO algorithm [20, 23, 28].

Up till now, various mutation operators have been adopted in order to improve PSO's performance, for example Xie *et al.* [38] aimed to improve the performance of PSO by applying a random mutation operator to the standard PSO. Later, Zhang and Xie [39] presented a hybrid method between PSO and differential evolution operator. Furthermore, the Gaussian mutation operator, which is frequently used in GA, was incorporated with PSO by Higashi and Iba [34], while Stacey *et al.* [19] applied Cauchy distribution to the standard PSO. In addition, non-uniform mutation operator that was originally proposed by Michalewicz [40], was adopted and combined with PSO as presented in [23]. The outcomes reveal that applying non-uniform mutation operators to PSO increases the PSO capability, while solving simple unconstrained and constrained optimisation problems.

In [37], Ting *et al.* employed a new class of operators for improving convergence speed of PSO, i.e. single dimension mutation, differential mutation, log mutation, etc. Instead of choosing particle's positions in a mutation, Ratnaweera *et al.* [36] preferred to mutate the velocities. Further, Li *et al.* [41] proposed a modified PSO with mutation operator by re-random both positions and velocities of the particles. The mutation process will be carried out whenever the best position among all the particles (*gbest*) continues to the stagnation state for a number of iterations. Since there has been a significant interest in combining mutation operators with the standard PSO, it was important to compare searching capability of those hybrid methods. Such comparison was presented by Andrews [23], who has investigated the effects of using different mutation operators in both unconstrained and constrained optimisation problems. In his implementation, the positions of the swarm rather than velocities have been mutated. The simulation results illustrated that there were some improvements of PSO's searching ability, especially in multimodal functions and constrained optimisation problems, however, for unimodal functions its performance was worse than the standard PSO. Nevertheless, it has been concluded that selection of the mutation operators should be based on the nature of the problem. Whereas the proposed hybrid algorithms from the literature review show a success in enhancing the performance of the standard PSO, they is still a need to look for the new more efficient and robust algorithms.

Recently, Zhang and Lu [42] has successfully applied a new real-valued mutation operator to hybrid real-coded genetic algorithm with a quasi-simplex technique in order to cope with the lack of population diversity in global search, whilst the quasi-simplex technique has been used to enhance and guarantee the ability of local search. To verify the effectiveness of their proposed method, both unimodal and multimodal functions have been taken into account. In addition, two groups of multimodal functions (multimodal functions with few and several local optimal solutions) have also been investigated. From their simulation results, the proposed method has shown the effectiveness of searching the optimal solution not only for unimodal but particularly for multimodal functions. It can be therefore concluded that the application of the real-valued mutation operator a can improve the global search performance of the GA.

From the literature review, it can be seen that PSO with mutation operators can effectively optimise a wide range of engineering problems. This research intends to enhance PSO performance by adding the real-valued mutation operator (RVM), which is initially proposed and justified in [42], into the traditional PSO algorithms. As the results presented in Chapter 3 show, such modification has contributed to an increase in the global search diversity. In addition, the positions of PSO are widely used to mutate compared to the velocities, and this will also be taken into consideration in this thesis.

(5) ***Applications*** - Due to easy implementation with less computation time [21], PSO has been extensively applied to a wide variety of the problems, i.e. engineering optimisation problem with constraints [43], multi-objective optimisation problems [44], etc. This thesis aims at developing and applying the hybrid PSO algorithm to the applications of some optimisation problems in power systems, namely for solving an Economic Dispatch and Unit Commitment problems. Thus, the further section will focus on the applications of PSO in the area of power systems.

2.4 The Review of PSO Applications in Power Systems

Until now, substantial efforts related to the applications of PSO to various areas in power systems have been carried out. In [21, 45], the authors have summarised the development of PSO associated with the different areas of power systems, for example optimal power flow [18, 46, 47], transmission planning [48, 49], reactive power optimisation [15, 50, 51], load forecasting [52], power system controller design [53, 54], generation expansion planning [55, 56], etc. Regarding ED and UC problems, they are essential tools in managing and planning power system operation, and therefore there is need to solve these problems in an effective and efficient manner. The survey of the PSO applications in ED and UC problems are briefly summarised in the following sub-section.

2.4.1 Application of PSO in ED problem

For simplicity, the application of PSO in an ED problem will be divided into two main groups according to PSO algorithms themselves: (*A*) the traditional PSO algorithms and (*B*) the modified PSO algorithms.

A. The traditional PSO algorithms for ED problem

In [6], Gaing employed the standard PSO algorithm in order to solve the ED problem whilst observing a range of realistic constraints, e.g. power balance, generation limits, prohibited operating zone, and line flow constraints. To validate the proposed method performance, GA has been compared with respect to both solution quality and computation time. From the simulation results, the proposed PSO method succeeded in achieving higher quality solutions with less computation times compared to GA.

Then, Gaing [57] extended his work for solving Dynamic Economic Dispatch (DED) problem that is more complicated than the traditional ED problem. In general, DED is aimed at scheduling the output of generators over the scheduled time period subjected to not only generators constraints as mentioned in [6] but also spinning reserve constrains.

Zhao *et al.* [58] utilised an idea of the incorporation of constriction factor into the standard PSO algorithm so as to address the DED problem in a competitive electricity market. Further, Park *et al.* [16] has presented a new method to solve ED problem with non-smooth cost functions that follow from valve-point loading and multiple fuels effects. In this research, the standard PSO has been adopted and also incorporated with a modified heuristic search for manipulating the equality and inequality constraints. Additionally, the dynamic space reduction strategy has been proposed so that it can enhance the convergence speed during searching period.

In this article [59], Jeyakumar *et al.* have implemented the standard PSO for optimising a variety of ED problems, e.g. ED problem considering prohibited operating zone, multiple fuel effects, environmental constrains, and multi-area dispatch, respectively.

B. The modified PSO algorithms for ED problem

In [13], Victoire and Jeyakuma proposed the combination of the standard PSO and the Sequential Quadratic Programming (SQP) for non-smooth cost function with valve-point loading. At each iteration, the PSO has been initially used to explore the optimum solution of the ED problem, and then the optimum solution from the first step has been taken as the initial input for SQP, which has been then employed to fine tune the final solution. There are some similarities between this work [13] and the subsequent work published by the same authors [60], in which a hybrid PSO-SQP has been implemented to solve the reserve constrained DED problem by taking valve-point loading into consideration. However, it differs from the previous research in respect to the calculation processes, namely when the particles have started the stagnation stage with a number of predefined iterations, the "*crazy*"[1] particles will be generated under the concept of re-randomisation of the velocities. This process will

[1] *Crazy particle* is re-initialisation the velocities of the particle randomly when a random number (0,1) is less than or equal to the predefined probability.

eventually increase the diversity of the particles.

Recently, Swarup and Kumar [61] have modified the standard PSO based on a constriction factor version by using an attractive and repulsive PSO (ARPSO), and applied it to ED problem by taking both line-flow and voltage constraints into account. Their work shows that applying ARPSO prevents the particles from the loss of diversity in the swarm by means of the diversity factor which impose whether the particles' velocity should be adjusted or not.

According to [7], Park *et al.* have extended the above research in the same direction as work published in [16] that has been discussed in the previous section. In the case of [7], only the standard PSO has been modified for improving its global search capability by integrating a chaotic sequences technique into PSO for adjusting the inertia weight factor (w).

Recently, Selvakumar and Thanushkodi [62] have introduced a new modified PSO algorithm (NPSO) for ED problem considering a number of constraints. Regarding the concept of this algorithm, the second component of velocity's equation, called *cognitive component*, has been modified by adding another component named *bad experience component*. This component is applied to PSO to account for the worst positions of the particles, and it will help in finding the undiscovered areas. In addition, a simple local search algorithm (LRS) has also been incorporated into NPSO for enhancing its searching ability.

2.4.2 Application of PSO in UC problem

In [63], Ting *et al.* analysed the traditional Unit Commitment (UC) problem subject to the operating constraints by using a hybrid PSO where a combination of the standard PSO algorithm (real-valued version) and the binary version of PSO was proposed. Concerning the hybrid PSO, the binary version of PSO algorithm was employed to deal with UC problem, whereas the real-valued version was applied for solving ED problem. To investigate the efficiency of the proposed method, four different scenarios were presented as follows: (1) the standard PSO, (2) the standard PSO with differential mutation, (3) the standard PSO with linearly decreasing inertia weight factor (w), and (4) the standard PSO without re-initialisation when the violations of the constraints occur. The simulation results shown that the standard PSO with differential mutation (2nd scenario) yielded the better results compared to other scenarios. However, their simulation results in all scenarios did not satisfy the spinning reserve constraint.

According to [17], Gaing also proposed the combination of the binary version of PSO and lambda-iteration method for addressing the UC problem considering various constraints, for example power balance, spinning reserve, generation limit, and minimum up/down time constraints. It differs from Ting's work [63] in that lambda-iteration method was adopted to solve ED problem instead of the standard PSO. Furthermore, Balci and Valenzuela [64] presented another hybrid method called PSO-LR where PSO was combined with Lagrangian Relaxation (LR). It was aimed at improving the performance of the LR method by applying PSO for updating the Lagrange multipliers.

In [65], the both traditional UC and profit based UC has been investigated by Victoire and Jeyakumar. The proposed PSO-SQP method, which is based on exactly the same concept as presented in their prior research [13], has been only used to solve the ED problem (sub-problem of UC problem), whilst the Tabu search (TS) method has been utilised as the main algorithm for solving UC problem.

Recently, Ting *et al.* [66] have modified their hybrid method [63] by developing a heuristic search method that has prevented the solutions from violating the constraints before solving ED problem. Nonetheless, the main concept of the hybrid method between the binary version and the real-valued version of PSO algorithm has remained unchanged. In addition, the investigation of parameter variations has been carried out in order to choose the proper parameters for obtaining the high quality solutions.

In [67], Zhao *et al.* have proposed an improved PSO (IPSO) algorithm to increase the PSO searching performance for the UC problem. In IPSO, the global search ability is enhanced by using the data amongst the additional particles that will contribute to increase the chances of discovering the global optimum. Moreover, the parameter updating has been performed by a proposed adaptive approach.

2.4.3 Applications of PSO with Mutation Operators in ED and UC Problems

The hybrid PSO with mutation operations are not only commonly used for solving the mathematical problems but also employed in many other areas, for example sequencing problem [68], traveling salesman problem [69], navigation of mobile robot problem [70], control problem [71], scheduling problem [72], transportation problem [73], bin packing problem [74], etc. Although PSO algorithm with mutation technique been successfully applied to various areas of engineering problem, there are a small number of researches that focus on the area of power system operation especially, in ED and UC problems. The survey of the application of using hybrid PSO and mutation techniques for ED and UC problems can therefore be presented as shown below.

A. PSO with mutation for ED problem

Apart from the standard PSO, a number of techniques are still being developed for improving PSO performance to address ED problem. Using mutation operator for ED problem is one of the most accepted methods to enhance the searching ability of PSO.

In [75], Sinha and Purkayastha presented a hybrid technique between the standard PSO and the classical evolutionary programming (PSO-CEP) for ED problem with non-smooth cost functions. Regarding the main concept of their method, PSO was aimed at increasing the convergence ability, while Gaussian mutation had improved the searching diversity of the algorithm. Later, Hou *et al.* [76] has also utilised the advantages of mutation operator to guarantee global search capability. Moreover, they have also introduced another operator called *neighborhood magnifying operator* that has been used to prevent plunging into the local minima, thus improving the convergence accuracy.

B. PSO with mutation for UC problem

The concept of hybrid method between PSO and the mutation technique has been also applied to other areas in power systems, e.g. transmission planning [77], reactive power optimisation [78], etc. Regarding the UC problem, only a limited work has been carried out in applying the mutation technique to PSO. For example, Xiaohui *et al.* [79] have applied a modified PSO to profit-based UC problem. The concept of the proposed method is somewhat similar to Gaing's work [17] in which the binary version of PSO is used as the main algorithm for the UC problem, whereas the ED problem is solved by the standard PSO (real-valued version) in place of the lambda-iteration method. Moreover, they have also introduced *swap mutation operator* in order that the higher priority of the generating unit will be dispatched concerning the full-load average production cost.

From the above survey of literature, it can be observed that although the PSO with real-coded mutation have been successfully applied to various areas of engineering problem, there is still a limited research that is focused on the area of power system operation. For this reason, there is still a significant room for the research into the development and application of PSO algorithms for solving the problems in power system operation, including ED and UC. The aim of this thesis is to contribute to this area by proposing and investigating the application of hybrid method that combines PSO and the real-valued mutation operator to the solution of ED and UC problems. As analysis and results presented in Chapters 4 and 5 show, the here proposed hybrid model increases the swarm diversity, while maintaining the stability and reliability of the PSO.

Chapter 3: A Hybrid Algorithm between PSO and Real-Valued Natural Mutation (PSO-RVM)

3.1 Introduction

This chapter presents a method of combining the Particle Swarm Optimisation (PSO) algorithms with a Real-Valued Natural Mutation (RVM) [42] so as to enhance the global search capability and investigate the performance of the hybrid PSO algorithms compared with the traditional PSO algorithms. In addition, another hybrid method between PSO and Gaussian Mutation (PSO-GM) [34] is re-implemented in order to investigate and compare its performance with the here proposed method.

These algorithms are tested using a suite of five benchmark functions that are chosen from two different groups of the benchmark function. According to the classification of benchmark functions, both unimodal and multimodal functions have been classified and adopted to evaluate the performance of PSO algorithms [23, 27, 36, 80, 81]. The unimodal functions are the Sphere function (designated here as f_1) and the Quadric function (f_2) whilst the Griewank function (f_3), the Rastrigrin function (f_4), and the Ackley function (f_5) are the multimodal functions. The organisation of this chapter is as follows: section 3.2 presents the overview of PSO algorithms, while in section 3.3 a real-valued natural mutation is introduced. Section 3.4 illustrates the implementation of the hybrid PSO algorithms with the real-valued natural mutation operator. Simulation results will be shown and discussed in section 3.5. Finally, a summary is made in section 3.6.

3.2 PSO Algorithms

In 1995, Kennedy and Eberhart introduced a new evolutionary computation technique called *Particle Swarm Optimisation* (PSO) [82]. Similar to other evolutionary computation techniques, PSO employs the principle of a random initialised population and the concept of evaluation and modification of a population to find the optimal solution. In contrast, PSO does not utilise the operators during the modification step (e.g. mutation and crossover) as a GA does, since it can update itself [43],[83].

Mathematically, the fundamental model of PSO can be expressed by the following [84]:

Let a swarm have n particles in a d-dimensional search space. At the t^{th} iteration, $x_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{id}^t)$ expresses the position of the i^{th} particle and $pbest_i^t = (pbest_{i1}^t, pbest_{i2}^t, \ldots, pbest_{id}^t)$ shows the best previous position of the i^{th} particle. In addition, the best position among all the particles is represented by $gbest_d^t = (gbest_1^t, gbest_2^t, \ldots, gbest_d^t)$. The velocity of the i^{th} particle can be represented by $v_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{id}^t)$. Each of the population, called a *particle* or *agent*, can be updated or changed to the new position according to the current velocity, the difference between the current position and the best value itself (*pbest*) and its group (*gbest*) [85].

There are a number of algorithms [83] used to update the velocity of the i^{th} particle, and they are discussed below.

3.2.1 Original PSO algorithm (OPSO)

In this case, the modified velocity can be calculated from:

$$v_{id}^{t+1} = v_{id}^{t} + c_1 \times rand_1 \times (pbest_{id} - x_{id}^{t}) + c_2 \times rand_2 \times (gbest_d - x_{id}^{t}) \tag{3.1}$$

where the values of both c_1 and c_2 are set to a value of 2, while both $rand_1$ and $rand_2$ are random numbers between 0 and 1.

3.2.2 Basic PSO algorithm (BPSO)

As the OPSO does not adapt the velocity step size, it may lead to a poor searching ability. Consequently, BPSO utilises an inertia weight (w) in order to balance the global and the local searches. The updated velocity in the BPSO is calculated by:

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \times rand_1 \times (pbest_{id} - x_{id}^{t}) + c_2 \times rand_2 \times (gbest_d - x_{id}^{t}) \tag{3.2}$$

where w is 0.9 at the first iteration and linearly decreases to 0.4 at the final iteration [84].

3.2.3 Constriction factor PSO algorithm (CPSO)

CPSO has been proposed by Clerc [31, 32, 86] so as to ensure convergence of the PSO algorithm. The updated velocity in the CPSO can be expressed by:

$$v_{id}^{t+1} = k \times [v_{id}^{t} + c_1 \times rand_1 \times (pbest_{id} - x_{id}^{t}) + c_2 \times rand_2 \times (gbest_d - x_{id}^{t})] \tag{3.3}$$

$$k = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \quad \varphi = c_1 + c_2, \quad \varphi > 4 \tag{3.4}$$

where φ is generally set to 4.1, both c_1 and c_2 are set to 2.05 and k is 0.729 as presented in [86].

3.2.4 Original PSO including inertia weight and constriction factor (CBPSO)

In this algorithm, both the inertia weight and constriction factor are incorporated into the Original PSO, as presented in [15, 34, 35, 58]. The modified velocity of each particle can be calculated as follows:

$$v_{id}^{t+1} = k \times [w \cdot v_{id}^{t} + c_1 \times rand_1 \times (pbest_{id} - x_{id}^{t}) + c_2 \times rand_2 \times (gbest_d - x_{id}^{t})]. \quad (3.5)$$

Subsequently, the modified position of each particle can be calculated as shown in the following equations:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (3.6)$$

where

v_{id}^{t} : velocity of particle i at iteration t in d-dimensional space;

$$V_{d,\min} \le v_{id}^{t} \le V_{d,\max}\ ;\ i = 1, 2, \ldots, n,\ \ d = 1, 2, \ldots, m,$$

x_{id}^{t} : current position of particle i at iteration t,

w : inertia weight factor,

t : number of iterations,

n : number of particles in a group,

m : number of members in a particle,

k : constriction factor,

c_1, c_2 : acceleration constant,

$rand_1, rand_2$: uniformly distributed random number between 0 and 1.

3.3 A Real-Valued Natural Mutation

Recently, Zhang and Lu [42] have successfully applied a new real-valued mutation operator to the classical GA and quasi-simplex techniques so as to solve non-linear problem. Their simulation results confirm that applying this mutation operator improves the global search capability compared to the classical GA. In this research, the real-valued mutation operator is therefore adopted and combined with the traditional PSO algorithms instead of the classical GA. The details of the real-valued mutation operator can be briefly given as follows. The real-valued mutation operator, which is inspired by the spirit of mutation in natural genetics, is used to enhance the diversity of a swarm. It converts a big digit (≥ 5) to a small digit (< 5) and vice versa. Under the real-valued scheme, the j-th component in the i-th particle $x_i(j)$ can be represented by a sequence of digits including the decimal point as [42]:

$$x_i(j) = \underbrace{d_{ij}^{w_1} d_{ij}^{w_2} \ldots d_{ij}^{w_p}}_{\text{whole number}} \bullet \underbrace{d_{ij}^{f_1} d_{ij}^{f_2} \ldots d_{ij}^{f_q}}_{\text{fractional number}} \tag{3.7}$$

where $d_{ij}^{w_r}$ and $d_{ij}^{f_s}$ denote the r-th digit and the s-th digit in the "*whole number*" and "*fractional*" parts, respectively, with a number of digits $w_r = 1,2,\ldots, w_p$ and $f_s = 1,2,\ldots, f_q$, defined by constants p and q that are specified for a given $x_i(j)$. During the mutation operation, firstly, a mask denoted by:

$$mk_{ij} = b_{ij}^{w_1} b_{ij}^{w_2} \ldots b_{ij}^{w_p} b_{ij}^{f_1} b_{ij}^{f_2} \ldots b_{ij}^{f_q} \tag{3.8}$$

will be generated randomly for each $x_i(j)$, $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$, where $b_{ij}^{w_r}$ and $b_{ij}^{f_s}$, $r = 1,2,\ldots,p$ and $s = 1,2,\ldots,q$, are binary values. If $b_{ij}^{w_r} = 1$ or $b_{ij}^{f_s} = 1$, the corresponding $d_{ij}^{w_r}$ or $d_{ij}^{f_s}$ will mutate by:

$$\overline{d_{ij}^{\,r}} = \begin{cases} 9 - d_{ij}^{r}, & \text{if} \quad b_{ij}^{\,r} = 1, \\ d_{ij}^{r}, & \text{otherwise.} \end{cases} \qquad r = w_1, w_2, \ldots, w_p, f_1, f_2, \ldots f_q \tag{3.9}$$

3.4 Hybrid PSO Algorithms with the Real-Valued Natural Mutation Operator

3.4.1 Combination of PSO algorithms and the Mutation Operation

The new hybrid PSO algorithms incorporate the natural mutation operator (RVM) into the traditional PSO algorithms in such a way that the resulting algorithms are PSO-dominated and retain the attractive features of PSO while the mutation acts as a fractional complement. Suppose that the swarm flies from the k-th to the (k+1)-th stop, as shown in Figure 3.1. At the (k+1)-th stop, firstly, the particles are updated by a PSO procedure using (3.2), (3.3), or (3.5) to be x_i^{k+1}, and then a number of particles undertake mutation operation to further change their positions. The percentage of particles undertaking mutation is usually small. In Figure 3.1, we assume that only one particle, say the i-th particle, undertakes the mutation operation. After mutation, the i-th particle's position will become x_{i-mut}^{k+1}, i.e. $x_i^{k+1} = x_{i-mut}^{k+1}$ and the PSO procedure resumes and particles fly to the next stop.

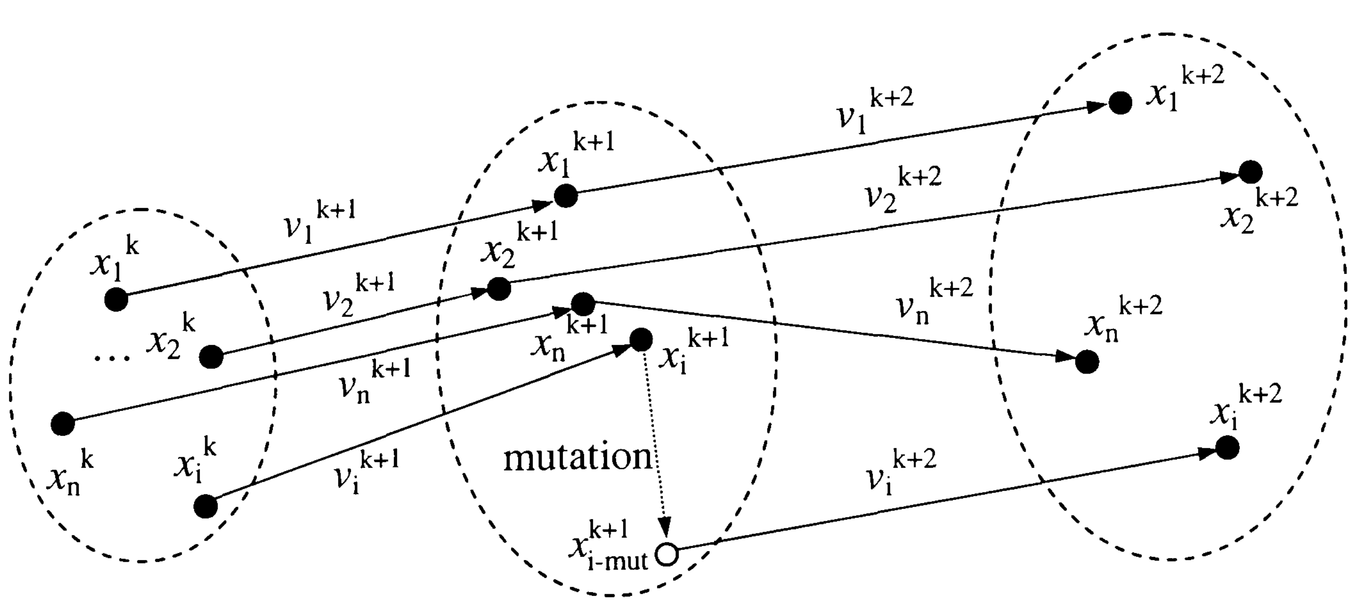


Figure 3.1 Concept of combination of PSO algorithm and mutation operation

3.4.2 Hybrid Algorithms

By incorporating the mutation operator (RVM) into the PSO algorithms, we form a set of hybrid algorithms (PSO-RVM), i.e. the basic PSO with mutation (BPSO-RVM), the advanced PSO with mutation (CPSO-RVM) and the original PSO with both inertia weight and constriction factor with mutation (CBPSO-RVM). These hybrid algorithms (PSO- RVM) can be described as follows:

Step 1. Initialisation

- Determine the number of particles, *m*;
- Randomly generate *m* feasible particles to be the candidate solutions to the optimisation problem;
- Set the mutation probability, P_m;
- Set the termination criteria;
- Initialise $pbest_i$, i = 1,2,...,m, to be the initial position of the *i*-th particle;
- Initialise *gbest* to be the best position of all particles in the swarm;
- Initialise the velocities of each particle randomly within the limit [- v_{max}, v_{max}];
- Set the values of parameters c_1 and c_2;
- Set the starting and ending values for the weight factor, *w*.

Step 2. Update the velocity for each particle by (3.2), (3.3), or (3.5) and make sure all its components within the limit [- v_{max}, v_{max}].

Step 3. Update the position for each particle using (3.6).

Step 4. A few particles perform the mutation operations;

- Randomly select $[P_m \cdot m]$ particles to perform mutation. For each selected particle, randomly choose the components that will mutate. For each chosen component, generate a mask and perform mutation operation on the digits whose corresponding mask values are "1" using (3.9);

- Update the positions of mutated particles to be the positions after mutation.

Step 5. Update the $pbest_i$ $i = 1,2,\ldots,m$ and *gbest*, respectively.

Step 6. Check the termination criteria. If the criteria are met, go to Step 7, otherwise, go to Step 2.

Step 7. Output *gbest* as the solution to the problem.

3.5 Simulation Results

3.5.1 PSO with Gaussian Mutation (PSO-GM)

In order to validate the effectiveness of the proposed hybrid PSO algorithms, their simulation results will be compared with the outcomes obtained from the traditional PSO algorithms. In addition, the PSO with Gaussian Mutation (PSO-GM) that has been introduced by [34], is re-implemented and compared to the proposed algorithms. The PSO-GM utilises a mutation operator, called *Gaussian Mutation* that is generally applied to Genetic Algorithm (GA). It is aimed at coping with the loss of diversity in global search by incorporating Gaussian Mutation into the traditional PSO as presented in [19, 23, 34]. Applying Gaussian Mutation improves the PSO searching ability by mutating some selected particles. The procedures of the implementation in this section are rather similar to the section 3.4.2 except the mutation part. Since the constriction factor PSO algorithm (CPSO) [86] has shown superiority over BPSO in case of solution quality, it is generally used in many areas of problem [23, 87, 88]. Therefore the modified velocity of particle can be calculated by (3.3). Concerning the mutation section, the Real-Valued Natural Mutation (RVM) changes into the Gaussian Mutation (GM). The implementation of Gaussian Mutation can therefore be expressed in details as follows:

Step 1: Determine the mutation probability (P_m) by:

$$P_m = \frac{R_m}{m} \tag{3.10}$$

where : m : the number of particles,

R_m : mutation rate. As reported in [34], R_m is set to 1 at the first iteration and linearly decreases to 0 at the final iteration.

Step 2: Generate a uniformly distributed random number ($rand_i$) between 0 and 1 for each particle.

Step3: Compare each generated random number ($rand_i$) with P_m. If $P_m > rand_i$, then mutate the particle by following equation [34].

$$x^t_{i,mutate} = x^t_i \times (1 + gaussian(\sigma)) \tag{3.11}$$

where $x^t_{i,mutate}$: mutated position of particle i at iteration t,

x^t_i : current position of particle i at iteration t,

$gaussian(\sigma)$: a random number drawn from a Gaussian distribution, $\sigma = 0.1 \times \textit{The length of search space}$.

3.5.2 Validation using a suite of five benchmark functions

The five test functions are chosen from the benchmark function class, which appears to be the most difficult class of problems for many optimisation algorithms, and tabulated in Table 3.1. As mentioned above, these functions are the Sphere Function (f_1), the Quadric Function (f_2), the Griewank Function (f_3), the Rastrigrin Function (f_4), and the Ackley Function (f_5). The number of local minima for each function increase exponentially with the problem dimension. All five functions are also plotted in three dimensions as illustrate in Table 3.2. The hybrid PSO algorithms as well as the PSO algorithms are implemented in Matlab. The parameters used in the implementation are listed in Table 3.3.

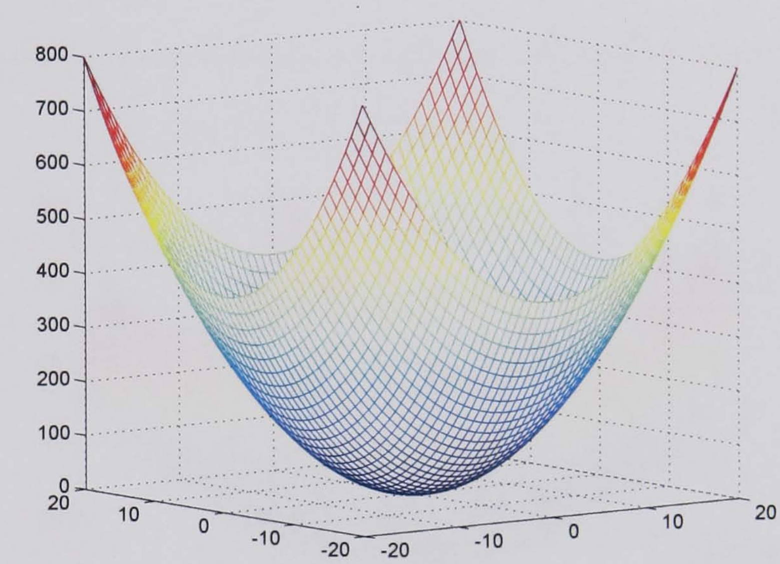
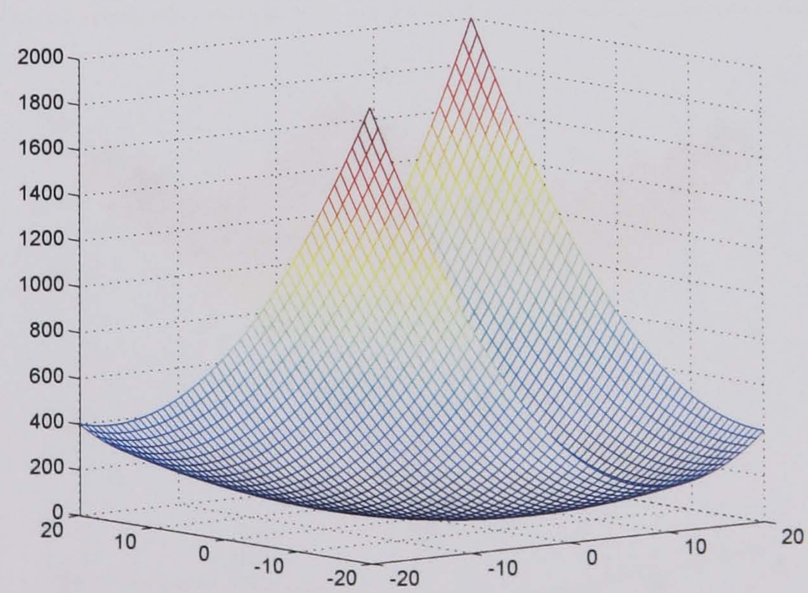
Table 3.1 Optimisation test functions

Function Name	Expression and Conditions
Sphere Function	$f_1(x)=\sum_{i=1}^{n} x_i^2$ n = 30 $x_i \in [-100,100]$, $v_{max}=100$ $\min(f_1)=f_1(0,\ldots,0)=0$
Quadric Function	$f_2(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$ n = 30 $x_i \in [-100,100]$, $v_{max}=100$ $\min(f_2)=f_2(0,\ldots,0)=0$
Griewank Function	$f_3=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$ n = 30 $x_i \in [-600,600]$, $v_{max}=600$ $\min(f_3)=f_3(0,\ldots,0)=0$
Rastrigrin Function	$f_4(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$ n = 30 $x_i \in [-5.12,5.12]$, $v_{max}=5.12$ $\min(f_4)=f_4(0,\ldots,0)=0$
Ackley Function	$f_5(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)$ $+20+e$

	n = 30 $x_i \in [-32,32]$, $v_{max} = 32$ $\min(f_5) = f_5(0,\ldots,0) = 0$

Note: n, x_i, v_{max} and $\min(f_i)$ represent the dimension of a function, variables, maximum particle velocity, and the known global minimum of a function i, respectively.

Table 3.2 The plots of various test functions in three dimensions

Function Name	Shape
Sphere Function	
Quadric Function	

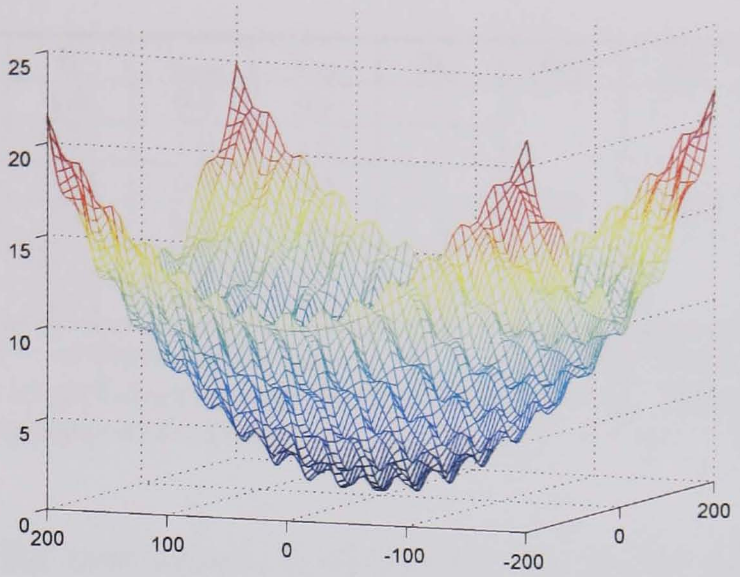
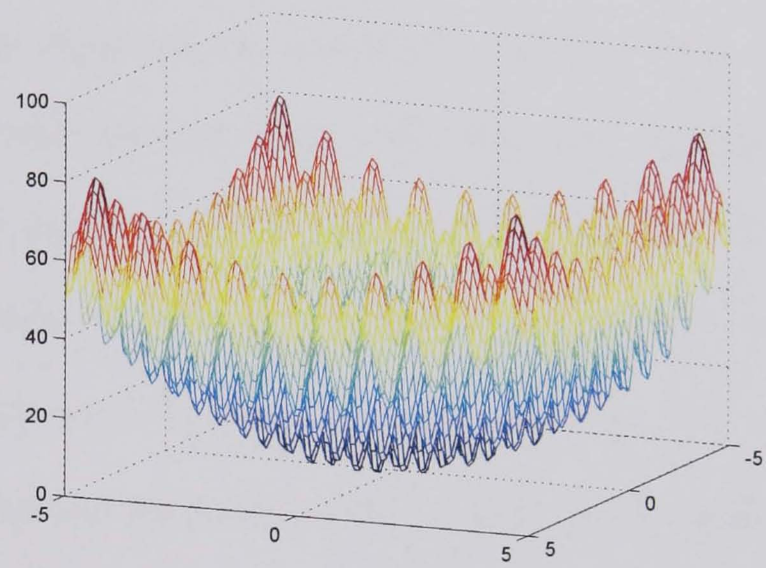
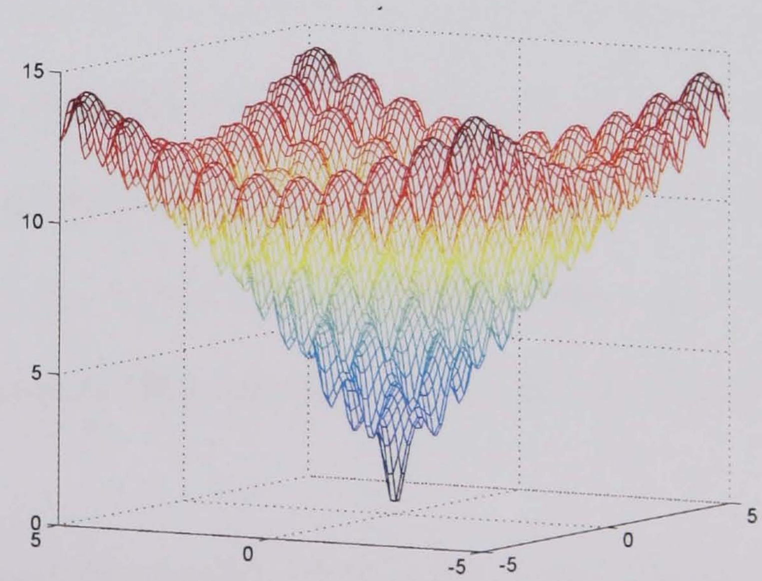
Griewank Function	
Rastrigrin Function	
Ackley Function	

Table 3.3 Parameters used in the implementation.

Methods	c_1/c_2	φ	K	w_{max}	w_{min}	N_m	Pop	Iter	Dim
BPSO	2.0	--	1.0	0.9	0.4	--	20	2000	30
CPSO	2.05	4.1	0.73	1.0	1.0	--			
CBPSO	2.05	4.1	0.73	0.9	0.4	--			
BPSO-RVM	2.0	--	1.0	0.9	0.4	1			
CPSO-RVM	2.05	4.1	0.73	1.0	1.0	1			
CBPSO-RVM	2.05	4.1	0.73	0.9	0.4	1			

Note: c_1,c_2 - acceleration constants, φ - summation of c_1 and c_2 , K - constriction factor, $w_{max,min}$ - max/min inertia weight, N_m - number of particles that participate in mutation, Pop - population size, Iter - total number of iterations, Dim - dimension of a function.

In order to reduce the effect of the randomness of the results, we run the program a number of times (N_{run}). In our experiments, N_{run} = 100. For each function, we record the mean best function values using all the algorithms, median, range, and the standard derivations of these best function values over N_{run} runs. Tables 3.4-3.8 tabulate the simulation results over N_{run} runs obtained from the traditional PSO algorithms (BPSO, CPSO, CBPSO), the hybrid PSO algorithms (BPSO-RVM, CPSO-RVM, CBPSO-RVM), and PSO with Gaussian Mutation (PSO-GM) for each test function. Figures 3.2 -3.6 plot the average convergence curves of these algorithms for each test function. For the sake of illustration, these tables and plots are paired up in terms of test functions. In the rest of this section, the hybrid PSO algorithms, the traditional PSO algorithms, and PSO with Gaussian mutation are compared from the following four aspects for each test function: (1) the hybrid PSO algorithms versus the traditional PSO algorithms; (2) the hybrid PSO algorithms versus the PSO with Gaussian mutation; (3) the traditional PSO algorithms alone; and (4) the hybrid PSO algorithm alone.

For the sphere function (f_1), which is also a unimodal function and the easiest function amongst the test functions, the following points can be observed from Table 3.4 and Figure 3.2 under the test condition:

- The hybrid PSO algorithms improve the searching ability of the conventional PSO algorithms, except for the BPSO-RVM that gives slightly worse solutions than its counterpart (BPSO).
- PSO-GM performs relatively poorly compared with its counterpart (CPSO), CPSO-RVM, and CBPSO-RVM, respectively.
- CPSO performs very well in this case while, BPSO and CBPSO get into the stagnation states after the first 200 iterations of search processes.
- CBPSO-RVM gives a good convergence rate for all range of search process and yields an outstanding result. CBPSO-RVM is therefore superior to other algorithms in this test function.

Table 3.4 Comparison of simulation results for Sphere function (f_1)

Method	Mean best	Median	Range	Std Dev
BPSO	6.31E+03	6.36E+03	[2.43E+03 : 9.67E+03]	1.74E+03
BPSO-RVM	8.61E+03	8.51E+03	[3.74E+03 : 1.72E+04]	2.42E+03
CPSO	9.59E-16	7.12E-20	[4.23E-26 : 7.73E-14]	7.78E-15
CPSO-RVM	1.19E-20	4.14E-21	[3.37E-24 : 2.87E-19]	3.29E-20
CBPSO	1.25E+01	2.14E+00	[1.13E-03 : 2.20E+02]	3.11E+01
CBPSO-RVM	1.14E-43	5.66E-45	[8.55E-47 : 2.02E-42]	3.22E-43
PSO-GM	3.06E-10	8.51E-11	[5.70E-13 : 2.53E-09]	5.31E-10

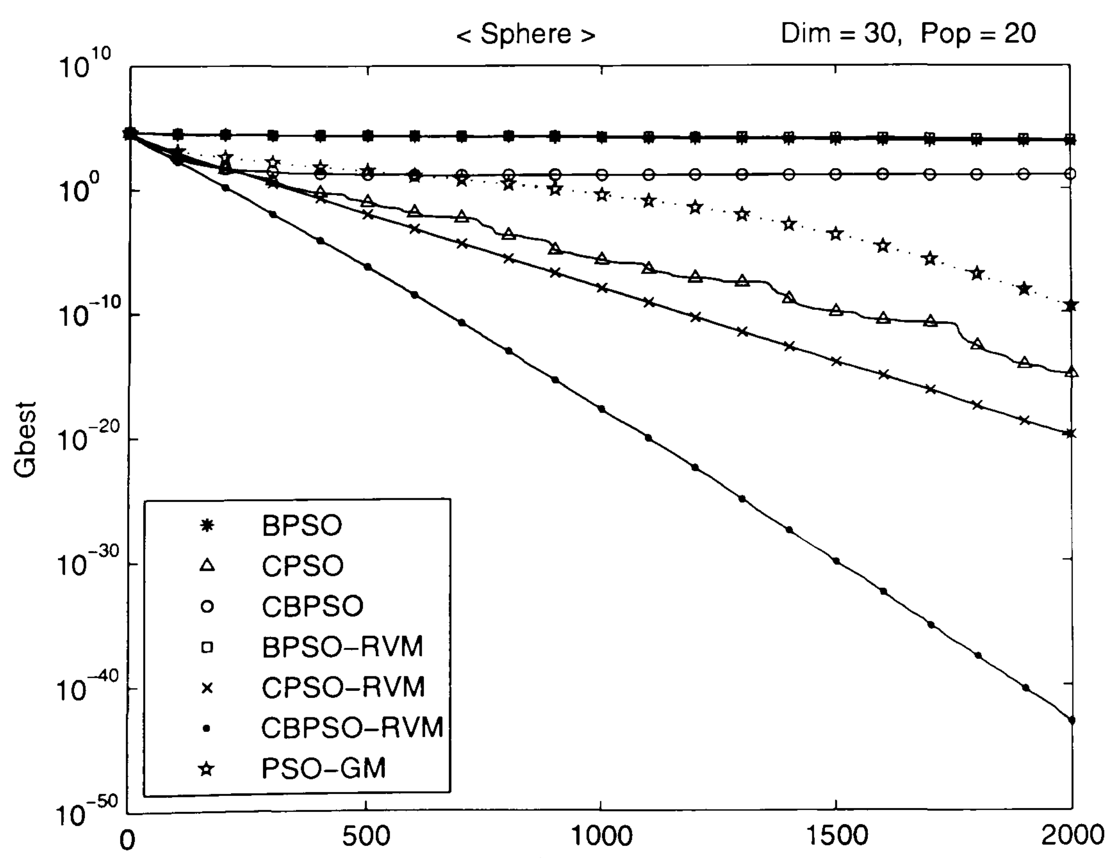


Figure 3.2 Convergence curves for Sphere function (f_1)

For the quadric function (f_2), which is a unimodal function as well, the following points can be observed from Table 3.5 and Figure 3.3 under the test condition:

- CBPSO-RVM and CPSO-RVM still perform better than the conventional PSO algorithms, whereas BPSO-RVM performs slightly worse than BPSO.
- Like Sphere function, PSO-GM performs poorly compared with CPSO, CPSO-RVM, and CBPSO-RVM.
- The performance of CPSO here is rather similar to the case of Sphere function in that it is superior to all PSO algorithms apart from CBPSO-RVM.
- Again, CBPSO-RVM is superior to other algorithms in this test case.

Table 3.5 Comparison of simulation results for Quadric function (f_2)

Method	Mean best	Median	Range	Std Dev
BPSO	2.44E+04	2.45E+04	[1.23E+04 - 3.59E+04]	5.05E+03
BPSO-RVM	3.11E+04	3.17E+04	[1.11E+04 - 4.51E+04]	6.19E+03
CPSO	2.95E+00	9.78E-01	[1.09E-01 - 8.06E+01]	8.76E+00
CPSO-RVM	1.53E+02	1.06E+02	[1.75E+01 - 9.24E+02]	1.44E+02
CBPSO	9.09E+02	6.50E+02	[9.06E+00 - 5.47E+03]	8.36E+02
CBPSO-RVM	3.04E-01	1.71E-01	[3.71E-03 - 3.99E+00]	4.77E-01
PSO-GM	1.85E+02	4.98E+01	[7.39E+00 - 6.69E+03]	8.29E+02

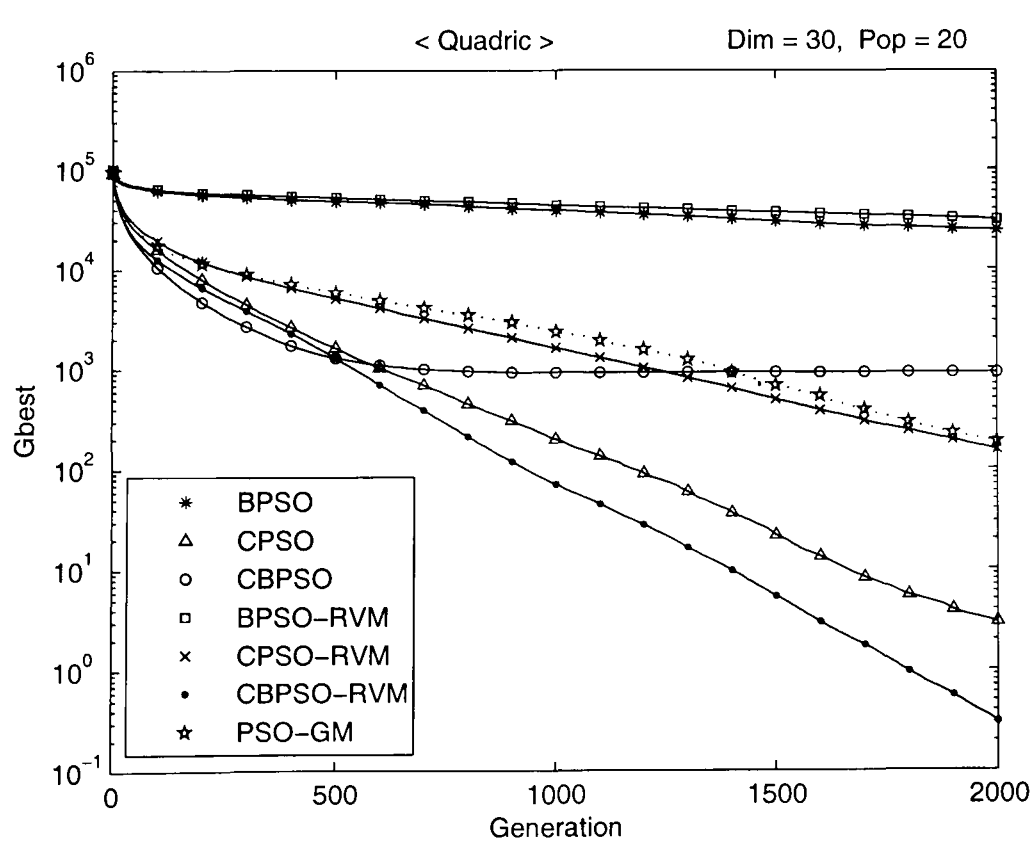


Figure 3.3 Convergence curves for Quadric function (f_2)

For the Griewank function (f_3), which is a multimodal function, the following points can be observed from Table 3.6 and Figure 3.4 under the test condition:

- In addition to BPSO-RVM, the hybrid PSO algorithms outperform the traditional PSO algorithms.
- In this case, PSO-GM performs better than CPSO; however, it still performs disappointingly compared to CBPSO-RVM and CPSO-RVM.
- CBPSO and CPSO perform very well in the early stage of search processes, but convergence rates deteriorate dramatically around 300 iterations and get into the stagnation states.
- CBPSO-RVM and CPSO-RVM converge quickly in the early stage of search processes, then their convergence rates significantly decrease after 500 iterations and approach to the stagnation states. BPSO-RVM performs in a similar way to BPSO, but it shows slightly worse than BPSO.
- CBPSO-RVM is superior to other algorithms.

Table 3.6 Comparison of simulation results for Griewank function (f_3)

Method	Mean best	Median	Range	Std Dev
BPSO	5.69E+01	5.69E+01	[1.82E+01 - 9.50E+01]	1.43E+01
BPSO-RVM	6.64E+01	6.37E+01	[2.71E+01 - 1.55E+02]	2.05E+01
CPSO	8.34E-02	3.80E-02	[0.00E+00 - 8.10E-01]	1.54E-01
CPSO-RVM	9.84E-03	3.70E-03	[0.00E+00 - 5.16E-02]	1.26E-02
CBPSO	8.63E-01	7.28E-01	[3.55E-02 - 6.83E+00]	8.63E-01
CBPSO-RVM	9.59E-03	7.40E-03	[0.00E+00 - 6.36E-02]	1.26E-02
PSO-GM	1.50E-02	9.86E-03	[1.04E-09 - 1.00E-01]	1.93E-02

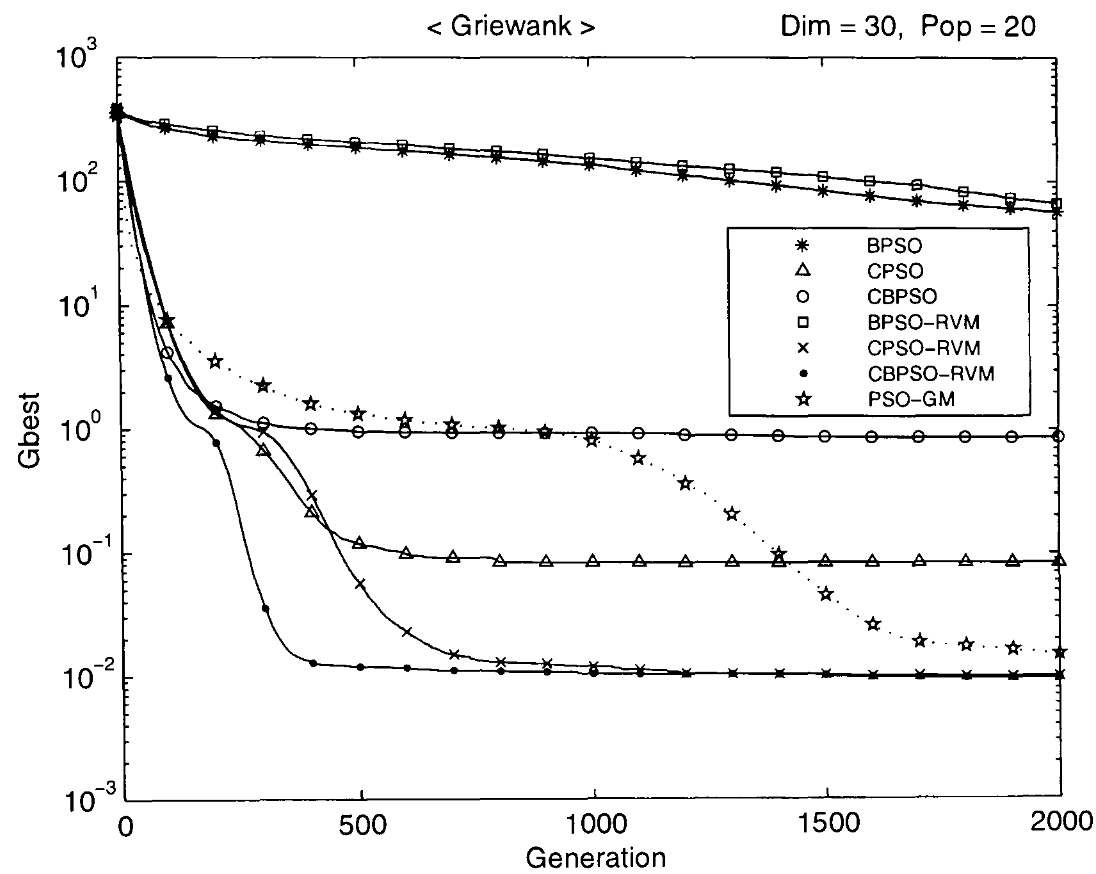


Figure 3.4 Convergence curves for Griewank function (f_3)

For the Rastrigrin function (f_4), which is the hardest function to find the global optimum amongst the test functions, the following points can be observed from Table 3.7 and Figure 3.5 under the test condition:

- The hybrid PSO algorithms give significantly better mean function values than their PSO counterparts with higher convergence rates except BPSO-RVM that performs better than BPSO up to 1100 iterations, then BPSO-RVM slows down the convergence rate.
- Besides CBPSO-RVM and CPSO-RVM, PSO-GM is also superior to other PSO algorithms.
- CPSO and CBPSO perform quite well in early stage of the search processes, but their performances deteriorate quickly around 300 iterations, while BPSO has a lower convergence rate.
- CBPSO-RVM and CPSO-RVM perform very well throughout the search process; on the other hand, BPSO-RVM performs poorly.

- CBPSO-RVM shows the superiority over other algorithms.

Table 3.7 Comparison of simulation results for Rastrigrin function (f_4)

Method	Mean best	Median	Range	Std Dev
BPSO	2.85E+02	2.83E+02	[2.03E+02 - 3.57E+02]	3.61E+01
BPSO-RVM	3.73E+02	3.74E+02	[3.17E+02 - 4.29E+02]	2.30E+01
CPSO	7.14E+01	7.06E+01	[3.78E+01 - 1.18E+02]	1.75E+01
CPSO-RVM	3.62E+01	3.09E+01	[1.09E-09 - 1.19E+02]	2.96E+01
CBPSO	8.35E+01	8.54E+01	[4.40E+01 - 1.37E+02]	2.23E+01
CBPSO-RVM	1.30E+01	1.53E+00	[0.00E+00 - 9.76E+01]	2.08E+01
PSO-GM	5.39E+01	5.28E+01	[8.44E-10 - 1.35E+02]	2.64E+01

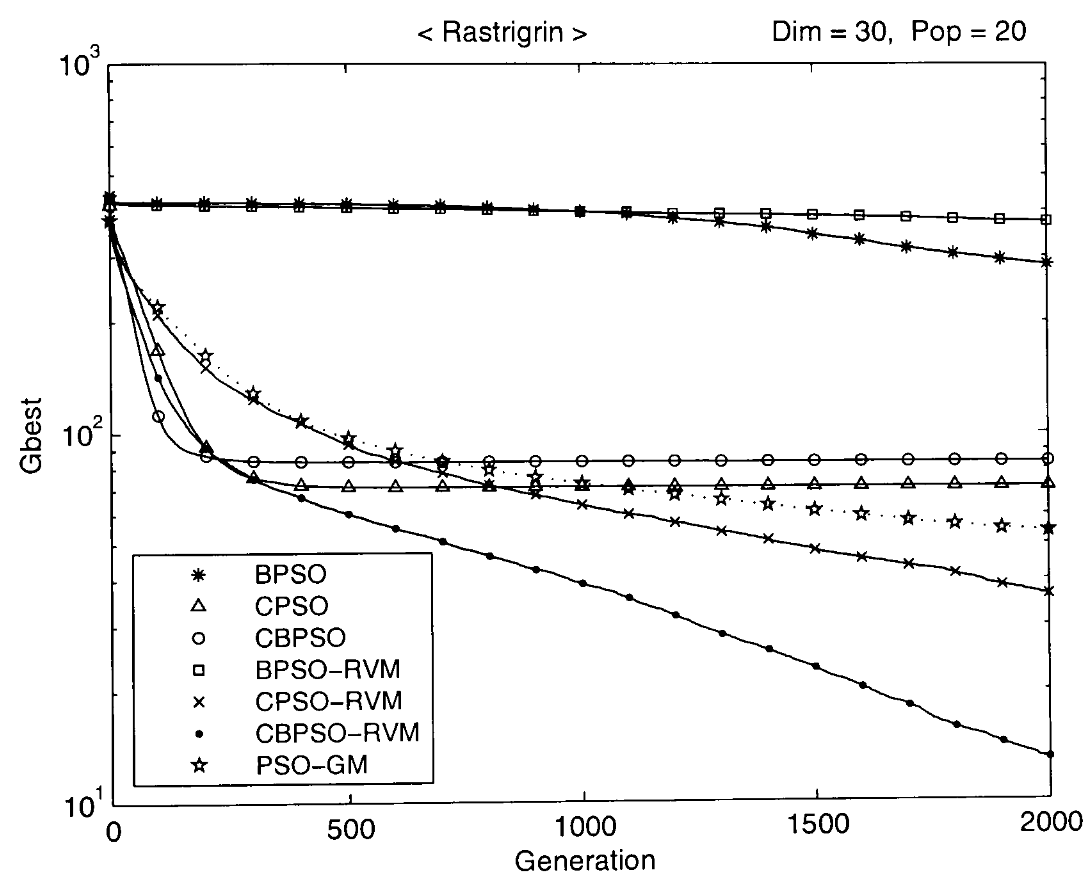


Figure 3.5 Convergence curves for Rastrigrin function (f_4)

For the Ackley function (f_5), which is another multimodal function, the following points can be observed from Table 3.8 and Figure 3.6 under the test condition:

- CBPSO-RVM and CPSO-RVM perform as well as the corresponding PSO algorithms in the very early stage of search processes (around the first 100 iterations) and get even better when the corresponding PSO algorithms start to perform poorly and keep the higher convergence rates to approach the optimal value.

- There is a similarity to other test cases in which PSO-GM still performs worse than CPSO-RVM and CBPSO-RVM. However, it outperforms CPSO in this case.
- The conventional PSO algorithms suffer from the convergence problem because they get into stagnation states from the very early stages of search processes.
- Once again, CBPSO-RVM shows the superiority in searching ability compared to others.

Table 3.8 Comparison of simulation results for Ackley function (f_5)

Method	Mean best	Median	Range	Std Dev
BPSO	1.39E+01	1.41E+01	[1.09E+01 - 1.62E+01]	9.73E-01
BPSO-RVM	1.48E+01	1.49E+01	[1.19E+01 - 1.67E+01]	9.32E-01
CPSO	3.43E+00	3.13E+00	[1.16E+00 - 7.60E+00]	1.45E+00
CPSO-RVM	3.65E-11	1.73E-11	[1.60E-12 - 3.24E-10]	4.88E-11
CBPSO	6.03E+00	5.88E+00	[2.33E+00 - 1.18E+01]	1.85E+00
CBPSO-RVM	1.40E-14	1.51E-14	[7.99E-15 - 1.51E-14]	2.45E-15
PSO-GM	1.46E-07	2.48E-08	[3.01E-09 - 8.10E-06]	8.46E-07

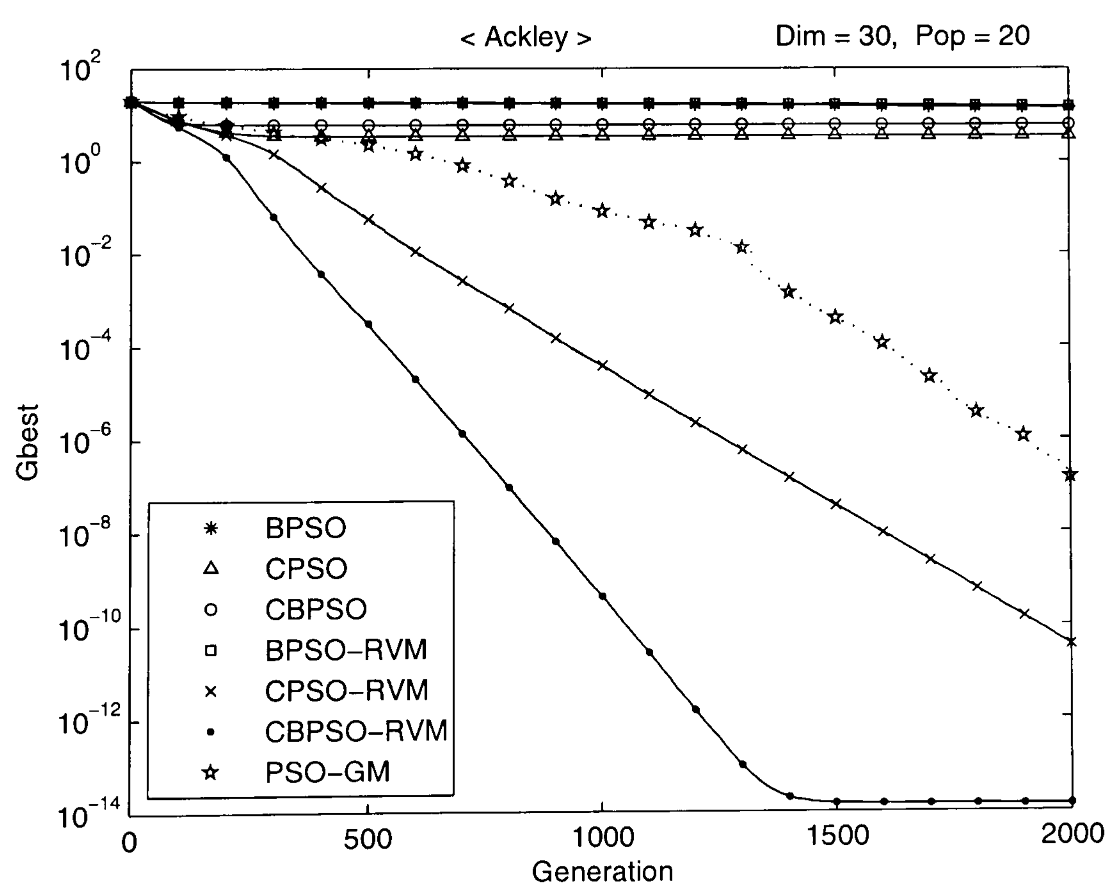


Figure 3.6 Convergence curves for Ackley function (f_5)

3.5.3 Discussion

In this section, the discussions will be separated according to modality into two parts (1) comparison of results for algorithms associated with unimodal functions, and (2) similar comparison for multimodal functions. Concerning the group of unimodal functions, the discussion can be expressed in details as follows:

- It seems that CPSO performs better than other algorithms, but its performance is worse than CBPSO-RVM for both Sphere and Quadric functions. However, between BPSO-RVM and BPSO, the effectiveness of mutation operator is slightly negative. This is reasonable because the incorporation of mutation operation increases the diversity of particles in the swarm, but higher diversity does not lead to higher global search capability for unimodal functions.
- As reported in [23], PSO with various mutation operators perform worse than PSO without mutation operators; however, its performance is superior to PSO without mutation in multimodal functions. Similarly, our simulation results show that the performance of PSO-GM is also unsatisfactory compared with its counterpart (CPSO).
- Although PSO-GM and PSO with various mutation operators as presented in [23] perform unsuccessfully in unimodal functions compared to CPSO, the proposed CBPSO-RVM in this research outperforms not only CPSO, but also other hybrid PSO algorithms under the test conditions.

For the group of multimodal functions, the discussion can be expressed in details as follows:

- The hybrid PSO algorithms give considerably improvement in the searching ability of PSO algorithm except for BPSO-RVM. This may be due to the fact that the incorporation of mutation operation increases the diversity of particles in the swarm and a higher diversity of particles slow down the convergence.

- PSO-GM performs well compared to its counterpart; in contrast, its performance is worse than CBPSO-RVM and CPSO-RVM as well.
- From above comparisons, it can be seen that the incorporation of the natural mutation operator into the PSO algorithms is significantly beneficial to multimodal functions. In addition, CBPSO-RVM outperforms all the other algorithms.

3.6 Summary

In this chapter, a method of combining the PSO algorithms with a Real-Valued Natural Mutation (RVM) operator is proposed. To illustrate the effectiveness of this method, both the hybrid PSO algorithms and the traditional PSO algorithms have been implemented and applied to optimise five benchmark non-linear functions. Additionally, a hybrid of PSO and Gaussian Mutation (PSO-GM) is re-implemented for investigation and validation. The results from optimising the benchmark functions show that the incorporation of mutation operator into PSO algorithms significantly enhances the searching diversity of the swarm. Moreover, amongst the hybrid PSO algorithms, CBPSO-RVM is indeed better than others in terms of the convergence characteristic and the solution quality.

Chapter 4: Application of PSO in Economic Dispatch

4.1 Introduction

This chapter proposes the application of Particle Swarm Optimisation (PSO) to the Economic Dispatch (ED) problem, which occurs in the operational planning of power systems. To solve the ED problem, the traditional PSO algorithms and the hybrid PSO algorithms are adopted. The performances of the PSO methods are validated by testing on various types of ED problem, which can be categorised according to the different characteristics of cost function. In this study, four different cost functions are therefore adopted as follows: (1) ED problem with smooth cost function, (2) ED problem with multiple fuels, (3) ED problem with valve-point loading, and (4) ED problem with both multiple fuels and valve-point loading. This chapter is organised as follows: Section 4.2 shows the problem formulation of ED problem. Section 4.3 briefly presents the details of the various ED problems according to cost function characteristics. Section 4.4, the implementation of the PSO algorithms is presented. The simulation results and the discussion are given in section 4.5. Finally, Section 4.6 summarises the chapter.

4.2 Problem Formulation

Economic Dispatch (ED) problem is a sub-problem of the general UC problem. In essence, the ED problem is to determine the optimum scheduling of generation at a particular time that minimises the total production cost while satisfying an equality constraint and inequality constraints, i.e. power balance constraint and operating limits [89]. In general, the mathematical model of the ED problem is as follows [3]:

$$Minimise : TC = \sum_{i=1}^{N} F_i(P_i) \tag{4.1}$$

Subject to:

a) Power balance constraint

$$\sum_{i=1}^{N} P_i = P_D \tag{4.2}$$

b) Operating limit constraints

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{4.3}$$

4.3 A Variety of ED Problems

From the different characteristics of cost function; therefore, they can be categorised as ED problem with smooth cost functions (the standard ED) and ED problem with non-smooth cost functions (the practical ED) as presented in [16, 90-92].

A. ED problem with smooth cost functions

For the sake of simplicity, the cost function of the standard ED problem (smooth cost functions) is generally a single quadratic function. The generator's fuel cost function can be represented by [3]:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i. \tag{4.4}$$

Figure 4.1 shows an example of smooth cost functions.

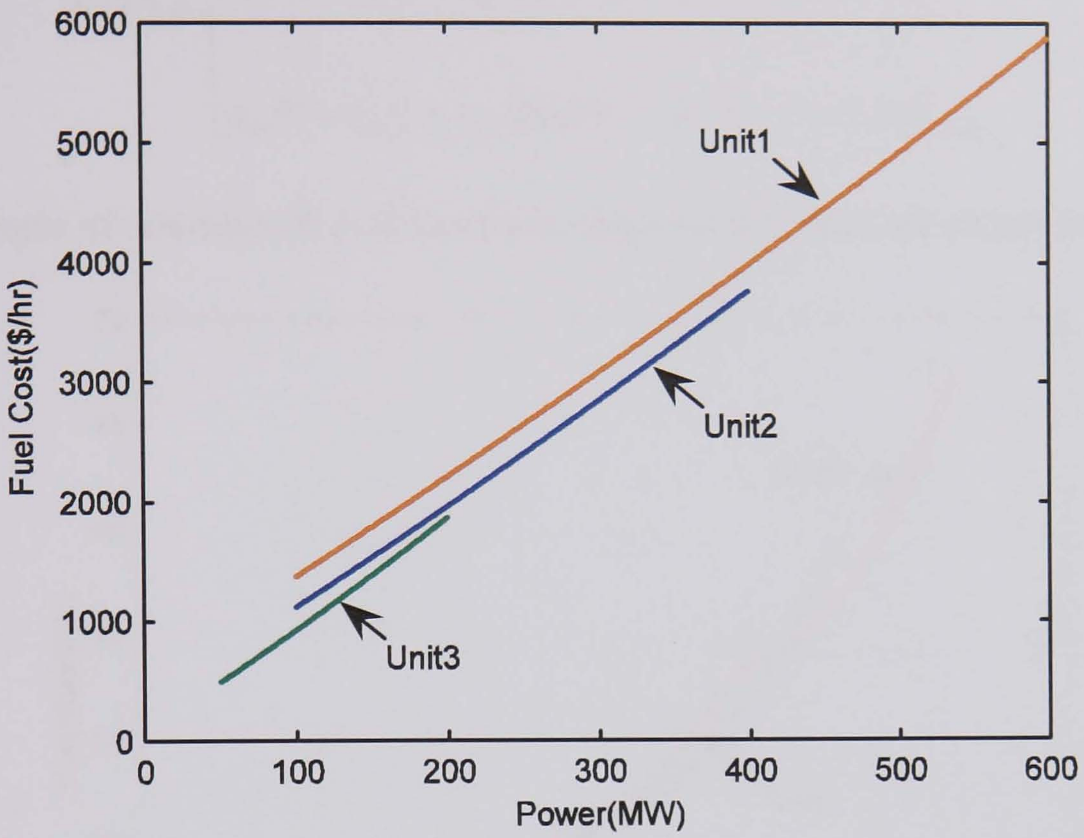


Figure 4.1 An example of input-output curve with smooth cost function

B. ED problem with non-smooth cost functions

In this section, three cases of non-smooth cost functions will be taken into consideration (i.e. non-smooth cost functions with multiple fuels, non-smooth cost functions with valve-point loading, and non-smooth cost functions with multiple fuels and valve-point loading).

B.1 Non-smooth cost functions with multiple fuels

Practically, some generators can be operated with multiple fuels [91, 92]. Therefore, changes of fuel type in ED problem will be responsible for changes in the cost function from a single quadratic function to a piecewise quadratic function [93, 94]. The generator's fuel cost function can be defined as follows [92]:

$$F_i(P_i) = \begin{cases} a_{i1}P_i^2 + b_{i1}P_i + c_{i1}, \text{(fuel 1)}, & \text{if } P_{i,\min} \le P_i \le P_{i,1} \\ a_{i2}P_i^2 + b_{i2}P_i + c_{i2}, \text{(fuel 2)}, & \text{if } P_{i,1} < P_i \le P_{i,2} \\ \vdots & \vdots \\ a_{ik}P_i^2 + b_{ik}P_i + c_{ik}, \text{(fuel } k\text{)}, & \text{if } P_{i,k-1} < P_i \le P_{i,\max}. \end{cases} \quad (4.5)$$

An example of non-smooth cost functions with multiple fuels are shown in Figure 4.2.

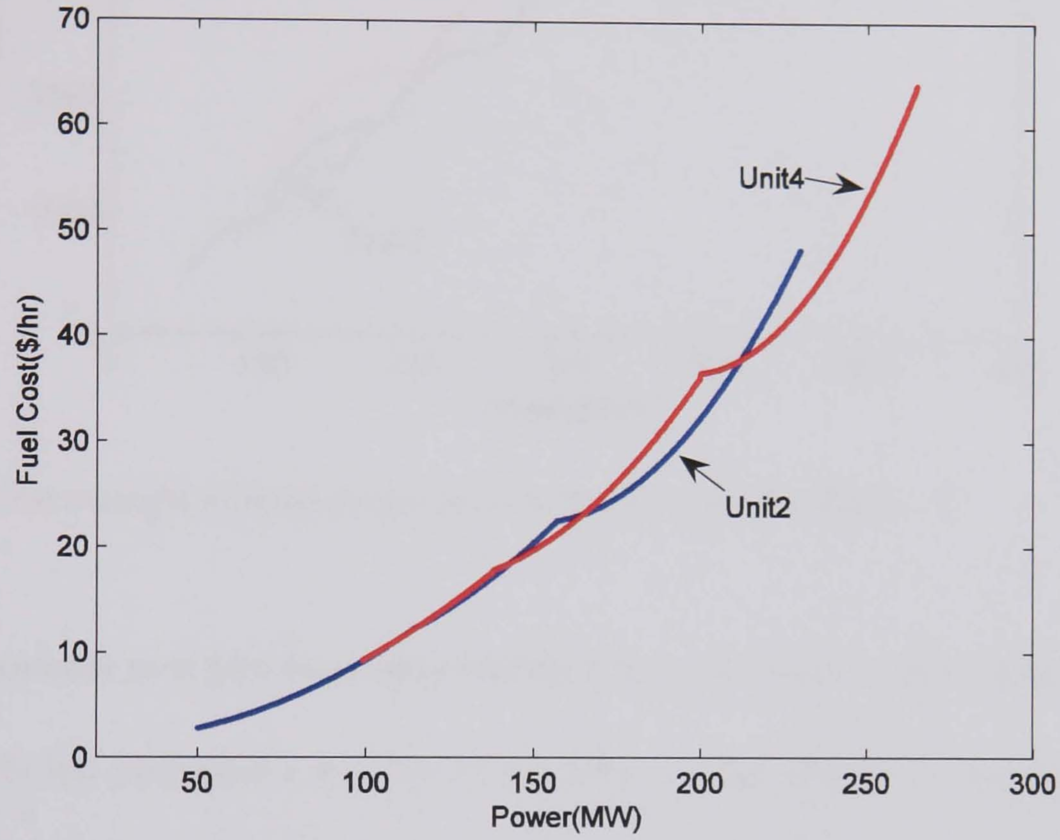


Figure 4.2 An example of input-output curve with multiple fuels.

B.2 Non-smooth cost functions with valve-point loading

In some large generators, their cost functions are also non-linear, due to the effect of valve-point loading [91]. Taking the valve point loading into account will increase multiple local minimum points in the cost function and make the problem more difficult [14]. The fuel cost function with valve-point loading can be expressed as [95]:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i \times \sin(f_i \times (P_{i,\min} - P_i)) \right|. \quad (4.6)$$

Figure 4.3 illustrates an example of non-smooth cost functions with valve-point loading.

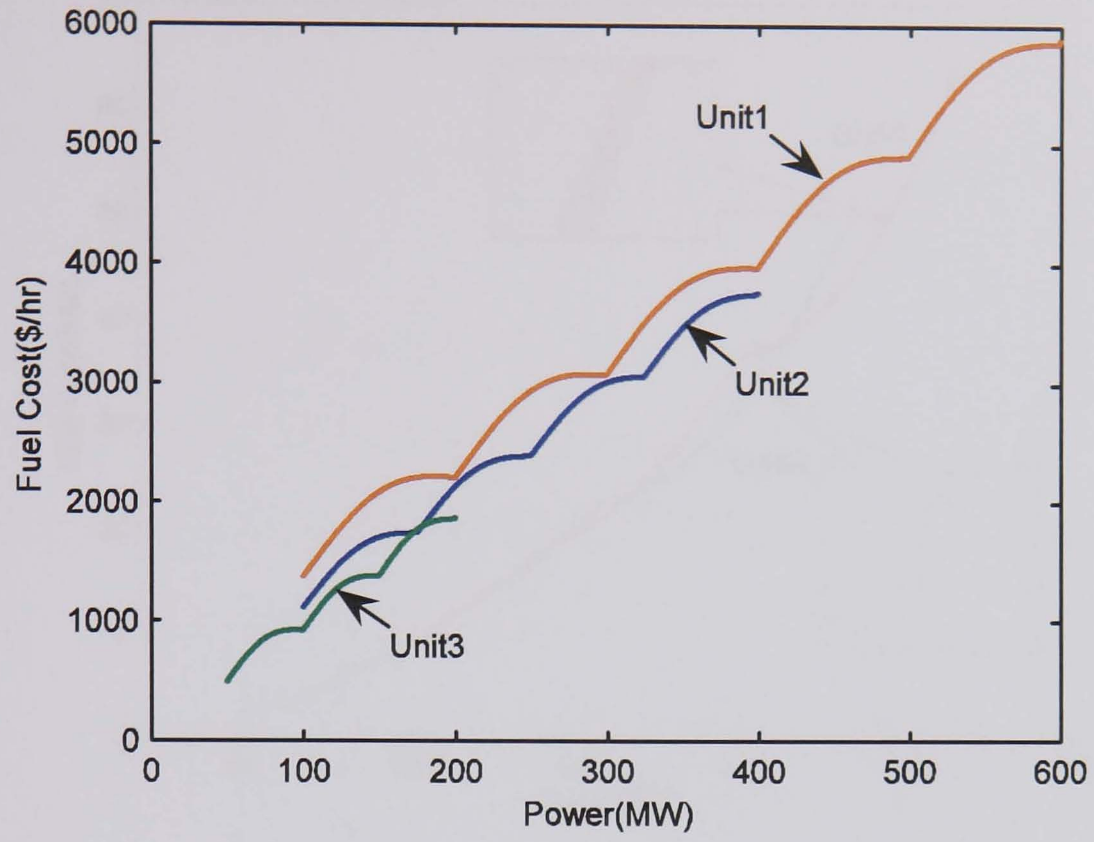


Figure 4.3 An example of input-output curve with valve-point loading

B.3 Non-smooth cost functions with multiple fuels and valve-point loading

Chiang [92] has presented a realistic ED problem considering both multiple fuels and valve-point loading simultaneously in order to make the ED solution more accurate. The fuel cost function with multiple fuels and valve-point loading is represented as follows [92]:

$$F_i(P_i)=\begin{cases} a_{i1}P_i^2+b_{i1}P_i+c_{i1}+\left|e_{i1}\times\sin(f_{i1}\times(P_{i1,\min}-P_i))\right|, & \text{(fuel 1), if } P_{i,\min}\le P_i\le P_{i,1} \\ a_{i2}P_i^2+b_{i2}P_i+c_{i2}+\left|e_{i2}\times\sin(f_{i2}\times(P_{i2,\min}-P_i))\right|, & \text{(fuel 2), if } P_{i,1}< P_i\le P_{i,2} \\ \quad\vdots & \quad\vdots \qquad\qquad \vdots \\ a_{ik}P_i^2+b_{ik}P_i+c_{ik}+\left|e_{ik}\times\sin(f_{ik}\times(P_{ik,\min}-P_i))\right|, & \text{(fuel } k\text{), if } P_{i,k-1}< P_i\le P_{i,\max}. \end{cases} \tag{4.7}$$

An example of non-smooth cost functions with multiple fuels and valve-point loading is presented in Figure 4.4.

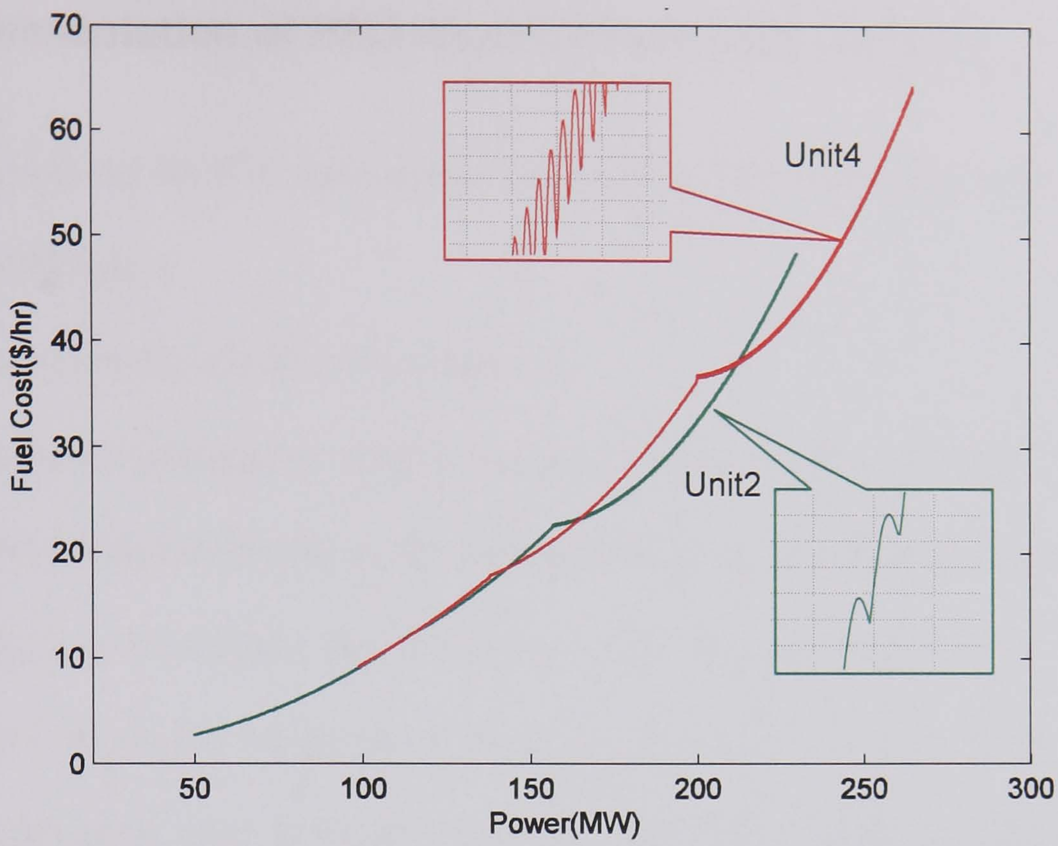


Figure 4.4 An example of input-output curve with multiple fuels and valve-point loading.

List of symbols

TC : total production cost,

$F_i(P_i)$: fuel cost of i^{th} generator; generator's fuel cost can be calculated from (4.4) - (4.7), where a_i, b_i and c_i are coefficients of the fuel cost function , while e_i and f_i are coefficients from the valve-point loading of the i^{th} generator,

P_i : power output of i^{th} generator,

P_D : power demand,

$P_{i,\min}$: minimum power output of i^{th} generator,

$P_{i,\max}$: maximum power out put of i^{th} generator,

N : number of generators,

k : fuel type.

4.4 Implementation of PSO Algorithms in ED problems

The major steps of the PSO approaches in the ED problems are summarised bellow:

Step 1. Initialisation

- Determine the number of particles, *m*.
- Randomly generate *m* feasible particles subject to power balance and operating limit constraints by using the modified heuristic search that is modified from [16] so as to enhance its performance. The strength of the modified version is that it increases a possibility of generating feasible initial solutions with less computation time. It is done by eliminating the variable one by one until the constraint is satisfied, instead of generating a completely random set of new particles when the power balance constraint is violated. The modified search procedures are therefore shown in Figure 4.5.
- Set the mutation probability, P_m.
- Set the termination criteria (the maximum generation).
- Set values for the parameters of PSO, i.e. acceleration constants (c_1,c_2), starting and ending values for the weight factor (w), and constriction factor (k), respectively.
- Set the generation limit, fuel cost coefficients, and power demand.
- Initialise the velocities of particles randomly.
- Calculate the total production cost (TC) of each particle using (4.1).
- Let all particles be $pbest$.
- Set the best position (the least cost) of all particles i.e. the minimum $pbest_i$, i = 1,2,...,m, to be $gbest$.

Step 2. Update the velocity and position of each particle by (3.2), (3.3), (3.5) and

(3.6), respectively. If $x_i^{(k+1)} > P_{i,\max}$, then $x_i^{(k+1)} = P_{i,\max}$ or If $x_i^{(k+1)} < P_{i,\min}$, then $x_i^{(k+1)} = P_{i,\min}$. Otherwise, set $x_i^{(k+1)} = x_i^{(k+1)}$.

Step 3. Mutation operations;

- Randomly select $[P_m \cdot m]$ particles to perform mutation. For each selected particle, randomly choose the components that will mutate. For each chosen component, generate a mask and perform mutation operation based on $b_{ij}^{w_r}$ and $b_{ij}^{f_s}$.
- Update the positions for the mutated particles.

Step 4. Check the position of particles for feasibility. The particles in the swarm obtained in Step 3 are checked and, if necessary, modified subject to power balance constraint and power limit constraints. The procedure is presented in Figure 4.6 in which the calculation is on the random basis. In comparison with the original version [16], this modified version prevents the calculation from repetition process.

Step 5. Calculate *TC* for each particle using (4.1).

Step 6. Update the *pbest*$_i$; $i = 1,2,\ldots,m$ and *gbest*, respectively.

Step 7. Check the termination criteria. If the criterion is met, go to Step 8, otherwise, go to Step 2.

Step 8. Output *gbest* as the solution to the ED problem.

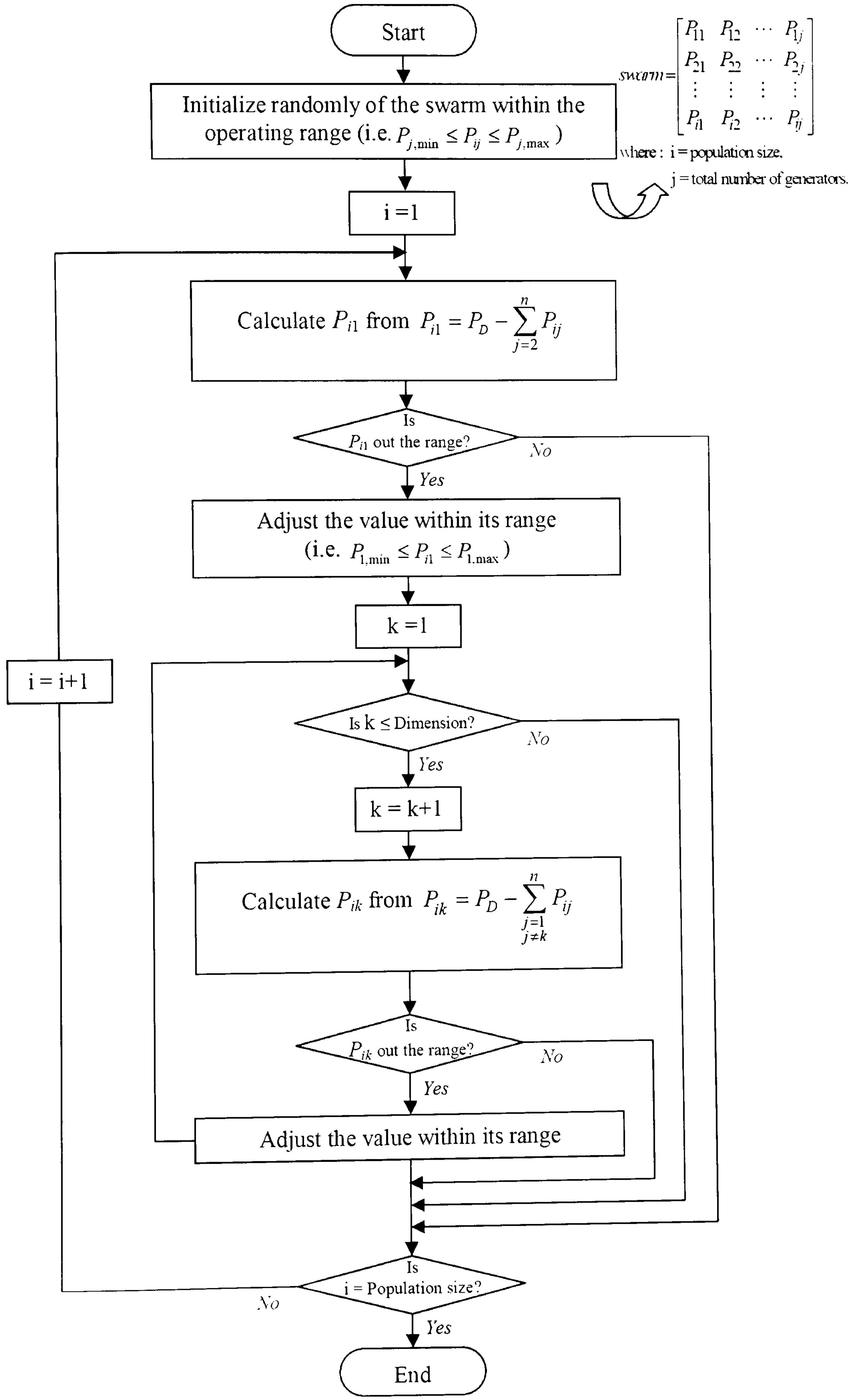


Figure 4.5 Flow chart of the modified heuristic search for initialisation (Step 1)

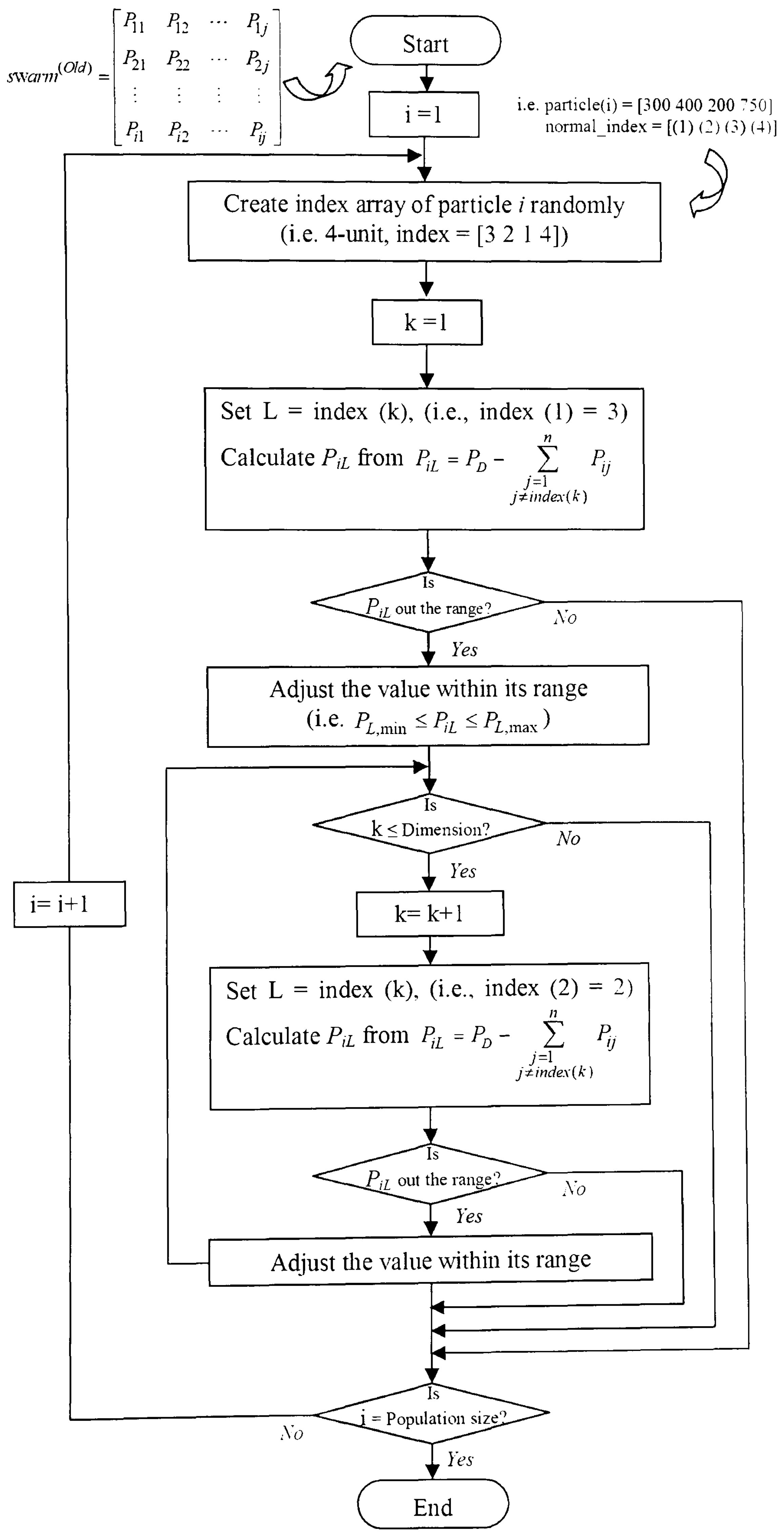


Figure 4.6 Flow chart of the modified heuristic search for particles' modification (Step 4)

4.5 Simulation Results

To investigate the efficiency of the proposed method, five different systems have been considered. The first system is the standard 3-unit system with smooth cost functions as given in [3] and the second is 10-unit system with multiple fuel functions as shown in [93]. The next two systems are 3-unit and 40-unit systems considering valve-point loading as illustrated in [95] and [14], respectively. The last system is 10-unit system that considers multiple fuels and valve-point loading as given in [92]. The data for each system are shown in Appendix A. The simulations are carried out using Matlab and executed on a Pentium IV, 3GHz personal computer with 512 MB RAM. In all cases, each algorithm was run for 100 times (N_{run} = 100) with different initial conditions. In each case, we record the mean cost, the minimum cost and the maximum cost using all the six algorithms and the standard deviation of the mean costs over the 100 runs. The simulation parameters used in the hybrid PSO and the PSO algorithms are listed in Table 4.1.

Table 4.1 Parameters used in the six algorithms for all case studies.

Methods	c_1/c_2	w_{max}	w_{min}	K	N_m	Case Study	n	Pop	Iter
BPSO	2.0	0.9	0.4	1.0	--	Case A (Smooth cost, 3-Unit)	3	10	100
CPSO	2.05	1.0	1.0	0.73	--	Case B (Multiple fuel, 10-Unit)	10	30	300
CBPSO	2.05	0.9	0.4	0.73	--	Case C (Valve-point loading, 3-Unit)	3	20	300
BPSO-RVM	2.0	0.9	0.4	1.0	1	Case D (Valve-point loading, 40-Unit)	40	60	1500
CPSO-RVM	2.05	1.0	1.0	0.73	1	Case E (Multiple fuels & Valve-point loading and , 10-Unit)	10	30	500
CBPSO-RVM	2.05	0.9	0.4	0.73	1				

Note: c_1,c_2 - acceleration constants, $w_{max,min}$ - max/min inertia weight, K - constriction factor, N_m - number of particles that participate in mutation, n - dimension of the problem, Pop - population size, Iter - total number of iterations.

4.5.1 ED problem with smooth cost function

Case A: 3-generator system

In this case, the population size and the maximum number of generations are set to 10 and 100 respectively, and the power demand is set to 850 MW. From the literature, the global solution is $8194.35612 as presented in [3]. Table 4.2 compares the mean cost, the minimum cost, the maximum cost and the standard deviation of the mean costs using the six PSO algorithms with the Modified Hopfield Neural Network (MHNN) [89], the Improved Evolutionary Programming (IEP) [94], the Numerical Method (NM) [3] and the Modified PSO (MPSO) [16]. Figure 4.7 illustrates the average convergence characteristics of the six algorithms. From the simulation of this case, the results show that the traditional PSO algorithms and the hybrid PSO algorithms can achieve the global solution (min. cost). Regarding mean cost and standard deviation, the both CBPSO-RVM and CBPSO methods are more effective than others. In addition, their convergence speeds are also better than others, especially in the first 20 iterations.

Table 4.2 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case A

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
MHNN* [89]	-	8187.00000	-	-
IEP [94]	-	8194.35614	-	-
NM [3]	-	8194.35612	-	-
MPSO [16]	-	8194.35612	-	-
BPSO	8194.35782	8194.35612	8194.40363	0.00540
BPSO-RVM	8194.36187	8194.35612	8194.41179	0.01032
CPSO	8194.35650	8194.35612	8194.36161	0.00077
CPSO-RVM	8194.36036	8194.35612	8194.39865	0.00705
CBPSO	8194.35612	8194.35612	8194.35612	0.00000
CBPSO-RVM	8194.35612	8194.35612	8194.35612	0.00000

* The simulation result illustrates a violation of power balance constraint

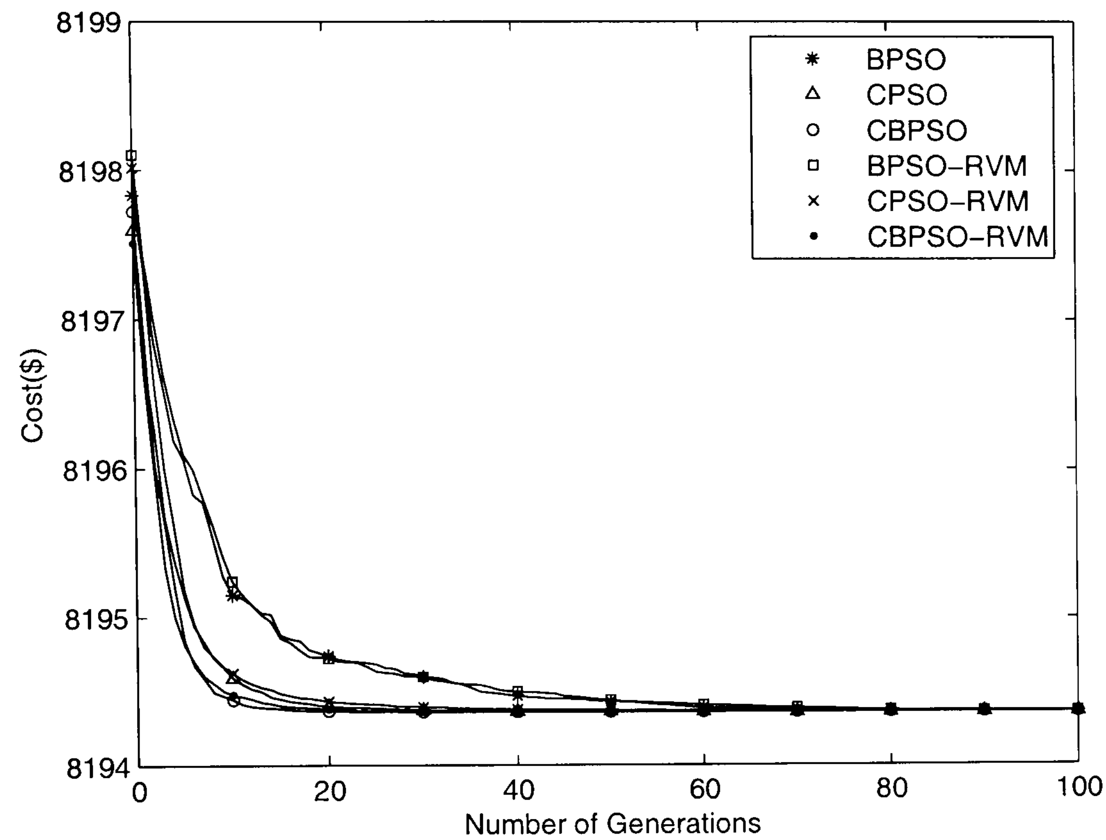


Figure 4.7 Convergence curves of the traditional and hybrid PSO algorithms for Case A.

4.5.2 ED problem with multiple fuels

Case B: 10 - generator system with multiple fuels

In this study, there are four sub-cases where the power demand is varied from 2400MW to 2700MW with a step side of 100. For all sub-cases, the population size and the maximum number of generations are set to 30 and 300, respectively. The parameters used in the six PSO algorithms are listed in Table 4.1 as well.

1) Power demand = 2400MW

Table 4.3 compares the mean cost, the minimum cost, the maximum cost and the standard deviation of the mean costs obtained from the six PSO algorithms with those of the Hierarchical Numerical Method (HM) [93], the Modified Hopfield Neural Network (MHNN) [89], the Improved Evolutionary Programming (IEP) [94]. From the results show that CBPSO-RVM and CBPSO outperform in finding the better solution compared with the other PSO algorithms as well as some selected algorithms.

Table 4.4 shows the frequencies of reaching the final solution over 100 different runs obtained from the six PSO methods. Concerning the number reaching minimum cost in the range of $481.5-$482.5, the CBPSO-RVM and CBPSO reach the final solution in every run whereas others may confront with getting trapped into a local optimum. The comparison of the best solution obtained from some selected methods and CBPSO-RVM is also shown in Table 4.5. In addition, the average convergence curves of the hybrid PSO algorithms as well as the traditional PSO algorithms are shown in Figure 4.8. By comparison, the convergence characteristics of CBPSO-RVM and its counterpart converge rapidly during the first 50 iterations and slowly change until they find the optimum solution in the 120th iteration.

Table 4.3 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case B (Load demand = 2400 MW)

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
HM* [93]	-	488.50	-	-
MHNN* [89]	-	487.87	-	-
IEP [94]	-	481.78	-	-
BPSO	482.1301	481.7740	483.0422	0.2426
BPSO-RVM	482.4550	481.8012	483.7813	0.3954
CPSO	482.0101	481.7951	482.5020	0.1587
CPSO-RVM	482.3552	481.8518	483.6183	0.3107
CBPSO	481.7226	481.7226	481.7226	0.0000
CBPSO-RVM	481.7226	481.7226	481.7227	0.0000

* The simulation result illustrates a violation of power balance constraint

Table 4.4 Frequency of convergence among various methods for Case B (Load demand = 2400 MW)

Method	Cost ($)									
	481.5 - 482.5	482.5 - 483.5	483.5 - 484.5	484.5 - 485.5	485.5 - 486.5	486.5 - 487.5	487.5 - 488.5	488.5 - 489.5	489.5 - 490.5	490.5 - 491.5
BPSO	94	6	-	-	-	-	-	-	-	-
BPSO-RVM	62	35	3	-	-	-	-	-	-	-
CPSO	99	1	-	-	-	-	-	-	-	-
CPSO-RVM	77	22	1	-	-	-	-	-	-	-
CBPSO	100	-	-	-	-	-	-	-	-	-
CBPSO-RVM	100	-	-	-	-	-	-	-	-	-

Table 4.5 The best simulation results obtained from various methods for Case B (Load demand = 2400 MW)

Unit	HM*[93]		MHNN*[89]		IEP[94]		CBPSO-RVM	
	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)
1	1	193.2	1	192.7	1	190.9309	1	189.7494
2	1	204.1	1	203.8	1	202.2978	1	202.3387
3	1	259.1	1	259.1	1	253.8909	1	253.8792
4	3	234.3	2	195.1	3	233.9410	3	233.0336
5	1	249.0	1	248.7	1	243.7515	1	241.8058
6	1	195.5	3	234.2	3	234.9799	3	233.0507
7	1	260.1	1	260.3	1	253.2107	1	253.3212
8	3	234.3	3	234.2	3	232.8043	3	233.0367
9	1	325.3	1	324.7	1	317.1512	1	320.3739
10	1	246.3	1	246.8	1	237.0417	1	239.4107
Total Power (MW)		2401.2	2399.8		2400.00		2400.00	
Total Cost($)		488.500	487.87		481.7793		481.7226	

* The simulation result illustrates a violation of power balance constraint

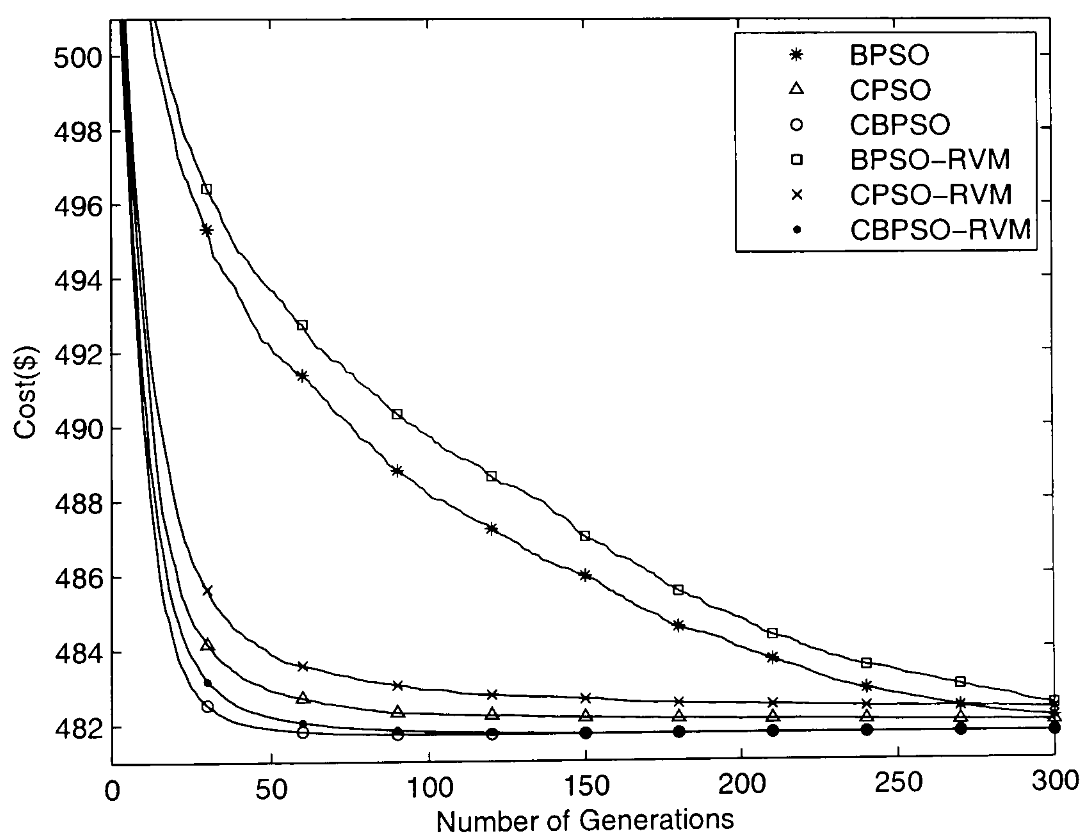


Figure 4.8 Convergence curve of the traditional and hybrid PSO algorithms for Case B (Load demand = 2400 MW)

2) Power demand = 2500MW

Table 4.6 compares the simulation results obtained from the six PSO algorithms with those of HM [93], MHNN [89], IEP [94]. The trend of simulation results are somewhat similar to the previous case (Power demand = 2400MW) in which CBPSO-

RVM and CBPSO are still perform better than others. Correspondingly, their frequencies of convergence to the final solution are also 100 in the range of $526-$527 as shown in the Table 4.7. The comparison of the best solution obtained from the proposed method (CBPSO-RVM) and some selected methods is given in Table 4.8. Concerning the convergence characteristics in Figure 4.9, both of CBPSO-RVM and its counterpart (CBPSO) outperform other PSO algorithms with respect to speed of the convergence.

Table 4.6 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case B (Load demand = 2500 MW)

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
HM*[93]	-	526.7000	-	-
MHNN*[89]	-	526.1300	-	-
IEP[94]	-	526.3040	-	-
BPSO	526.5948	526.2726	527.4458	0.2452
BPSO-RVM	526.9883	526.3064	528.4894	0.4548
CPSO	526.5261	526.2637	526.9709	0.1655
CPSO-RVM	526.9139	526.3167	527.7100	0.2793
CBPSO	526.2388	526.2388	526.2388	0.0000
CBPSO-RVM	526.2388	526.2388	526.2388	0.0000

* The simulation result illustrates a violation of power balance constraint

Table 4.7 Frequency of convergence among various methods for Case B (Load demand = 2500 MW)

Method	Cost ($)									
	526 - 527	527 - 528	528 - 529	529 - 530	530 - 531	531 - 532	532 - 533	533 - 534	534 - 535	535 - 536
BPSO	94	6			-	-	-	-	-	-
BPSO-RVM	55	41	4		-	-	-	-	-	-
CPSO	100				-	-	-	-	-	-
CPSO-RVM	66	34	-	-	-	-	-	-	-	-
CBPSO	100				-	-	-	-	-	-
CBPSO-RVM	100	-	-	-	-	-	-	-	-	-

Table 4.8 The best simulation results obtained from various methods for Case B (Load demand = 2500 MW)

Unit	HM*[93]		MHNN*[89]		IEP[94]		CBPSO-RVM	
	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)
1	2	206.6	2	206.1	2	203.0755	2	206.5197
2	1	206.5	1	206.3	1	207.2049	1	206.4586
3	1	265.9	1	265.7	1	266.9461	1	265.7375
4	3	236.0	3	235.7	3	234.5666	3	235.9540
5	1	258.2	1	258.2	1	259.8722	1	258.0180
6	3	236.0	3	235.9	3	236.8465	3	235.9521
7	1	269.0	1	269.1	1	270.8098	2	268.8661
8	3	236.0	3	235.9	3	234.3797	1	235.9530
9	1	331.6	1	331.2	1	331.4046	1	331.4843
10	1	255.2	1	255.7	1	254.8942	1	255.0568
Total Power (MW)		2501.1	2499.8		2500.0		2500.00	
Total Cost ($)		526.700	526.130		526.304		526.2388	

* The simulation result illustrates a violation of power balance constraint

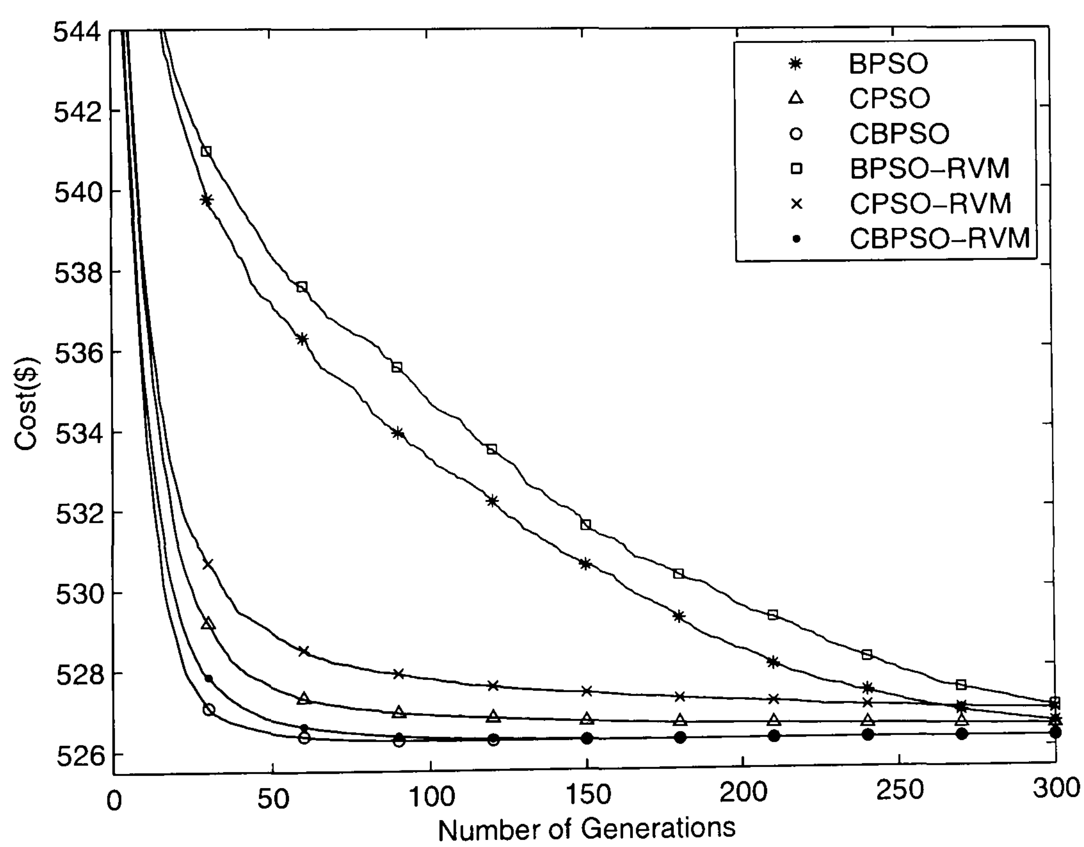


Figure 4.9 Convergence curve of the traditional and hybrid PSO algorithms for Case B (Load demand = 2500 MW)

3) Power demand = 2600MW

For this condition, there are some slight differences in the trend of simulation results. Namely, amongst the six PSO algorithms, CBPSO-RVM and its counterpart clearly outperform other PSO algorithms as well as some selected methods (i.e. HM [93], MHNN [89], and IEP [94]) in terms of the mean cost, the minimum cost, standard deviation, and convergence curve. Concerning the minimum cost in Table 4.9, it seems that CBPSO-RVM and CBPSO are slightly worse than HM [93] and MHNN [89]; however, both HM and MHNN methods violate the power balance constraint as shown in Table 4.11. Therefore, CBPSO-RVM and CBPSO are obviously superior to others in this case study.

Table 4.9 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case B (Load demand = 2600 MW)

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
HM*[93]	-	574.0300	-	-
MHNN*[89]	-	574.2600	-	-
IEP[94]	-	574.4730	-	-
BPSO	574.9592	574.4409	576.5666	0.3077
BPSO-RVM	575.3360	574.5659	576.4353	0.3894
CPSO	574.8085	574.3965	575.2636	0.2219
CPSO-RVM	575.2280	574.5098	576.1318	0.3380
CBPSO	574.5575	574.3808	574.7413	0.1811
CBPSO-RVM	574.5827	574.3808	574.7414	0.1798

* The simulation result illustrates a violation of power balance constraint

Table 4.10 Frequency of convergence among various methods for Case B (Load demand = 2600 MW)

Method	Cost ($)									
	574 - 575	575 - 576	576 - 577	577 - 578	578 - 579	579 - 580	580 - 581	581 - 582	582 - 583	583 - 584
BPSO	58	41	1	-	-	-	-	-	-	-
BPSO-RVM	16	75	9	-	-	-	-	-	-	-
CPSO	79	21	-	-	-	-	-	-	-	-
CPSO-RVM	20	76	4	-	-	-	-	-	-	-
CBPSO	100	-	-	-	-	-	-	-	-	-
CBPSO-RVM	100	-	-	-	-	-	-	-	-	-

Table 4.11 The best simulation results obtained from various method for Case B (Load demand = 2600 MW)

Unit	HM* [93]		MHNN* [89]		IEP [94]		CBPSO-RVM	
	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)
1	2	216.4	2	215.3	2	212.9858	2	216.5468
2	1	210.9	1	210.6	1	211.2541	1	210.8952
3	1	278.5	1	278.9	1	283.0745	1	278.5690
4	3	239.1	3	238.9	3	239.1562	3	239.0830
5	1	275.4	1	275.7	1	279.3285	1	275.4977
6	3	239.1	3	239.1	3	239.5382	3	239.1031
7	1	285.6	1	286.2	1	283.0806	1	285.7114
8	3	239.1	3	239.1	3	239.2435	3	239.0939
9	1	343.3	1	343.5	1	340.4721	1	343.4476
10	1	271.9	1	272.6	1	271.8665	1	272.0523
Total Power (MW)		2599.3	2599.8		2600.0		2600.0	
Total Cost ($)		574.03	574.26		574.4735		574.3808	

* The simulation result illustrates a violation of power balance constraint

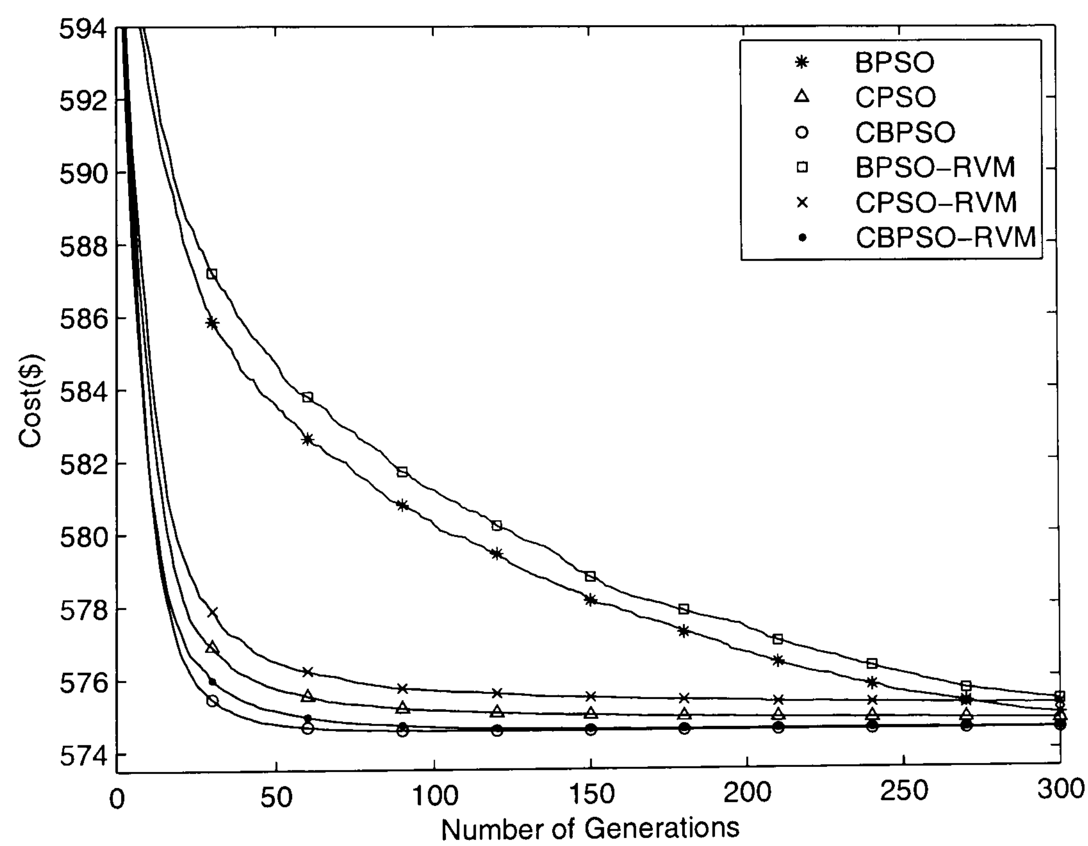


Figure 4.10 Convergence curve of the traditional and hybrid PSO algorithms for Case B (Load demand = 2600 MW)

4) Power demand = 2700MW

This sub-case compares the simulation results of six PSO algorithms with HM [93], MHNN [89], and IEP [94], the Conventional Genetic Algorithm with Multiplier Updating (CGA_MU) [92], and the Improved Genetic Algorithm with Multiplier Updating (IGA_MU) [92], respectively. It can be observed from the Table 4.12 – 4.14 and Figure 4.11 that CBPSO-RVM and CBPSO are more efficient and effective than other algorithms.

Table 4.12 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case B (Load demand = 2700 MW)

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
HM*[93]	-	625.18	-	-
MHNN*[89]	-	626.12	-	-
IEP[94]	-	623.851	-	-
CGA_MU[92]	-	623.8095	-	-
IGA_MU[92]	-	623.8093	-	-
BPSO	624.0412	623.8432	624.5751	0.1559
BPSO-RVM	624.2557	623.9105	624.8865	0.2060
CPSO	623.9839	623.8343	624.2819	0.0922
CPSO-RVM	624.2618	623.9105	624.9401	0.1949
CBPSO	623.8092	623.8092	623.8092	0.0000
CBPSO-RVM	623.8092	623.8092	623.8092	0.0000

* The simulation result illustrates a violation of power balance constraint

Table 4.13 Frequency of convergence among various methods for Case B (Load demand = 2700 MW)

Method	Cost ($)									
	623.5 - 624.5	624.5 - 625.5	625.5 - 626.5	626.5 - 627.5	627.5 - 628.5	628.5 - 629.5	629.5 - 630.5	631.5 - 632.5	632.5 - 633.5	633.5 - 634.5
BPSO	98	2	-	-	-	-	-	-	-	-
BPSO-RVM	88	12	-	-	-	-	-	-	-	-
CPSO	100	-	-	-	-	-	-	-	-	-
CPSO-RVM	89	11	-	-	-	-	-	-	-	-
CBPSO	100	-	-	-	-	-	-	-	-	-
CBPSO-RVM	100	-	-	-	-	-	-	-	-	-

Table 4.14 The best simulation results obtained from proposed method for Case B (Load demand = 2700 MW)

Unit	HM*[93]		MHNN*[89]		IEP[94]		CGA_MU[92]		IGA_MU[92]		CBPSO-RVM	
	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)
1	2	218.4	2	224.5	2	219.5362	2	218.4572	2	218.1248	2	218.2523
2	1	211.8	1	215.0	1	211.4415	1	211.5140	1	211.6826	1	211.6610
3	1	281.0	3	291.8	1	279.6780	1	280.8987	1	280.8630	1	280.7297
4	3	239.7	3	242.2	3	240.3161	3	239.6241	3	239.6533	3	239.6313
5	1	279.0	1	293.3	1	276.5291	1	278.5036	1	278.6304	1	278.4874
6	3	239.7	3	242.2	3	239.8704	3	239.6390	3	239.6140	3	239.6308
7	1	289.0	1	303.1	1	289.0023	1	288.6201	1	288.5725	1	288.5786
8	3	239.7	3	242.2	3	241.3097	3	239.6211	3	239.7057	3	239.6277
9	3	429.2	1	355.7	3	425.1423	3	428.5760	3	428.4542	3	428.5341
10	1	275.2	1	289.5	1	277.1743	1	274.5462	1	274.6995	1	274.8671
Total Power (MW)		2702.2	2699.7		2700.0		2700.0000		2700.0000		2700.0000	
Total Cost($)		625.18	626.12		623.851		623.8095		623.8093		623.8092	

* The simulation result illustrates a violation of power balance constraint

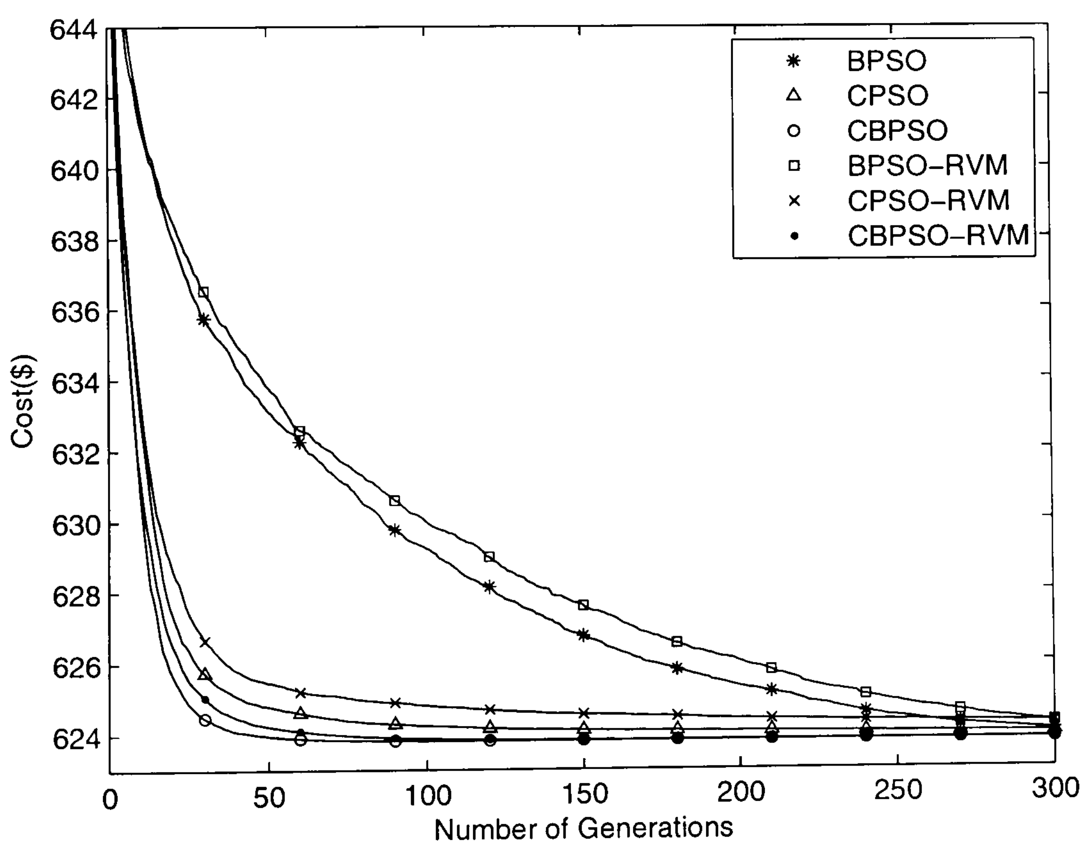


Figure 4.11 Convergence curve of the traditional and hybrid PSO algorithms for Case B (Load demand = 2700 MW)

4.5.3 ED problem with valve-point loading

Case C: 3-generator system with valve-point loading

In this case, the population size and maximum number of generations are set to 20 and 300. The power demand is assumed to be 850 MW. Based on the literature, the best solution is $8234.07 [96]. The simulation results of the six PSO algorithms are recorded in Table 4.15 so as to compare with the Genetic Algorithm (GA) [95], the Improved Evolutionary Programming (IEP) [94], the Taguchi Method (TM) [90] , the Modified PSO (MPSO) [16], the Improved Tabu Search (ITS) [96], the Classical Evolutionary Programming (CEP) [14], the Fast EP (FEP) [14], the Modified Fast EP (MFEP) [14], the Improved FEP (IFEP) [14], and the Efficient Evolutionary Strategy Optimisation (ESO) [97]. It is observed that the hybrid PSO algorithms can reach the global best solution (min. cost). Moreover, they outperform the conventional PSO algorithms and the other methods in terms of its mean cost and its standard deviation.

Table 4.15 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case C

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
GA [95]	-	8237.60	-	-
IEP [94]	-	8234.09	-	-
TM [90]	-	8234.07	-	-
MPSO [16]	-	8234.07	-	-
ITS [96]	8234.68	8234.07	8241.22	-
CEP [14]	8235.97	8234.07	8241.83	-
FEP [14]	8234.24	8234.07	8241.78	-
MFEP [14]	8234.71	8234.08	8241.80	-
IFEP [14]	8234.16	8234.07	8234.54	-
ESO [97]	8234.53	8234.07	8241.22	-
BPSO	8237.5290	8234.0717	8241.5875	3.7647
BPSO-RVM	8234.0721	8234.0717	8234.0836	0.0017
CPSO	8238.2054	8234.0717	8241.5875	3.7579
CPSO-RVM	8234.0725	8234.0717	8234.0801	0.0016
CBPSO	8240.4345	8234.0717	8382.7283	15.0920
CBPSO-RVM	8234.0717	8234.0717	8234.0718	0.0000

From the Table 4.16, it can be seen that the hybrid PSO algorithms reach the global solution in every run while BPSO, CPSO and CBPSO reach 54, 45 and 41 out of 100 runs in the range of $8234-$8236, respectively. It is also observed that CBPSO-RVM reaches the range of global solution ($8234-$8236) with the same rate as IFEP [14], but CBPSO-RVM gives a lower mean cost value. Figure 4.12 illustrates the average convergence curves of the six algorithms, while the graph shows that CBPSO-RVM is also superior to other PSO algorithms in regard to the convergence speed.

Table 4.16 Frequency of convergence among various methods for Case C

Method	Cost ($)					
	8234 - 8236	8236 - 8238	8238 - 8240	8240 - 8242	8242 - 8244	>8244
ITS [96]	92	-	-	8		
CEP [14]	76	-	-	24	-	-
FEP [14]	99	-	-	-	1	-
MFEP [14]	93	-	-	7	-	-
IFEP [14]	100	-	-	-	-	-
ESO [97]	98	-	-	2		
BPSO	54	-	-	46	-	
BPSO-RVM	100	-	-	-	-	-
CPSO	45	-	-	55	-	
CPSO-RVM	100	-	-	-	-	-
CBPSO	41	-	-	52	-	7
CBPSO-RVM	100	-	-	-	-	-

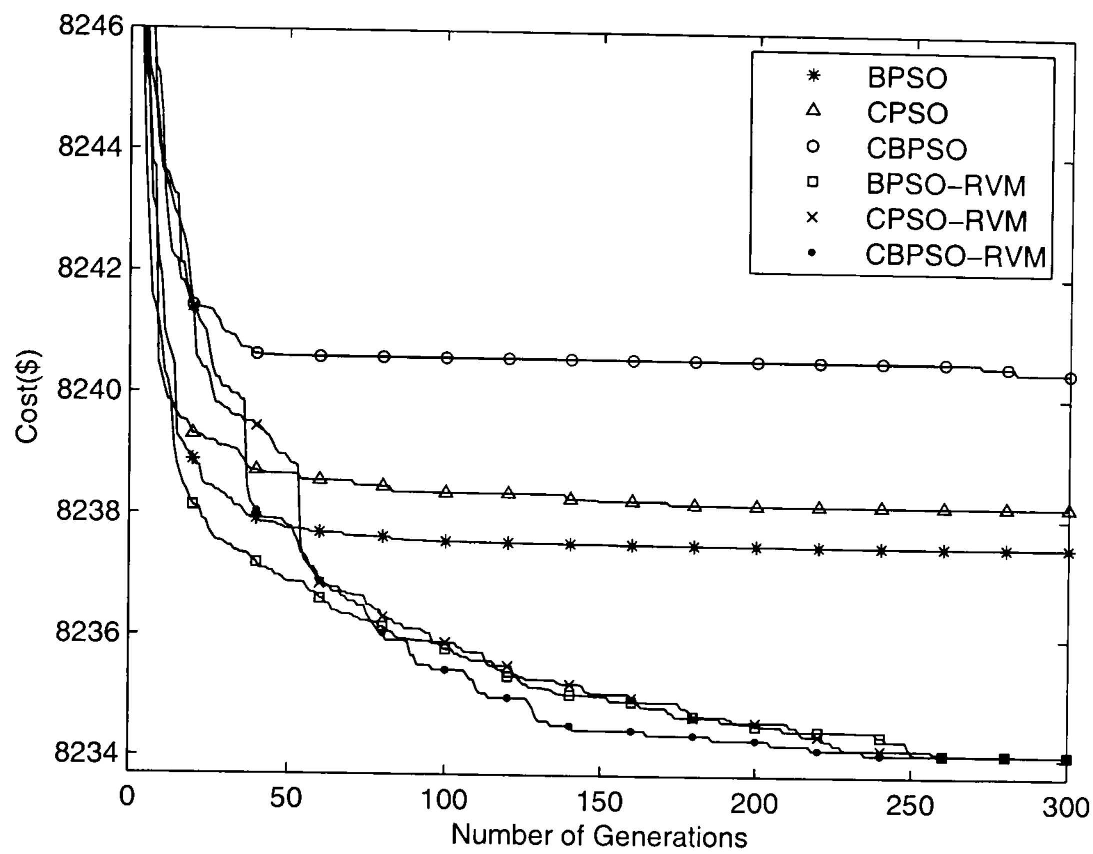


Figure 4.12 Convergence curves of the traditional and hybrid PSO algorithms for Case C

Case D: 40- generator system with valve-point loading

The parameter settings in this experiment are: population size = 60, maximum iteration = 1500, and power demand = 10500 MW. As in Case C, the simulation results of the hybrid PSO algorithms and the PSO algorithms are recorded and tabulated with the results of the CEP [14], the FEP [14], the MFEP [14], the IFEP [14], the TM [90], the MPSO [16] and the ESO [97] in Table 4.17. The results have been compared from four aspects, which are the hybrid PSO algorithms versus the other methods, the hybrid PSO algorithms versus the traditional PSO algorithms, the traditional PSO algorithms themselves, and the hybrid PSO algorithms themselves. Considering both the mean cost and the minimum cost, the hybrid PSO algorithms as well as the traditional PSO algorithms yield better results compared with the other methods. The hybrid PSO algorithms are superior to their PSO counterparts with regard to the mean cost and the standard derivation. The BPSO performs best and CBPSO yields the worst result within the traditional PSO algorithms with regard to the mean cost and the standard derivation. CBPSO-RVM outperforms the other within the hybrid PSO algorithms as far as the mean cost and the minimum cost are concerned. Table 4.18 shows the frequencies of reaching the final solution over 100 different runs obtained from the methods considered. Regarding the number of reaching minimum cost in the range of $120000-$122500, the six PSO algorithms are superior to the other algorithms. Moreover, the three hybrid PSO algorithms perform better than the three conventional PSO algorithms. Again, it can be seen that CBPSO-RVM shows its superiority to all the other methods in regard to reliability of the solutions.

Table 4.17 Comparison of calculation results obtained by the six PSO algorithms and various methods for Case D

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
CEP [14]	124,793.48	123,488.29	126,902.89	-
FEP [14]	124,119.37	122,679.71	127,245.59	-
MFEP [14]	123,489.74	122,647.57	124,356.47	-
IFEP [14]	123,382.00	122,624.35	125,740.63	-
TM [90]	123,078.21	122,477.78	124,693.81	-
MPSO [16]	-	122,252.26	-	-
ESO [97]	122,524.07	122,122.16	123,143.07	-
BPSO	122,353.87	121,835.97	122,706.12	198.39
BPSO-RVM	122,338.41	121,884.73	122,705.88	183.71
CPSO	122,469.64	121,885.11	123,767.36	307.15
CPSO-RVM	122,386.90	121,812.66	123,089.69	266.37
CBPSO	122,474.86	121,765.38	123,759.27	383.37
CBPSO-RVM	122,281.14	121,555.32	123,094.98	259.99

The average convergence curves of the hybrid PSO algorithms as well as the conventional PSO algorithms are shown in Figure 4.13. It can be seen that the convergence characteristic of CBPSO-RVM is superior to CPSO-RVM and both of them have better convergence characteristics than their PSO partners. Moreover, BPSO-RVM and BPSO, CPSO and CBPSO show very similar convergence characteristics in this case. It is interesting to see that CBPSO-RVM converges quickly up to about 100 iterations but gradually slows down afterwards while BPSO-RVM converges quickly up to about 50 iterations and then steadily approaches the minimum cost with a higher rate. The best solution obtained from CBPSO-RVM is also shown in Table 4.19.

Table 4.18 Frequency of convergence among various methods for Case D

Method	Cost ($)									
	120000-122500	122500-123000	123000-123500	123500-124000	124000-124500	124500-125000	125000-125500	125500-126000	126000-126500	126500-127000
CEP[14]	-	-	2	4	42	22	16	-	4	10
FEP[14]	-	6	24	26	20	10	2	4	-	6
MFEP[14]	-	10	50	26	14	-	-	-	-	-
IFEP[14]	-	22	50	18	4	4	-	2	-	-
TM[90]	10	22	52	12	2	2	-	-	-	-
MPSO[16]	47	53	-	-	-	-	-	-	-	-
BPSO	70	30	-	-	-	-	-	-	-	-
BPSO-RVM	76	24	-	-	-	-	-	-	-	-
CPSO	59	34	6	1	-	-	-	-	-	-
CPSO-RVM	73	24	3	-	-	-	-	-	-	-
CBPSO	58	32	9	1	-	-	-	-	-	-
CBPSO-RVM	83	16	1	-	-	-	-	-	-	-

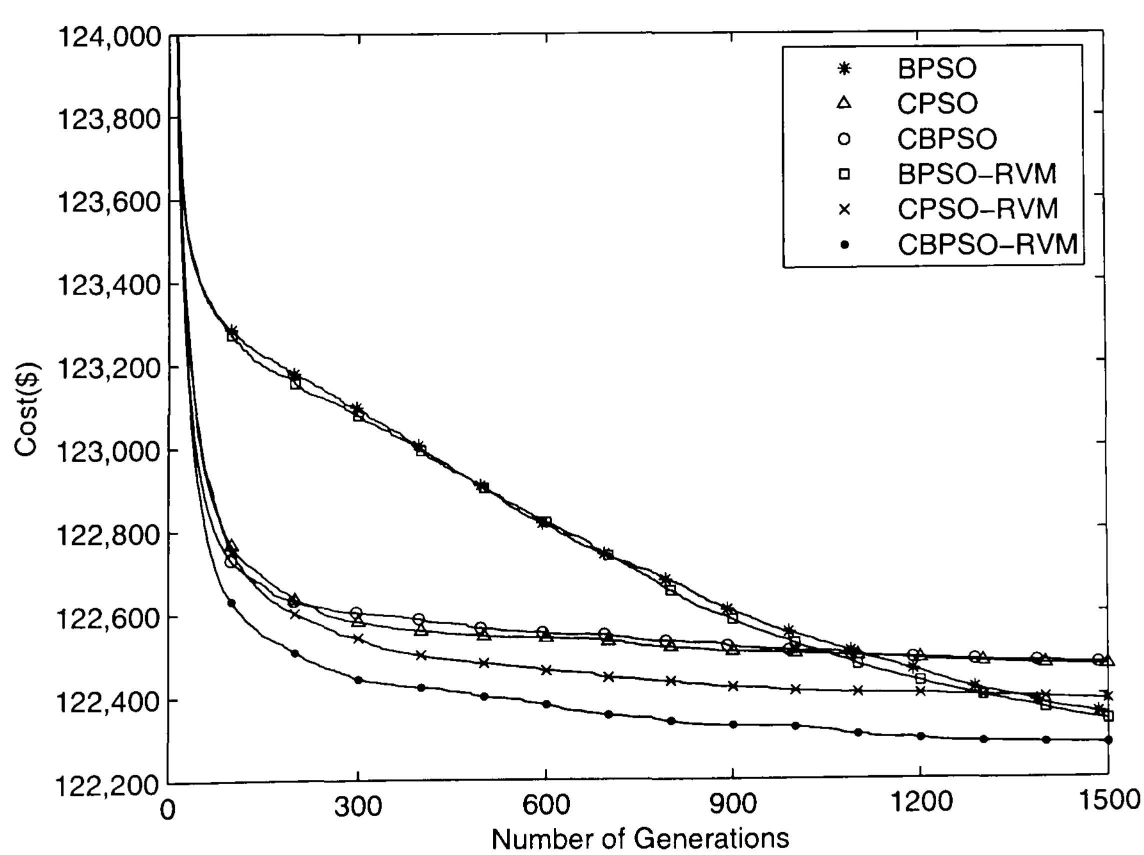


Figure 4.13 Convergence curves of the traditional and hybrid PSO algorithms for Case D

Table 4.19 The best simulation result obtained from CBPSO-RVM for Case D

Unit	Power (MW)	Cost ($)	Unit	Power (MW)	Cost ($)
1	114.0000	978.1563	21	523.2796	5071.2943
2	114.0000	978.1563	22	523.2794	5071.2903
3	97.4859	1192.2186	23	523.2797	5057.2297
4	179.7331	2143.5503	24	523.2802	5057.2398
5	97.0000	853.1776	25	523.2795	5275.0912
6	140.0000	1596.4643	26	523.2794	5275.0891
7	300.0000	3216.4240	27	10.0000	1140.5240
8	300.0000	3052.3095	28	10.0000	1140.5240
9	286.0079	2823.9515	29	10.0000	1140.5240
10	130.0000	2502.0650	30	97.0000	853.1776
11	94.0000	1893.3054	31	190.0000	1643.9913
12	94.0000	1908.1668	32	190.0000	1643.9913
13	214.7598	3792.0703	33	190.0000	1643.9913
14	304.5196	5149.6995	34	200.0000	2101.0170
15	394.2794	6436.5870	35	166.8603	1574.9864
16	394.2794	6436.5870	36	200.0000	2043.7270
17	489.2794	5296.7114	37	110.0000	1220.1661
18	489.2794	5288.7658	38	110.0000	1220.1661
19	511.2794	5540.9299	39	110.0000	1220.1661
20	511.2794	5540.9099	40	511.2794	5540.9299
Total Power(MW) and Total Cost($)				10500.00	121555.32

4.5.4 ED problem with multiple fuels and valve-point loading

Case E: 10- generator system with multiple fuels and valve-point loading

In this case, the population size, the maximum number of generations, and the power demand are set to 30, 500 and 2700, respectively. To compare with other methods, the simulation results of the hybrid PSO algorithms as well as the traditional PSO algorithms are recorded and tabulated with the results of the CGA_MU [92] ,the IGA_MU [92], the classical PSO with local random search (PSO-LRS) [62], the new PSO (NPSO) [62], and the new PSO with local random search (NPSO-LRS) [62] in Table 4.20. The simulation results illustrate that the six PSO algorithms achieve better result than other methods when the mean cost, the minimum and maximum costs, the standard deviation, and frequency of convergence to the final solution are taken into considerations. The comparison of the best solution obtained from various methods also shown in Table 4.22. Amongst the PSO algorithms, CPBSO-RVM performs somewhat similar to CBPSO in terms of solution quality and convergence characteristic. Although the simulation results of CBPSO-RVM are slightly different with CBPSO, it provides a satisfactory solution while it still maintains high quality of solution and stable convergence characteristic.

Table 4.20 Comparison of average cost and best cost among various methods for Case E

Method	Mean cost ($)	Min. cost ($)	Max. cost ($)	Std. Dev.
CGA_MU [92]	627.6087	624.7193	633.8652	-
IGA_MU [92]	625.8692	624.5178	630.8705	-
PSO-LRS [62]	625.7887	624.2297	628.3214	-
NPSO [62]	625.2180	624.1624	627.4237	-
NPSO-LRS [62]	624.9985	624.1273	626.9981	-
BPSO	624.5192	624.2434	624.8691	0.1426
BPSO-RVM	624.6990	624.2508	625.1834	0.2006
CPSO	624.4884	624.1715	624.7638	0.1188
CPSO-RVM	624.7335	624.1603	625.2964	0.1773
CBPSO	624.0170	623.9165	624.2048	0.0501
CBPSO-RVM	624.0918	623.9727	624.2467	0.0619

Table 4.21 Frequency of convergence among various methods for Case E

Method	Cost ($)										
	623.5 - 624.5	624.5 - 625.5	625.5 - 626.5	626.5 - 627.5	627.5 - 628.5	628.5 - 629.5	629.5 - 630.5	630.5 - 631.5	631.5 - 632.5	632.5 - 633.5	633.5 - 634.5
CGA_MU [92]	-	5	20	31	21	10	7	3	2	-	1
IGA_MU [92]	-	39	45	11	2	2	-	1	-	-	-
PSO-LRS [62]	5	37	36	17	5	-	-	-	-	-	-
NPSO [62]	18	54	16	12	-	-	-	-	-	-	-
NPSO-LRS [62]	20	58	17	5	-	-	-	-	-	-	-
BPSO	48	52	-	-	-	-	-	-	-	-	-
BPSO-RVM	15	85	-	-	-	-	-	-	-	-	-
CPSO	56	44	-	-	-	-	-	-	-	-	-
CPSO-RVM	7	93	-	-	-	-	-	-	-	-	-
CBPSO	100	-	-	-	-	-	-	-	-	-	-
CBPSO-RVM	100	-	-	-	-	-	-	-	-	-	-

Table 4.22 The best simulation result obtained from various methods for Case E

Unit	CGA_MU[92]		IGA_MU[92]		PSO-LRS [62]		NPSO [62]		NPSO-LRS [62]		CBPSO-RVM	
	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)	Fuel	Power (MW)
1	2	222.0108	2	219.1261	2	219.0155	2	220.6570	2	223.3352	2	213.1421
2	1	211.6352	1	211.1645	1	213.8901	1	211.7859	1	212.1957	1	212.2026
3	1	283.9455	1	280.6572	1	283.7616	1	280.4062	1	276.2167	1	281.5960
4	3	237.8052	3	238.4770	3	237.2687	3	238.6013	3	239.4187	3	239.1019
5	1	280.4480	1	276.4179	1	286.0163	1	277.5621	1	274.6470	1	279.2966
6	3	236.0330	3	240.4672	3	239.3987	3	239.1204	3	239.7974	3	240.7149
7	1	292.0499	1	287.7399	1	291.1767	1	292.1397	1	285.5388	1	285.9226
8	3	241.9708	3	240.7614	3	241.4398	3	239.1530	3	240.6323	3	240.1771
9	3	424.2011	3	429.3370	3	416.9721	3	426.1142	3	429.2637	3	429.3542
10	1	269.9005	1	275.8518	1	271.0623	1	274.4637	1	278.9541	1	278.4920
Total Power (MW)	2700		2700		2700		2700		2700		2700	
Total Cost ($)	624.7193		624.5178		624.2297		624.1624		624.1273		623.9727	

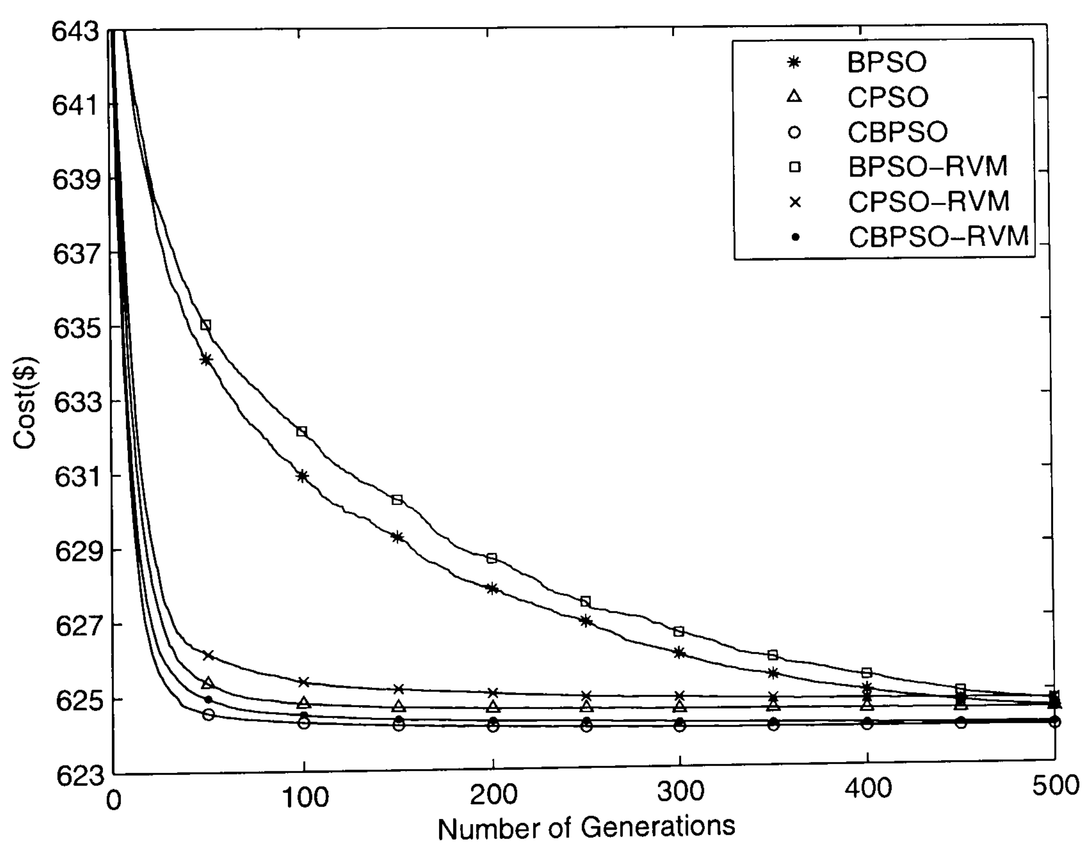


Figure 4.14 Convergence curves of the traditional and hybrid PSO algorithms for Case E

Table 4.23 lists the comparison of computation times of the algorithms under consideration. The values listed can only be regarded as a reference because the simulations using the other methods are carried out using different computers.

Table 4.23 Comparison of computation time

Method	Mean time (s)				
	Case A Smooth cost 3-Unit	Case B Multiple fuel 10-Unit	Case C Valve-point loading 3-unit	Case D Valve-point loading 40-unit	Case E Multiple fuels & Valve-point loading 10-Unit
CEP[1] [14]	-	-	20.46	1956.93	-
FEP[1] [14]	-	-	4.54	1039.16	-
MFEP[1] [14]	-	-	8.00	2196.10	-
IFEP[1] [14]	-	-	6.78	1167.35	-
TM [90]	-	-	-	94.28	-
CGA_MU [92] [2]	-	19.42	-	-	26.64
IGA_MU [92] [2]	-	5.27	-	-	7.32
BPSO[3]	0.175	3.091	0.399	10.745	5.147
BPSO-RVM[3]	0.348	5.703	0.863	29.818	8.449
CPSO[3]	0.186	3.108	0.394	10.147	5.154
CPSO-RVM[3]	0.361	5.096	0.862	29.649	8.493
CBPSO[3]	0.198	3.104	0.388	10.045	5.101
CBPSO-RVM[3]	0.359	5.094	0.854	29.225	8.512

[1] Simulations were executed on Pentium-II, 350 MHz, 128-MB RAM.
[2] Simulations were executed on Pentium-III, 700 MHz, in Fortran-90.
[3] Simulations were executed on Pentium-IV, 3GHz, 512-MB RAM.

4.6 Summary of PSO application Economic Dispatch

This chapter presents an application of the hybrid PSO with a Real-Valued Natural Mutation (RVM) to ED problem considering various cost function characteristics. In order to validate its capability, the proposed method has been applied to five different case studies as follows:

- **Case A** is the standard 3-unit system with smooth cost function,
- **Case B** is a 10-unit system with multiple fuels,
- **Case C** is a 3-unit considering valve-point loading,
- **Case D** is a 40-unit systems considering valve-point loading,
- **Case E** is a 10-unit generating units considering both multiple fuels and valve-

point loading.

Concerning the simulation results in Case A, B, and E, they show that CBPSO-RVM and CBPSO method succeed in reaching the global solution. Although the performance of CBPSO-RVM is rather similar to its counterpart (CBPSO) in these three case studies, the simulation results from Case C and Case D (3-unit and 40-unit system considering valve-point loading) confirm that the CBPSO-RVM is more powerful than other methods. Due to the fact that the characteristic of cost functions in Case A, B, and E are less non-linear and non-smooth than Case C and D, both of the algorithms perform well. However, for the latter two more difficult cases, CBPSO-RVM yields the total production cost slightly cheaper than others, which is particularly revealed when considering large-scale systems. Therefore, it can be concluded that CBPSO-RVM has a great potential and a satisfactory computation time in the practical application as ED problem.

Chapter 5: Application of PSO in Unit Commitment

5.1 Introduction

This chapter presents the application of PSO to both of the Unit Commitment (UC) problems that are the traditional UC and the profit-based UC problems. In chapters 3 and 4, CBPSO-RVM has shown the outstanding performance over the other PSO algorithms; therefore, this chapter highlights only the comparison between CBPSO-RVM and its counterpart (CBPSO) so as to make matters concise.

Moreover, this chapter is divided into two main parts. The first part focuses on the application of PSO in the traditional UC. In this section, CBPSO-RVM is applied to update the Lagrange multipliers (λ_t, μ_t) and is also incorporated into the Lagrange Relaxation (LR) method to improve its performance. Secondly, the application of PSO in a profit-based UC will be presented. For this section, CBPSO-RVM still combines with LR to enhance its capability, but CBPSO-RVM will be only utilized as updating λ_t, whereas Gradient method is used to update μ_t. The organisation of the both sections is similar: sections 5.2.1 and 5.3.1 present the problem formulation of

UC problems. Sections 5.2.2 and 5.3.2 show methodology of above discussed applications, while the simulation results are given in section 5.2.3 and 5.3.3. Finally, the conclusions are summarised in section 5.2.4 and 5.3.4.

5.2 Application of PSO in the Traditional Unit Commitment

Unit Commitment (UC) is a problem in power system operation that determines the schedule of generating units to meet electricity demand and operating constraints over a time horizon [3, 4]. It involves finding which generators will be on and how much they will generate at each time interval, over the given time horizon.

5.2.1 Problem formulation

The objective of the traditional UC problem is to minimise the sum of generation cost and start-up cost over a short term period where the objective function can be mathematically formulated by the following equation [3, 4]:

$$Minimise : TC = \sum_{t=1}^{T}\sum_{i=1}^{N}\left[F_i(P_{it}) + ST_i(1-U_{i(t-1)})\right] U_{it} \tag{5.1}$$

Subject to the following constraints:

a) Power balance

$$\sum_{i=1}^{N} P_{it}U_{it} = P_{Dt} \tag{5.2}$$

b) Spinning reserve

$$\sum_{i=1}^{N} P_{i,\max}U_{it} - P_{Dt} - SR_t \geq 0 \tag{5.3}$$

c) Operating limit

$$P_{i,\min}U_{it} \leq P_{it} \leq P_{i,\max}U_{it} \tag{5.4}$$

d) Minimum up/down time

$$U_{it} = 1 \quad for \quad \sum_{h=t-T_{i,\mathrm{up}}}^{t-1} U_{ih} < T_{i,\mathrm{up}}$$
$$U_{it} = 0 \quad for \quad \sum_{h=t-T_{i,\mathrm{down}}}^{t-1} (1-U_{ih}) < T_{i,\mathrm{down}} \tag{5.5}$$

In this study, the start-up cost is calculated as follows [98]:

$$ST_i = \begin{cases} HSC_i, & if \quad T_{i,\mathrm{off}} \le T_{i,\mathrm{down}} + CSH_i, \\ CSC_i, & \text{otherwise.} \end{cases} \tag{5.6}$$

where,

TC : total production cost ,

$F_i(P_{it})$: fuel cost of generator i given by a quadratic function

$$F_i(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i \ ,$$

ST_i : start-up cost of unit i,

U_{it} : on/off status of unit i at hour t,

P_{it} : generation output of unit i at hour t,

P_{Dt} : load demand at hour t,

SR_t : spinning reserve at hour t,

$P_{i,\min}$: minimum power output of unit i,

$P_{i,\max}$: maximum power output of unit i,

$T_{i,\mathrm{up}}$: minimum up time of unit i,

$T_{i,\mathrm{down}}$: minimum down time of unit i,

N : total number of generators,

T : total number of hours ,

$T_{i,\mathrm{off}}$: unit's off time,

HSC_i : unit's hot start-up cost,

CSC_i : unit's cold start-up cost, and

CSH_i : cold start hour.

5.2.2 Methodology

Lagrange Relaxation (LR) is one of the possible optimisation approaches to solve the UC problem. Although the computation of LR is fast, it has problems with numerical convergence and poor quality of solution [4, 5]. To overcome these, it is proposed here to incorporate the CBPSO-RVM algorithm into the LR method so as to improve its performance. The implementation processes of the proposed method are explained below:

A. Lagrange Relaxation (LR)

The basic concept of the LR procedure is to relax or ignore the coupling constraints of the UC problem (e.g. power balance and spinning reserve constraints). In addition, it decomposes the main problem into sub-problems which are easier to solve. In the LR method, Lagrange multipliers (λ_t and μ_t) are integrated into the main problem in order to create the penalty terms [1, 3, 99-101].

The main principle of the LR is to find Lagrange multipliers so as to maximise dual function (q), while minimising primal function (J). For example, if the primal problem is defined as [3, 100]:

$$min\ J = f(x_1, x_2) \tag{5.7}$$

Subject to:

$$g(x_1, x_2) \leq 0 \tag{5.8}$$

then its Lagrange function is:

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda \cdot g(x_1, x_2). \tag{5.9}$$

The dual function can then be defined as:

$$q(\lambda) = \min_{x_1, x_2} L(x_1, x_2, \lambda) \tag{5.10}$$

And the dual problem is to find

$$q^*(\lambda) = \max_{\lambda \geq 0} q(\lambda). \tag{5.11}$$

The gap between the solutions obtained for dual and primal functions is called *duality gap*, and it is expected to be minimised. However, since UC problem is non-convex finding its optimal primal value is difficult [2], and therefore the LR approach seeks to find the optimal dual value which is easer to solve. However, the updating of the Lagrange multipliers is a difficult problem and it will considerably affect the quality of the solution [102]. Up till now, a number of approaches have been proposed to update Lagrange multipliers i.e. sub-gradient method, modified sub-gradient method, multiplier adjustment method, reduced complexity bundle method, etc [103-105].

In the case of problem formulation given in (5.1)-(5.6), the Lagrange function can be formed as follows [3, 99]:

$$\begin{aligned} L(P,U,\lambda,\mu) = TC(P_{it},U_{it}) + \sum_{t=1}^{T} \lambda_t (P_{Dt} - \sum_{i=1}^{N} P_{it} U_{it}) \\ + \sum_{t=1}^{T} \mu_t (P_{Dt} + SR_t - \sum_{i=1}^{N} P_{i,\max} U_{it}). \end{aligned} \tag{5.12}$$

As described above, from the concept of dual optimisation, we can obtain the values of the Lagrange multipliers by maximising the Lagrange function (L) with respect to the Lagrange multipliers λ_t and μ_t, whilst minimising with respect to P_{it} and U_{it}, that is

$$q^*(\lambda, \mu) = \max_{\lambda_t, \mu_t} q(\lambda, \mu) \tag{5.13}$$

where

$$q(\lambda,\mu)=\min_{P_{it},U_{it}} L(P,U,\lambda,\mu) \tag{5.14}$$

and

$$\min q(\lambda,\mu)=\sum_{i=1}^{N}\min\sum_{t=1}^{T}[F_i(P_{it})+ST_i(1-U_{i(t-1)})-\lambda_t P_{it}-\mu_t P_{i,\max}]U_{it} \tag{5.15}$$

subject to the operating limit constraints and minimum up/down time constraints.

Often, two-state dynamic programming is applied to solve (5.15). The main idea of applying two-state dynamic programming is to find the path that offers minimum total cost, which consists of the summation of the fuel cost and start-up cost, up to the current hour. So in each hour, there are two possible states for a generator (i.e. "1" = on and "0"= off), and their total costs are compared to make a decision for choosing the minimum cost in that path. Moreover, minimum up/down time constraints will also be taken into account as illustrated in Figure 5.3. Thereafter, the minimum cost and its path from the previous hour will be stored and the same processes carried out as in the earlier step until the final hour [5].

B. Incorporation of PSO into LR for UC

Conventionally, LR uses a Gradient method to update the Lagrange multipliers. It has a limitation however in that sometimes the solution is trapped in a local optimum causing a convergence problem [106]. In this section, a new hybrid method (LR-CBPSO_RVM) is proposed to solve the UC problem. CBPSO_RVM is applied to update the Lagrange multipliers and is also incorporated into the LR method to improve its performance. Note that, in general applying, CBPSO_RVM to update both λ_t and μ_t can cause problems due to the curse of dimensionality, however, the following analyses were preformed using this method for the update of both types of multipliers. The flow chart describing the procedures of the proposed method is shown in Figure 5.1.

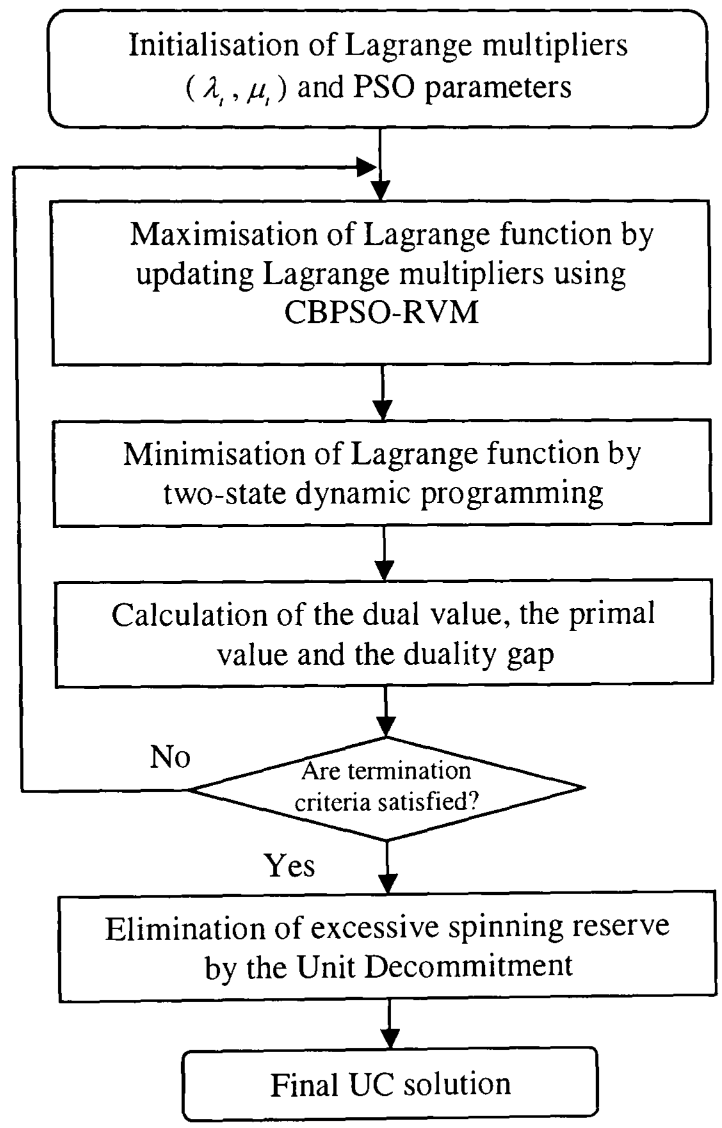


Figure 5.1 The basic flow chart of the proposed method

The steps of the computation method as presented in Figure 5.1 are discussed below.

Step 1: Initialisation of Lagrange multipliers and PSO parameters

- Generate an initial population of particles (λ_t and μ_t). Normally, each particle is generated randomly within an allowable range. The members of the population are stored in a matrix form which defines the Lagrange multipliers as shown in Figure 5.2.
- Subsequently, initialise the parameters of the PSO algorithm (e.g. population size, initial/final inertia weight, velocity of particle, acceleration constant, constriction factor, number of particles in mutation, the maximum generation, and the duality gap).

- Define each particle as *pbest* and the best position of all particles as *gbest*.

Hour

$$\text{Individual No.}\quad \begin{array}{c} \\ 1 \\ 2 \\ \cdot \\ \cdot \\ N \end{array} \begin{array}{cccccccccc} 1 & 2 & \cdot & \cdot & T & 1 & 2 & \cdot & \cdot & T \end{array}$$

$$\begin{bmatrix} \lambda_1^1 & \lambda_2^1 & \cdots & \lambda_{T-1}^1 & \lambda_T^1 & \mu_1^1 & \mu_2^1 & \cdots & \mu_{T-1}^1 & \mu_T^1 \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_{T-1}^2 & \lambda_T^2 & \mu_1^2 & \mu_2^2 & \cdots & \mu_{T-1}^2 & \mu_T^2 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \lambda_1^{N-1} & \lambda_2^{N-1} & \cdots & \lambda_{T-1}^{N-1} & \lambda_T^{N-1} & \mu_1^{N-1} & \mu_2^{N-1} & \cdots & \mu_{T-1}^{N-1} & \mu_T^{N-1} \\ \lambda_1^N & \lambda_2^N & \cdots & \lambda_{T-1}^N & \lambda_T^N & \mu_1^N & \mu_2^N & \cdots & \mu_{T-1}^N & \mu_T^N \end{bmatrix}$$

$$\underbrace{\qquad\qquad}_{\lambda} \qquad\qquad \underbrace{\qquad\qquad}_{\mu}$$

Figure 5.2 Population in the form of a matrix

Step 2: Maximisation of Lagrange function by updating Lagrange multipliers using PSO

- Calculate the evaluation value or dual value (q) of each individual (λ_t, μ_t) as follows.

$$\begin{aligned} q(\lambda,\mu) = & \sum_{i=1}^{N}\sum_{t=1}^{T}[F_i(P_{it}) + ST_i(1-U_{i(t-1)})]U_{it} \\ & -\sum_{i=1}^{N}\sum_{t=1}^{T}(\lambda_t P_{it} + \mu_t P_{i,\max})U_{it} + \sum_{t=1}^{T}\left(\lambda_t P_{Dt} + \mu_t(P_{Dt} + SR_t)\right) \end{aligned} \tag{5.16}$$

- Compare each evaluation value with the previous *pbest*. If the current value is more, let it be *pbest*. Similarly, if the best value in *pbest*'s group is more than *gbest*, let the value be *gbest*.

- Update the member velocity (v) of each individual (λ_i, μ_i) by (3.5).

- If $v_{id}^{(t+1)} > V_{d,\max}$, then $v_{id}^{(t+1)} = V_{d,\max}$ or if $v_{id}^{(t+1)} < -V_{d,\max}$, then $v_{id}^{(t+1)} = -V_{d,\max}$. The maximum velocity can be calculated as follows [18]:

$$V_{d,\max} = \frac{(x_{id,\max} - x_{id,\min})}{N} \tag{5.17}$$

where N is a chosen number of intervals.

- Update the member position of each individual (λ_i, μ_i) from (3.6).

- Apply the Real-Coded Mutation Operator (RVM) to the swarm.

Step 3: Minimisation of Lagrange function by two-state dynamic programming

- Minimise the Lagrange function using two-state dynamic programming for P_{it} and U_{it}, where $i = 1...N$, and $t = 1...T$.
- Use the Forward Dynamic Programming (FDP) to solve the dual problem. The objective is to minimise q. Figure 5.3 illustrates the concept of two-state dynamic programming [4].

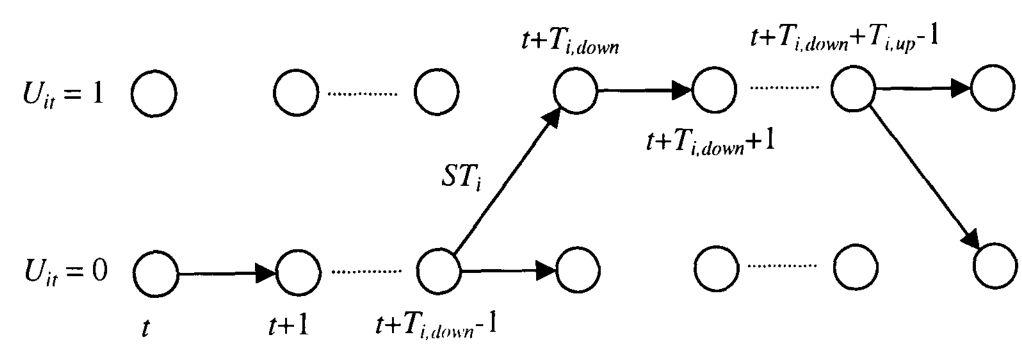


Figure 5.3 Two-state dynamic programming [4]

Step 4: Calculation of the dual value, the primal value, and the duality gap

- Determine the dual value from (5.16) using P_{it} and U_{it} obtained from step 3.
- To solve the economic dispatch problem, use U_{it} from Step 3 to obtain P_{it}^*, and then calculate the primal value (J).

$$J = \sum_{t=1}^{T}\sum_{i=1}^{N}[F_i(P_{it}^*) + ST_i(1 - U_{i(t-1)})]U_{it} \tag{5.18}$$

- The difference between the primal and dual values, named the duality gap (ε), is used as a terminating criterion. The duality gap can be calculated from

$$\varepsilon = \frac{J - q}{q}. \tag{5.19}$$

Step 5: If either the predefined maximum number of generations is reached, or the duality gap is less than a setting threshold, then stop. The latest P_{it}^{*} is the optimal solution. Otherwise, return to Step 2.

Step 6: Elimination of excessive spinning reserve by the Unit Decommitment

According to [5], over committed units in some hours may lead to an excessive spinning reserve requirement that will result in high total production cost. Accordingly, the elimination of excessive spinning reserve is necessary to apply for this case. To deal with this problem, a heuristic search method (Unit Decommitment) is adopted, as presented in [5, 107]. The procedures of the Unit Decommitment are shown in Figure 5.4.

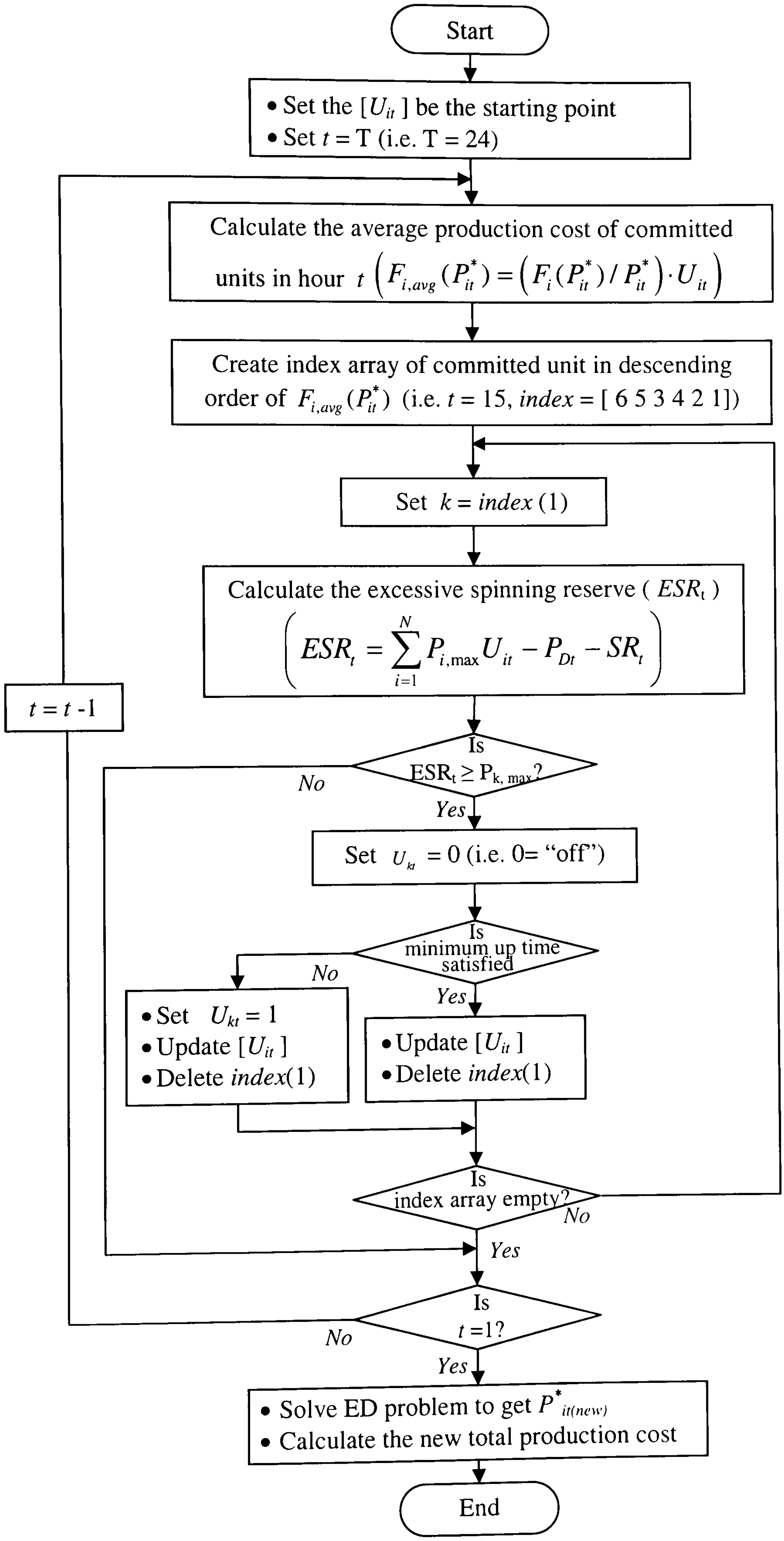


Figure 5.4 Flow chart of the Unit Decommitment for eliminating excessive spinning reserve

5.2.3 Simulation Results

In this section, the proposed LR-CBPSO_RVM method is applied to solve the UC problem. In order to illustrate its effectiveness, the method is applied to two different systems, namely a 3-unit 4-hour system and a 10-unit 24-hour system. The data used in both cases are adopted from [3] and [2] and the details are in Appendix B. The simulations are carried out using Matlab. The parameters of the PSO used in all simulations are: initial inertia weight (w_{max}) = 0.9; final inertia weight (w_{min}) = 0.4; acceleration constants (c_1, c_2) = 2.05 and constriction factor (k) = 0.73 and number of particles in mutation (N_m) = 1, respectively.

Case A: 3-unit, 4-hour system

In this case, the simulation parameters of the proposed method are population size = 40, number of runs = 30, maximum number of generations = 18, and duality gap = 0.02. For the LR method [3], the parameters for simulation are α_1 = 0.01 and α_2 = 0.002. Furthermore, spinning reserve is not considered; therefore the elimination of excessive spinning reserve section will be neglected. To examine the effectiveness of the proposed method, the simulation results are compared with the results obtained from the LR method, which is re-implemented. The optimal solution is \$20162.75 as reported in [4]. From the simulation results, the proposed method reaches the optimal solution (\$20162.75) in every run. Table 5.1 presents the optimal solution obtained from the proposed method (LR-CBPSO_RVM). Furthermore, the comparison of the average convergence curves between the proposed method and the LR method are demonstrated in Figure 5.5. It can be observed that the LR method has a numerical convergence problem. Furthermore, for the LR method, the stopping criterion is satisfied in the 18th iteration, while the duality gap (ε = 0.0185) of the proposed method satisfies the stopping criterion already at the 10th iteration. From the comparison of the two methods, it is clearly shown that the proposed modification for the LR method is more effective than the traditional LR method in terms of overcoming the convergence problem and closing the duality gap.

Table 5.1 The optimal solution obtained from the proposed method (LR-CBPSO_RVM)

Hour	Load (MW)	Unit Number			Fuel cost ($)
		1	2	3	
1	170	0	0	170	1264.50
2	520	0	320	200	4616.00
3	1100	500	400	200	11400.00
4	330	0	130	200	2882.25
Total					20162.75

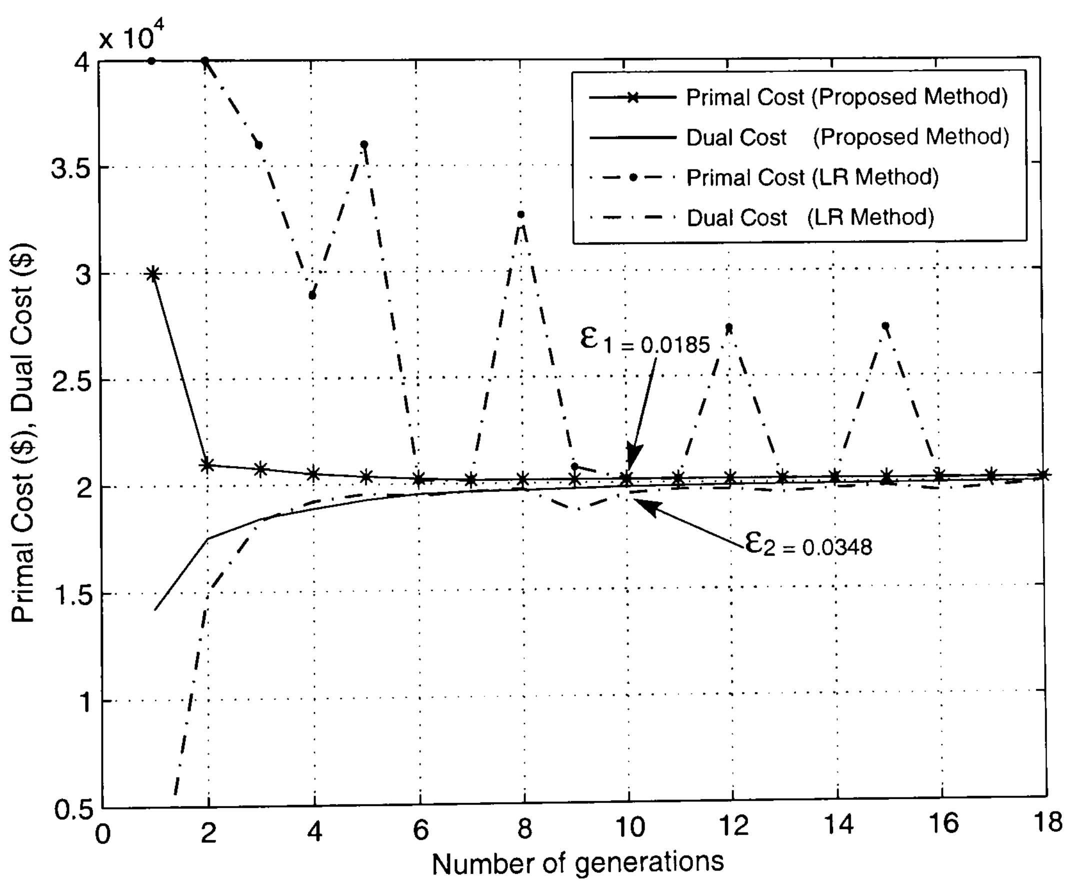


Figure 5.5 Comparison of convergence curves between the proposed method and the LR method (ε_1 - Duality gap of the proposed method and ε_2 - Duality gap of the LR method)

Case B: 10-unit, 24-hour system

For this case, the set parameters of the proposed method is population size = 100. Since the spinning reserve will be taken into account, it is assumed to be 10% of the load demand. To investigate the effect of excessive spinning reserve, two versions of the LR-CBPSO_RVM method, one including and one without the Unit

Decommitment are simulated. Table 5.2-5.4 compares the best solutions obtained from the LR method, the proposed LR-CBPSO_RVM method without the Unit Decommitment, and the proposed LR-CBPSO_RVM method, respectively. The total cost of the LR method is $565823, while the total cost of the proposed method without applying the Unit Decommitment is $565275, which is $548 less than the LR method. In addition, the total cost obtained from the proposed LR-CBPSO_RVM method with Unit Decommitment is $455 cheaper than the total cost obtained from the same method without applying the Unit Decommitment. The saving in total cost is a consequence of elimination of excessive spinning reserve in the 15th hour, as illustrated in Table 5.3. In addition, Table 5.5 compares the result of the proposed method with some other methods from literature. From the simulation results, it can therefore be concluded that the performance of the proposed LR-CBPSO_RVM method is better than other methods in terms of total production cost. Since the simulations were carried out on different types of computers, the computation time will not be compared here.

Table 5.2 The best solution obtained from the LR method

Hour	Load (MW)	Generation schedule (MW)										Fuel cost ($)	Startup cost ($)	Total cost ($)
		U1	U2	U3	U4	U5	U6	U7	U8	U9	U10			
1	700	455	245	0	0	0	0	0	0	0	0	13683.13	0	13683.13
2	750	455	295	0	0	0	0	0	0	0	0	14554.50	0	14554.50
3	850	455	370	0	0	25	0	0	0	0	0	16809.45	900	17709.45
4	950	455	340	0	130	25	0	0	0	0	0	19145.70	560	19705.70
5	1000	455	390	0	130	25	0	0	0	0	0	20020.02	0	20020.02
6	1100	455	360	130	130	25	0	0	0	0	0	22387.04	1100	23487.04
7	1150	455	410	130	130	25	0	0	0	0	0	23261.98	0	23261.98
8	1200	455	455	130	130	30	0	0	0	0	0	24150.34	0	24150.34
9	1300	455	455	130	130	85	20	25	0	0	0	27251.06	860	28111.06
10	1400	455	455	130	130	162	33	25	10	0	0	30057.55	60	30117.55
11	1450	455	455	130	130	162	73	25	10	10	0	31916.06	60	31976.06
12	1500	455	455	130	130	162	80	25	43	10	10	33890.16	60	33950.16
13	1400	455	455	130	130	162	33	25	10	0	0	30057.55	0	30057.55
14	1300	455	455	130	130	85	20	25	0	0	0	27251.06	0	27251.06
15	1200	455	440	130	130	25	20	0	0	0	0	24605.73	0	24605.73
16	1050	455	310	130	130	25	0	0	0	0	0	21513.66	0	21513.66
17	1000	455	260	130	130	25	0	0	0	0	0	20641.82	0	20641.82
18	1100	455	360	130	130	25	0	0	0	0	0	22387.04	0	22387.04
19	1200	455	415	130	130	25	20	25	0	0	0	25341.60	430	25771.60
20	1400	455	455	130	130	162	33	25	10	0	0	30057.55	60	30117.55
21	1300	455	455	130	130	85	20	25	0	0	0	27251.06	0	27251.06
22	1100	455	360	130	130	25	0	0	0	0	0	22387.04	0	22387.04
23	900	455	420	0	0	25	0	0	0	0	0	17684.69	0	17684.69
24	800	455	345	0	0	0	0	0	0	0	0	15427.42	0	15427.42
Total												561733.23	4090	565823.23

Table 5.3 The best solution obtained from the proposed LR-CBPSO_RVM method without the Unit Decommitment

Hour	Load (MW)	Generation schedule (MW)										Fuel cost ($)	Startup cost ($)	Total cost ($)
		U1	U2	U3	U4	U5	U6	U7	U8	U9	U10			
1	700	455	245	0	0	0	0	0	0	0	0	13683.13	0	13683.13
2	750	455	295	0	0	0	0	0	0	0	0	14554.50	0	14554.5
3	850	455	370	0	0	25	0	0	0	0	0	16809.45	900	17709.45
4	950	455	455	0	0	40	0	0	0	0	0	18597.67	0	18597.67
5	1000	455	390	0	130	25	0	0	0	0	0	20020.02	560	20580.02
6	1100	455	360	130	130	25	0	0	0	0	0	22387.04	1100	23487.04
7	1150	455	410	130	130	25	0	0	0	0	0	23261.98	0	23261.98
8	1200	455	455	130	130	30	0	0	0	0	0	24150.34	0	24150.34
9	1300	455	455	130	130	85	20	25	0	0	0	27251.06	860	28111.06
10	1400	455	455	130	130	162	33	25	10	0	0	30057.55	60	30117.55
11	1450	455	455	130	130	162	73	25	10	10	0	31916.06	60	31976.06
12	1500	455	455	130	130	162	80	25	43	10	10	33890.16	60	33950.16
13	1400	455	455	130	130	162	33	25	10	0	0	30057.55	0	30057.55
14	1300	455	455	130	130	85	20	25	0	0	0	27251.06	0	27251.06
15	1200	455	440	130	130	25	20	0	0	0	0	24605.73	0	24605.73
16	1050	455	310	130	130	25	0	0	0	0	0	21513.66	0	21513.66
17	1000	455	260	130	130	25	0	0	0	0	0	20641.82	0	20641.82
18	1100	455	360	130	130	25	0	0	0	0	0	22387.04	0	22387.04
19	1200	455	415	130	130	25	20	25	0	0	0	25341.60	430	25771.6
20	1400	455	455	130	130	162	33	25	10	0	0	30057.55	60	30117.55
21	1300	455	455	130	130	85	20	25	0	0	0	27251.06	0	27251.06
22	1100	455	360	130	130	25	0	0	0	0	0	22387.04	0	22387.04
23	900	455	420	0	0	25	0	0	0	0	0	17684.69	0	17684.69
24	800	455	345	0	0	0	0	0	0	0	0	15427.42	0	15427.42
Total												561185.19	4090	565275.2

Table 5.4 The best solution obtained from the proposed LR-CBPSO_RVM method

Hour	Load (MW)	Generation schedule (MW)										Fuel cost ($)	Startup cost ($)	Total cost ($)
		U1	U2	U3	U4	U5	U6	U7	U8	U9	U10			
1	700	455	245	0	0	0	0	0	0	0	0	13683.13	0	13683.13
2	750	455	295	0	0	0	0	0	0	0	0	14554.50	0	14554.50
3	850	455	370	0	0	25	0	0	0	0	0	16809.45	900	17709.45
4	950	455	455	0	0	40	0	0	0	0	0	18597.67	0	18597.67
5	1000	455	390	0	130	25	0	0	0	0	0	20020.02	560	20580.02
6	1100	455	360	130	130	25	0	0	0	0	0	22387.04	1100	23487.04
7	1150	455	410	130	130	25	0	0	0	0	0	23261.98	0	23261.98
8	1200	455	455	130	130	30	0	0	0	0	0	24150.34	0	24150.34
9	1300	455	455	130	130	85	20	25	0	0	0	27251.06	860	28111.06
10	1400	455	455	130	130	162	33	25	10	0	0	30057.55	60	30117.55
11	1450	455	455	130	130	162	73	25	10	10	0	31916.06	60	31976.06
12	1500	455	455	130	130	162	80	25	43	10	10	33890.16	60	33950.16
13	1400	455	455	130	130	162	33	25	10	0	0	30057.55	0	30057.55
14	1300	455	455	130	130	85	20	25	0	0	0	27251.06	0	27251.06
15	1200	455	455	130	130	30	0	0	0	0	0	24150.34	0	24150.34
16	1050	455	310	130	130	25	0	0	0	0	0	21513.66	0	21513.66
17	1000	455	260	130	130	25	0	0	0	0	0	20641.82	0	20641.82
18	1100	455	360	130	130	25	0	0	0	0	0	22387.04	0	22387.04
19	1200	455	415	130	130	25	20	25	0	0	0	25341.60	430	25771.60
20	1400	455	455	130	130	162	33	25	10	0	0	30057.55	60	30117.55
21	1300	455	455	130	130	85	20	25	0	0	0	27251.06	0	27251.06
22	1100	455	360	130	130	25	0	0	0	0	0	22387.04	0	22387.04
23	900	455	420	0	0	25	0	0	0	0	0	17684.69	0	17684.69
24	800	455	345	0	0	0	0	0	0	0	0	15427.42	0	15427.42
Total												560729.80	4090	564819.80

Table 5.5 Comparison of simulation results

Method	Total production costs ($)
LR [2]	565,825
GA [2]	565,825
HPSO [63]	574,153
LR*	565,823
LR-CBPSO_RVM**	565,275
LR-CBPSO_RVM	564,820

* Re-implemented the LR method

** The proposed LR-CBPSO_RVM method without Unit Decommitment

5.2.4 Summary of PSO application in the Traditional Unit Commitment

This section presents a new methodology, called LR-CBPSO_RVM or the Hybrid Particle Swarm Optimisation (CBPSO-RVM) combined with Lagrange Relaxation (LR) method, to solve the UC problem. Applying the LR-CBPSO_RVM method improves the performance of the LR method since PSO is used to update the Lagrange multipliers. To illustrate its performance, the proposed method is tested on 3-unit 4-hr system and 10-unit 24-hr system. Compared with the LR, the Genetic Algorithm (GA) as well as the Hybrid Particle Swarm Optimisation (HPSO) methods, the proposed LR-CBPSO_RVM method has provided a satisfactory performance in terms of solution quality. Furthermore, it could be extended to solve a large-scale system and a profit-based unit commitment problem under the competitive environment of power systems.

5.3 Application of PSO in A Profit-Based Unit Commitment

With respect to restructuring in electrical industry, the competitiveness becomes the main factor under new structure. Concerning the power system operation, the main objective completely contrasts with the previous one in respect to business profits

[108].

5.3.1 Problem formulation

The objective of the profit-based UC problem is to maximise the expected profit of the generation companies (GENCOs) over a short term period rather than minimise cost as shown in traditional UC problem [109, 110]. The calculation of the expected profit can be represented as the difference between the expected revenue and cost. Namely, GENCOs can calculate the expected revenue and cost based on the forecasted values of price, demand, as well as spinning reserve. Thus, the objective function can be presented by the following equation [106]:

$$Maximise \sum_{i=1}^{N}\sum_{t=1}^{T}\left(Expected\ Revenue_{it} - Cost_{it}\right) \tag{5.20}$$

Subject to the following constraints:

a) Power demand

$$\sum_{i=1}^{N} P_{it}U_{it} - P_{Dt} \leq 0 \tag{5.21}$$

b) Spinning reserve

$$\sum_{i=1}^{N} R_{it}U_{it} - SR_t \leq 0 \tag{5.22}$$

c) Operating limit

$$P_{i,\min} \leq P_i \leq P_{i,\max} \tag{5.23}$$

$$0 \leq R_i \leq P_{i,\max} - P_{i,\min} \tag{5.24}$$

$$R_i + P_i \leq P_{i,\max} \tag{5.25}$$

d) Minimum up/down time

$$
\begin{aligned}
U_{it} &= 1 \quad for \quad \sum_{h=t-T_{i,\mathrm{up}}}^{t-1} U_{ih} < T_{i,\mathrm{up}} \\
U_{it} &= 0 \quad for \quad \sum_{h=t-T_{i,\mathrm{down}}}^{t-1} (1-U_{ih}) < T_{i,\mathrm{down}}
\end{aligned}
\tag{5.26}
$$

In this study, the payment for reserve allocated is adopted to validate the performance of the proposed method. This payment method is originally introduced by Allen and Ilic [111] and then, Attavirayanupap *et al.* [109] and Yu *et al.* [112] have adopted it to investigate the effectiveness of their proposed methods. For this payment method, it can be concluded that GENCOs will get the payment from selling the reserve whether the reserve is used or not. Namely, if the reserve is not used, the reserve price will be taken into account. In contrast, the spot price will be applied when the reserve is used. From the details above, reserve price in this case is very cheap compared with the sport price [111]. Therefore, the expected revenue and cost can be calculated as shown below.

$$
Expected\ Revenue_{it} = \left(P_{it} \cdot \rho_{SP_t} + \left((1-r)\cdot \rho_{RP_t} + r \cdot \rho_{SP_t}\right) R_{it}\right) \cdot U_{it} \tag{5.27}
$$

$$
Cost_{it} = \left((1-r)F(P_{it}) + r \cdot F(P_{it} + R_{it}) + ST_i\right) \cdot U_{it} \tag{5.28}
$$

where,

$F_i(P_{it})$: fuel cost of generator i given by a quadratic function

$F_i(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i$,

P_{it} : generation output of unit i at hour t,

R_{it} : spinning reserve of unit i at hour t,

ST_i : start-up cost of unit i,calculated by (5.6),

P_{Dt} : load demand at hour t,

U_{it} : on/off status of unit i at hour t,

$P_{i,\min}$: minimum power output of unit i,

$P_{i,\max}$: maximum power output of unit i,

$T_{i,\text{up}}$: minimum up time of unit i,

$T_{i,\text{down}}$: minimum down time of unit i,

N : total number of generators,

T : total number of hours,

ρ_{SP_t} : forecasted spot price at hour t,

ρ_{RP_t} : forecasted spinning reserve price at hour t,

r : estimated probability that spinning reserve is called and generated.

5.3.2 Methodology

In this section, both Lagrange Relaxation (LR) method and LR-CBPSO_RVM are applied to solve the profit-based UC problem. As discussed in section 5.2.3, for the previous case for the traditional UC formulation, LR suffers from the convergence and solution quality problems. Thus, CBPSO_RVM is utilised to enhance the performance of LR, especially in updating Lagrange multipliers. The implementation processes are discussed as follows:

A. Lagrange Relaxation (LR)

Applying LR to profit-based UC is practically the same as the traditional UC. Namely, ignoring the coupling constraints and separating the main problem into sub-problems are still the main concept [3],[99],[1]. The Lagrange function can be therefore formed as follows [109]:

$$L(P,R,\lambda,\mu)=\sum_{i=1}^{N}\sum_{t=1}^{T}\left(Cost_{it}-Expected\ Revenue_{it}\right)-\sum_{t=1}^{T}\lambda_t(P_{Dt}-\sum_{i=1}^{N}P_{it}U_{it}) \\ -\sum_{t=1}^{T}\mu_t(SR_t-\sum_{i=1}^{N}R_{it}U_{it}) \quad (5.29)$$

It is quite similar to the section 5.2.2 in terms of dual optimisation concept. Therefore, we can obtain the values of the Lagrange multipliers by maximising the Lagrange function (L) with respect to the Lagrange multipliers λ_t and μ_t, whilst minimising with respect to P_{it}, R_{it} and U_{it}, that the minimum of L can be formed as follows:

$$\min q(\lambda,\mu)=\sum_{i=1}^{N}\min\sum_{t=1}^{T}\{(1-r)F(P_{it})+r\cdot F(P_{it}+R_{it})+ST_t \\ -\left(P_{it}\cdot\rho_{SP_t}+\left((1-r)\cdot\rho_{RP_t}+r\cdot\rho_{SP_t}\right)R_{it}\right)+\lambda_t P_{it}+\mu_t R_{it}\}\cdot U_{it}. \quad (5.30)$$

subject to the operating limit constraints and minimum up/down time constraints that are (5.23)-(5.26).

B. Incorporation of PSO into LR for the profit-based UC

In the section of the traditional UC, first a gradient method was used for updating the Lagrange multipliers (λ_t and μ_t), and then a hybrid method (LR-CBPSO_RVM) is proposed to solve the problem. The modification of the LR consist of using CBPSO-RVM to update the λ_t and μ_t. As in the case of the traditional UC, the effect of using CBPSO_RVM to update both λ_t and μ_t may lead to the problem of the curse of dimensionality. In this part, the hybrid PSO algorithm (CBPSO-RVM) will be used first to update both λ_t and μ_t, and then the results will be compared with the case when only λ_t are updated via CBPSO-RVM, whilst μ_t is updated by the Gradient method as proposed by [64]. The flow chart describing the procedures of the proposed method is illustrated in Figure 5.6.

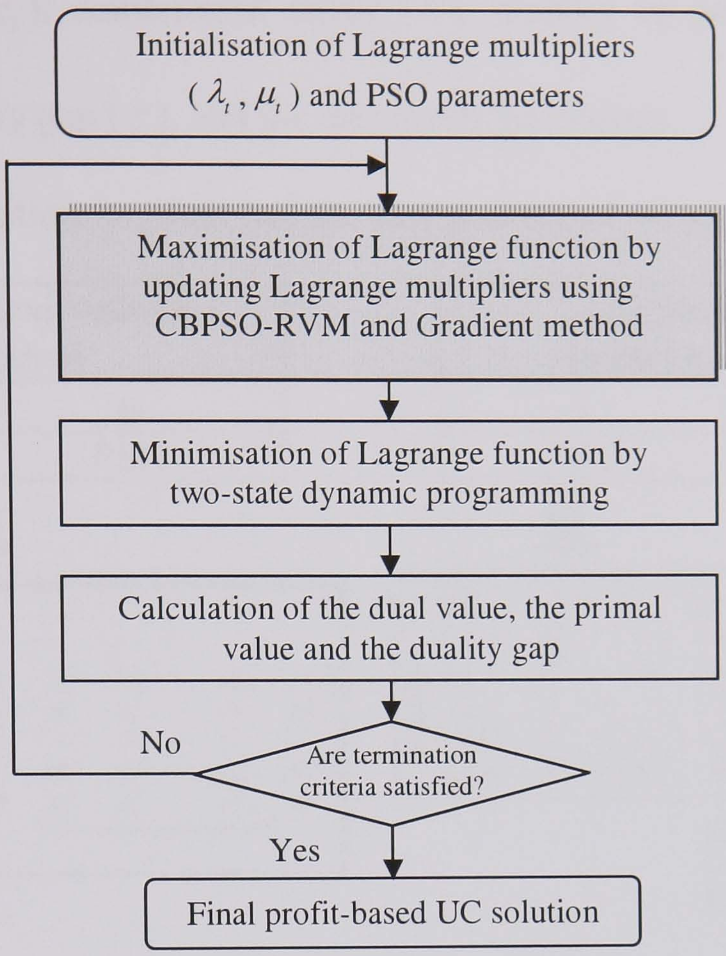


Figure 5.6 The basic flow chart of the proposed method for the profit-based UC.

The steps of the computation method as presented in Figure 5.6 are discussed below.

Step 1: Initialisation of Lagrange multipliers and PSO parameters

- Generate an initial population of particles (λ_t) within an allowable range randomly.
- Set initial of values of μ_t to zero as recommended in [3]. Note that as mentioned above, updating of Lagrange multipliers μ_t is done both by using CBPSO-RVM and by applying Gradient method. This affects the structure and the values of the initial values of Lagrange multipliers as indicated in Figure 5.7.
- Subsequently, initialise the parameters of the PSO algorithm e.g. population size, initial/final inertia weight (w), velocity of particle (v), acceleration

constants (c_1, c_2), constriction factor (k), number of particles in mutation (N_m), the duality gap (ε), and the maximum generation.

- Define each particle as *pbest* and the best position of all particles as *gbest*.

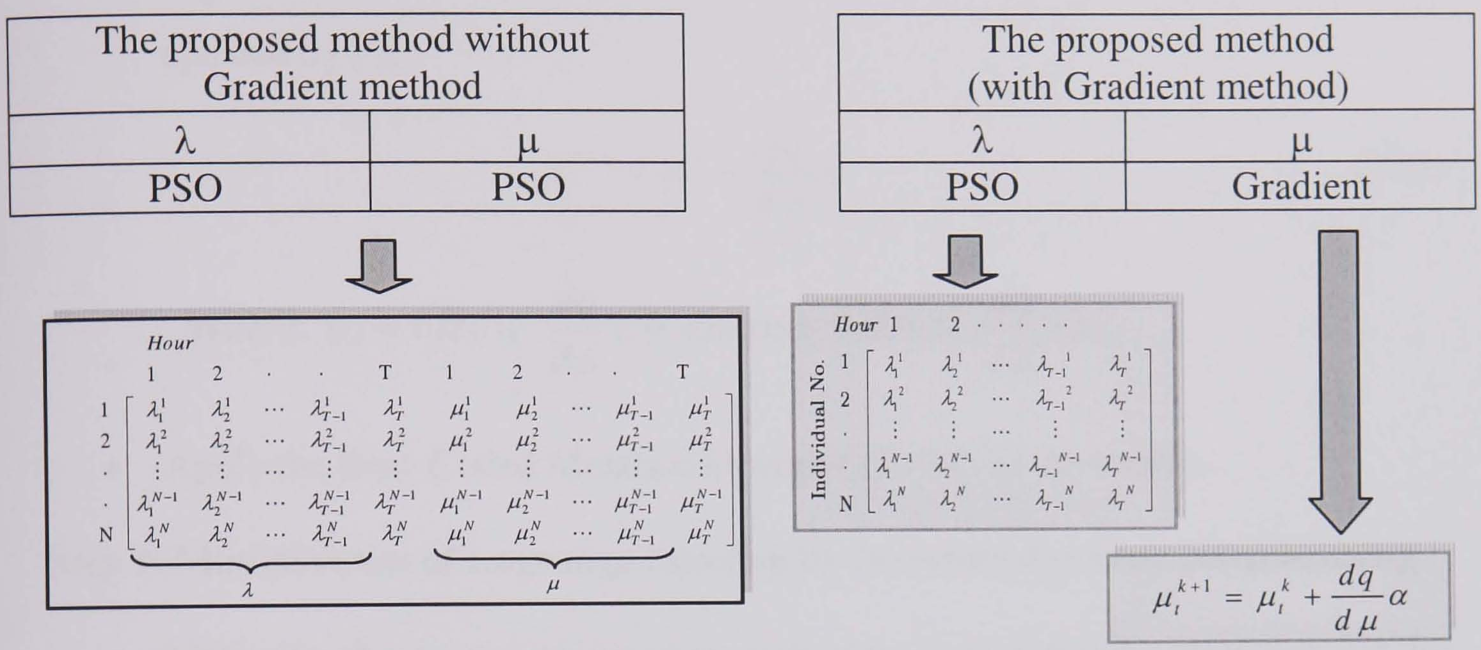


Figure 5.7 Population in the form of a matrix between the proposed method and the proposed method without Gradient method

Step 2: Maximisation of Lagrange function by updating Lagrange multipliers using CBPSO-RVM and Gradient method

- Calculate the evaluation value or dual value (q) of each individual (λ_t, μ_t) by (5.31).

$$
\begin{aligned}
q(\lambda,\mu) = \sum_{i=1}^{N}\sum_{t=1}^{T}\{(1-r)F(P_{it}) + r\cdot F(P_{it}+R_{it}) + ST_t \\
-\left(P_{it}\cdot\rho_{SP_t} + \left((1-r)\cdot\rho_{RP_t} + r\cdot\rho_{SP_t}\right)R_{it}\right)U_{it} \\
-\sum_{t=1}^{T}\lambda_t(P_{Dt} - \sum_{i=1}^{N}P_{it}U_{it}) - \sum_{t=1}^{T}\mu_t(SR_t - \sum_{i=1}^{N}R_{it}U_{it})
\end{aligned}
\tag{5.31}
$$

- Compare each evaluation value with the previous *pbest*. If the current value is more, let it be *pbest*. Similarly, if the best value in *pbest*'s group is more than *gbest*, let the value be *gbest*.
- Update the member velocity (v) of each individual (λ_i) by (3.5).

- If $v_{id}^{(t+1)} > V_{d,\max}$, then $v_{id}^{(t+1)} = V_{d,\max}$ or if $v_{id}^{(t+1)} < -V_{d,\max}$, then $v_{id}^{(t+1)} = -V_{d,\max}$. The maximum velocity can be calculated by (5.12).
- Update the member position of each individual (λ_i) from (3.6) while μ_i is updated by [3]:

$$\mu_t^{k+1} = \mu_t^k + \frac{dq}{d\mu}\alpha \tag{5.32}$$

Where: $\alpha_1 = 0.01$ if $\frac{dq}{d\mu} \geq 0$ and $\alpha_2 = 0.001$ if $\frac{dq}{d\mu} < 0$.

- Apply the Real-Coded Mutation Operator (RVM) to the swarm.

Step 3: Minimisation of Lagrange function by two-state dynamic programming

- Minimise the Lagrange function using two-state dynamic programming for P_{it}, R_{it} and U_{it}, where $i = 1...N$, and $t = 1...T$.

Step 4: Calculation of the dual value, the primal value, and the duality gap

- Determine the dual value (q) from (5.31) using P_{it}, R_{it} and U_{it} obtained from Step 3.
- Calculate P_{it}^* and R_{it}^* by using the Economic Dispatch. However, the ED problem here called *the profit-based ED problem* which is different from the traditional ED problem in which it is aimed at maximising expected profit instead of minimising cost. Therefore, the objective function of the profit-based ED can be expressed by:

$$\textit{Maximise} \sum_{i=1}^{N} \left(\textit{Expected Revenue}_i - \textit{Cost}_i \right) \tag{5.33}$$

subject to the various constraints (5.21)-(5.25). *Expected Revenue* and *Cost* of unit i can be calculated from the following equation:

$$Expected\ Revenue_i = \left(P_i^* \cdot \rho_{SP} + \left((1-r)\cdot \rho_{RP} + r\cdot \rho_{SP}\right)R_i^*\right) \quad (5.34)$$

$$Cost_i = \left((1-r)F(P_i^*) + r\cdot F(P^*_i + R_i^*)\right) \quad (5.35)$$

- Calculate primal value (J) by substituting obtained P_{it}^* and R_{it}^* into the following equation:

$$J = \sum_{i=1}^{N}\sum_{t=1}^{T}\left(Cost^*_{it} - Expected\ Revenue^*_{it}\right) \quad (5.36)$$

where it can be rewritten as:

$$J = \sum_{i=1}^{N}\sum_{t=1}^{T}\left((1-r)F(P_{it}^*) + r\cdot F(P_{it}^* + R_{it}^*) + ST_i\right)\cdot U_{it} \\ -\left(P_{it}^*\cdot \rho_{SP_t} + \left((1-r)\cdot \rho_{RP_t} + r\cdot \rho_{SP_t}\right)R_{it}^*\right)\cdot U_{it}. \quad (5.37)$$

- Determine the duality gap (ε) from (5.19).

Step 5: If either the predefined maximum number of generations is reached, or the duality gap is less than a setting threshold, then stop. The latest P_{it}^* and R_{it}^* is the optimal solution. Otherwise, return to Step 2.

5.3.3 Simulation Results

In order to validate the efficiency of the proposed method, two different case studies, a 3-unit 12-hour system and a 10-unit 24-hour system, are considered. The simulation data used in both cases are adopted from [106] and the details are given in Appendix B. The parameters for experiments in both cases are set to: initial inertia weight (w_{max}) = 0.9; final inertia weight (w_{min}) = 0.4; acceleration constants (c_1,c_2) = 2.05, constriction factor (k) = 0.73, number of particles in mutation (N_m) = 1, and the estimated probability of calling spinning reserve (r) = 0.005, respectively.

Case A: 3-unit, 12-hour system

For simplicity, the simulations in this case are divided into two groups. Firstly, the traditional UC using LR method and the profit-based UC using the proposed method are taken into consideration. Secondly, the proposed method with and without gradient method will be compared.

A1. Comparison between the traditional UC and the profit-based UC in terms of quality solution

The simulation parameters of the traditional UC using LR method are $\alpha_1 = 0.01$ and $\alpha_2 = 0.001$ and the parameters used for the profit-based UC using the proposed method with out gradient method are: the population size = 20, the maximum numbers of generations = 18. Tables 5.6 and 5.7 illustrate the best solution obtained from the traditional UC using LR method and the profit-based UC using the proposed method. It can be seen from the simulation results that the profit-based UC yields $9136 of the expected profit which is the same as reported in [106], whilst the traditional UC provides $4262.71. It can be therefore concluded that the profit-based UC can yield better expected profit than the traditional UC.

Table 5.6 The best solution obtained from the traditional UC using LR method for Case A.

Hour	Load (MW)	Reserve (MW)	Power (MW)			Reserve (MW)			Revenue ($)	Cost ($)	Expected Profit ($)
			U1	U2	U3	U 1	U2	U3			
1	170	20	0	100	70	0	0	20	1802.95	1670.18	132.77
2	250	25	0	100	150	0	0	25	2599.09	2238.45	360.64
3	400	40	0	200	200	0	40	0	3616.13	3501.82	114.31
4	520	55	0	320	200	0	55	0	4937.28	4618.68	318.61
5	700	70	100	400	200	70	0	0	7031.36	7373.69	-342.33
6	1050	95	450	400	200	95	0	0	11860.38	10810.70	1049.68
7	1100	100	500	400	200	100	0	0	12480.62	11406.10	1074.52
8	800	80	200	400	200	80	0	0	8558.17	7984.38	573.79
9	650	65	100	350	200	15	50	0	6757.64	6429.50	328.14
10	330	35	100	100	130	0	0	35	3713.56	3610.81	102.75
11	400	40	100	100	200	0	40	0	4319.26	4146.72	172.54
12	550	55	100	250	200	0	55	0	5856.12	5478.83	377.29
Total											4262.71

Table 5.7 The best solution obtained from the profit-based UC using the proposed method for Case A

Hour	Load (MW)	Reserve (MW)	Power (MW)			Reserve (MW)			Revenue ($)	Cost ($)	Expected Profit ($)
			U1	U2	U3	U 1	U2	U3			
1	170	20	0	0	170	0	0	20	1802.95	1265.28	537.67
2	250	25	0	0	200	0	0	0	2070.00	1500.00	570.00
3	400	40	0	0	200	0	0	0	1800.00	1500.00	300.00
4	520	55	0	0	200	0	0	0	1890.00	1500.00	390.00
5	700	70	0	330	200	0	70	0	5331.36	5115.69	215.67
6	1050	95	0	400	200	0	0	0	6750.00	5400.00	1350.00
7	1100	100	0	400	200	0	0	0	6780.00	5400.00	1380.00
8	800	80	0	400	200	0	0	0	6390.00	5400.00	990.00
9	650	65	0	387.2	200	0	12.8	0	6083.46	5273.05	810.41
10	330	35	0	130	200	0	35	0	3713.56	2883.78	829.78
11	400	40	0	200	200	0	40	0	4319.26	3501.82	817.44
12	550	55	0	350	200	0	50	0	5853.74	4908.72	945.03
Total											9136.00

A2. Comparison between the profit-based UC with and without Gradient method in terms of convergence characteristics

In this case, the simulations are carried out in order to examine the capability and the efficiency of the proposed method with and without combining it with a Gradient method. The setting of the parameters between two versions of the proposed method is almost the same except the section of Gradient method. Namely, α_1 and α_2 are set to 0.01 and 0.001, respectively. Figure 5.8 illustrates the average convergence characteristics between the proposed method with and without applying Gradient method over 30 different trial runs. It can be seen that combination with the gradient method improves the convergence capability when compared with the case when proposed method is used without Gradient method.

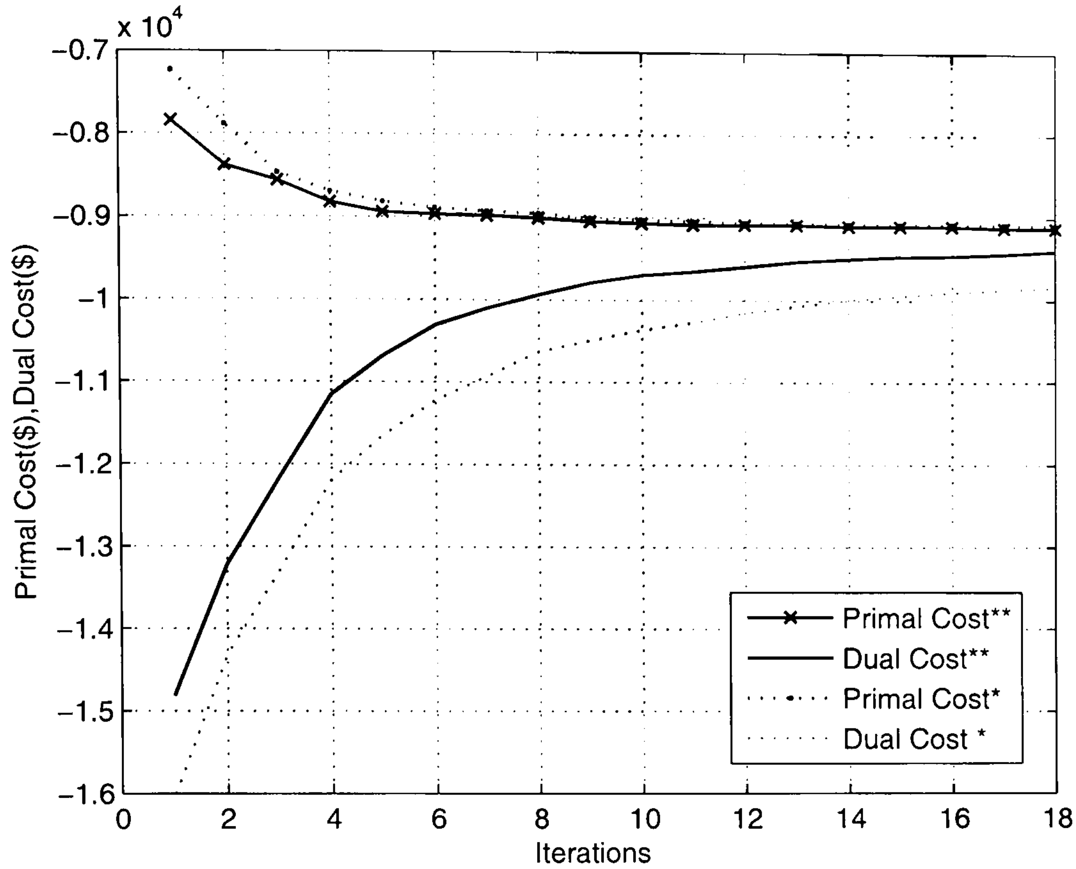


Figure 5.8 Convergence curves between the proposed method with and without Gradient method (* The proposed method without Gradient method, ** The proposed method with Gradient method)

Case B: 10-unit, 24-hour system

The simulation parameters in this case are: the population size = 50, the maximum numbers of generations = 300, respectively. In order to show the reliability of proposed algorithm, a 10-unit 24-hour system [106] is taken into consideration. As shown in [106], the optimum solution is \$107838.57. The best simulation results obtained from the proposed method with applying Gradient method is illustrated in Table 5.8. These results show that the proposed method with Gradient method combination shows a satisfactory performance, although it is applied to a large-scale problem.

Table 5.8 The best solution obtained from the profit-based UC using the proposed method with Gradient method for Case B

Hr	Load (MW)	Reserve (MW)	Power (MW)										Reserve (MW)										Revenue ($)	Cost ($)	Profit ($)
			U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10			
1	700	70	455	245	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0	15528.18	13689.23	1838.95
2	750	75	455	295	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	16524.67	14561.05	1963.62
3	850	85	455	395	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	19655.72	16307.15	3348.57
4	950	95	455	455	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20611.50	17353.30	3258.20
5	1000	100	455	415	0	130	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	23263.90	20076.28	3187.63
6	1100	110	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23868.00	20213.96	3654.04
7	1150	115	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23400.00	20213.96	3186.04
8	1200	120	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23036.00	20213.96	2822.04
9	1300	130	455	455	130	130	130	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	29650.91	29087.36	563.55
10	1400	140	455	455	130	130	162	68	0	0	0	0	0	0	0	0	0	12	0	0	0	0	41095.27	29109.61	11985.65
11	1450	145	455	455	130	130	162	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42571.80	29047.98	13523.82
12	1500	150	455	455	130	130	162	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44689.80	29047.98	15641.82
13	1400	140	455	455	130	130	162	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32767.20	26851.61	5915.59
14	1300	130	455	455	130	130	130	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	31861.72	26187.36	5674.36
15	1200	120	455	455	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26325.00	23105.76	3219.24
16	1050	105	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23192.00	20213.96	2978.04
17	1000	100	455	415	0	130	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	22263.31	19516.28	2747.03
18	1100	110	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22932.00	20213.96	2718.04
19	1200	120	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23088.00	20213.96	2874.04
20	1400	140	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23556.00	20213.96	3342.04
21	1300	130	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24024.00	20213.96	3810.04
22	1100	110	455	455	0	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23868.00	20213.96	3654.04
23	900	90	455	445	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	20478.40	17178.79	3299.61
24	800	80	455	345	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	18066.97	15434.42	2632.55
Total																									107838.57

5.3.4 Summary of PSO application in a Profit-Based Unit Commitment

This section presents a new hybrid method that is the combination of Lagrange Relaxation method (LR) and PSO algorithm with a real-valued natural mutation operation (CBPSO-RVM) to solve a profit-based UC problem. The proposed method not only uses CBPSO-RVM to update the Lagrange multiplier (λ_t) as shown in the section of the traditional UC, but also enhances its performance by adopting the Gradient method to update another Lagrange multiplier (μ_t). To validate the performance of the proposed method with Gradient method, both 3-unit 12-hour and 10-unit 24-hour systems are considered. Moreover, the simulation results are also compared to LR and LR-EP [106] as well. From the simulation results, it can be concluded that the proposed method with Gradient method can obtain the optimum solution even in the large-scale system.

Chapter 6: Sensitivity Analysis of PSO Parameters in ED and UC Problems

6.1 Introduction

This chapter aims at studying the influence of different parameters setting on ED and UC problems. Over the years, there are several pieces of research that investigate the sensitivity analysis of PSO parameters as shown in [13, 16, 41, 52, 64, 66, 84, 113, 114]. These can be classified into two main categories: (1) sensitivity analysis in the mathematical problems [41, 84, 113, 114], and (2) sensitivity analysis in the real-world problems [13, 16, 52, 64, 66]. Regarding these two categories, there are reports on some testing on the values, which the best solution is obtained for the particular problem and applied method. On the other hand, this study focuses directly on sensitivity analyses that investigate the effect of parameters variation on ED and UC problems.

6.2 Sensitivity Analysis of PSO Parameters in ED Problem

To study the effect of varying PSO parameters on ED problem, a 3-unit system with non-smooth cost functions has been investigated by using the six PSO algorithms: BPSO, CPSO, CBPSO, BPSO-RVM, CPSO-RVM, and CBPSO-RVM, respectively. The data of the system are adopted from [95] and the details are shown in Appendix A.

The investigations are divided into two groups according to the parameter settings. The first group focuses on sensitivity of the population size, while the second considers inertia weight factor (w) and acceleration constants (c_1,c_2). Furthermore, each group will be compared and discussed from four aspects: (1) frequency of convergence to global solution, (2) average cost, (3) average computation time, and (4) standard deviation, respectively. Due to the randomness of the simulation results, each point on the following graphs is obtained from an average of final results over 100 different runs.

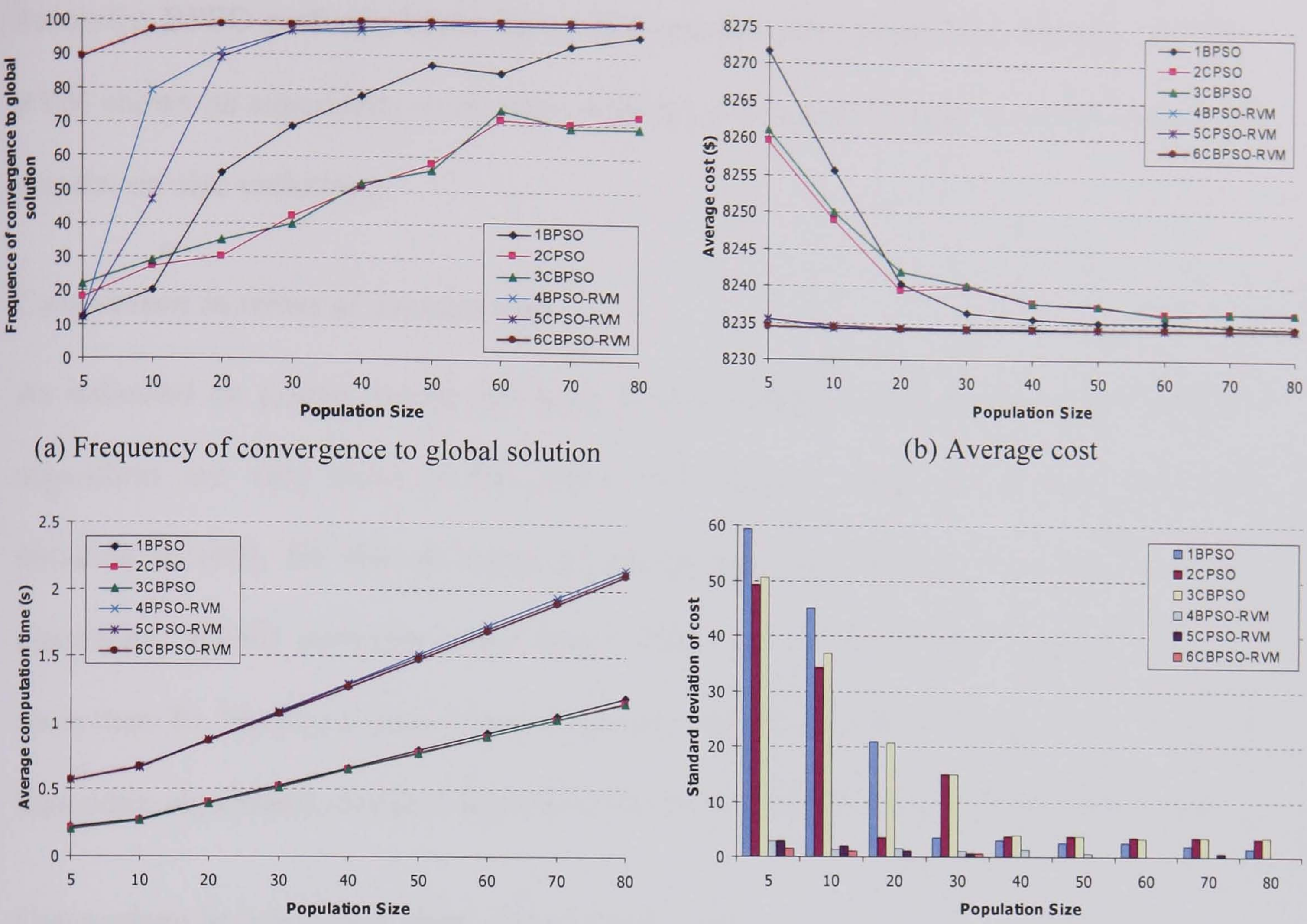
6.2.1 Sensitivity of the population size

In this section, the population sizes are varied from 5 to 80 whilst other parameters are the same as used in Chapter 4. Table 6.1 shows the parameters used in this simulation. Figure 6.1(a)-(d) illustrates the comparison of the simulation results amongst the six PSO algorithms according to four patterns as mentioned before.

Table 6.1 Parameters used in the implementation of the six algorithms for ED problem

Methods	c_1/ c_2	φ	K	w_{max}	w_{min}	N_m	Iter	Pop*
BPSO	2.0	-	1.0	0.9	0.4	-	300	5-80
CPSO	2.05	4.1	0.73	1.0	1.0	-	300	
CBPSO	2.05	4.1	0.73	0.9	0.4	-	300	
BPSO-RVM	2.0	-	1.0	0.9	0.4	1	300	
CPSO-RVM	2.05	4.1	0.73	1.0	1.0	1	300	
CBPSO-RVM	2.05	4.1	0.73	0.9	0.4	1	300	

Note: c_1,c_2 - acceleration constants, φ - summation of c_1 and c_2 , K - constriction factor, $w_{max,min}$ - max/min inertia weight, N_m - number of particles that participate in mutation, Iter - total number of iterations, Pop*- different cases of various population sizes.



(a) Frequency of convergence to global solution

(b) Average cost

(c) Average computation time

(d) Standard deviation of cost

Figure 6.1 Comparison of various population sizes obtained from BPSO, CPSO, CBPSO, BPSO-RVM, CPSO-RVM, and CBPSO-RVM for ED problem

Comparison in terms of frequency of convergence to global solution

For simplification, the discussions will be divided into three parts: (1) the hybrid PSO algorithms (BPSO-RVM, CPSO-RVM, and CBPSO-RVM) against the traditional PSO algorithms (BPSO, CPSO and CBPSO); (2) the traditional PSO algorithms themselves; and (3) the hybrid PSO algorithms themselves.

Firstly, results indicate that the hybrid PSO algorithms can achieve higher rate of the frequency of convergence than the conventional PSO algorithms. Moreover, the hybrid PSO algorithms respond well to the different population sizes. In all comparisons, 70% of frequency of convergence to global solution has been used as the criterion to guarantee the reliable solutions, as recommended by Victoire and Jeyakumar [13].

Secondly, BPSO performs better than other standard PSO algorithms. Finally, CBPSO-RVM shows its superiority to all PSO algorithms that follow from the best response to population size variations.

Comparison in terms of average cost

As indicated on graphs shown in Figure 6.1(b) solutions obtained by the hybrid PSO algorithms are very close to the global best solution ($8234.07), which has been obtained in [96], for the all range of population size. Among the traditional PSO algorithms, BPSO performs better than CPSO and CBPSO when population size is more than 30. Finally, Figure 6.1(b) illustrates that CBPSO-RVM is still more desirable than other algorithms, because it gives solutions that are the closes to the optimal ones.

Comparison in terms of average computation time

From Figure 6.1(c), the simulation results can be separated into two groups between the group of traditional PSO and the group of hybrid PSO. It can be seen that for the group of traditional PSO computation times are roughly two times smaller than for the hybrid PSO group.

Comparison in terms of standard deviation

It can be observed from Figure 6.1(d) that the hybrid PSO algorithms yield less standard deviation than the traditional PSO algorithms over the variation of the population sizes. Furthermore, the standard deviations of the hybrid PSO algorithms are almost zero when the population size is more than 30.

From the above comparisons, it can be clearly summarised that the conventional PSO algorithms are more sensitive to the population size variations than the hybrid PSO algorithms. In addition, the outcomes of the proposed method (CBPSO-RVM) show the superiority over other PSO algorithms. In this study, CBPSO-RVM can find the optimal solution with the probability of around 90% even with the population size is 5, however, larger population size gives better reliability of the solution but with longer computational times. Thus, both of these characteristics have to be considered when deciding on a population size, and based on results shown in Figure 6.1 the recommendation of this work is that the optimal population size of CBPSO-RVM should be larger than 20.

6.2.2 Sensitivity of the inertia weight factor (w) and the acceleration constants (c_1,c_2)

In this section, the variation of inertia weight factor (w) is considered for 5 different combinations of values of acceleration constants (c_1,c_2), as shown in Table 6.2. In addition, all simulations for BPSO-RVM are carried out for three values of particles that participate in mutations, N_m.

A. Comparison of simulation results between BPSO and BPSO-RVM

In order to highlight the impact of the variations of w on ED problem, w will be varied from 0.2 to 1.2. This is a larger interval when compared to commonly used range of w between 0.4 and 0.9, that has been investigated in [6, 17, 83, 84, 86]. As recommended by Ting *et al* [66], the variation of w should be related to the pair of c_1 and c_2, namely the summation of c_1 and c_2 is usually set as 4. The parameter settings of BPSO and BPSO-RVM used in these analyses are given in Table 6.2.

Table 6.2 Parameters used in the implementation of BPSO and BPSO-RVM for ED problem

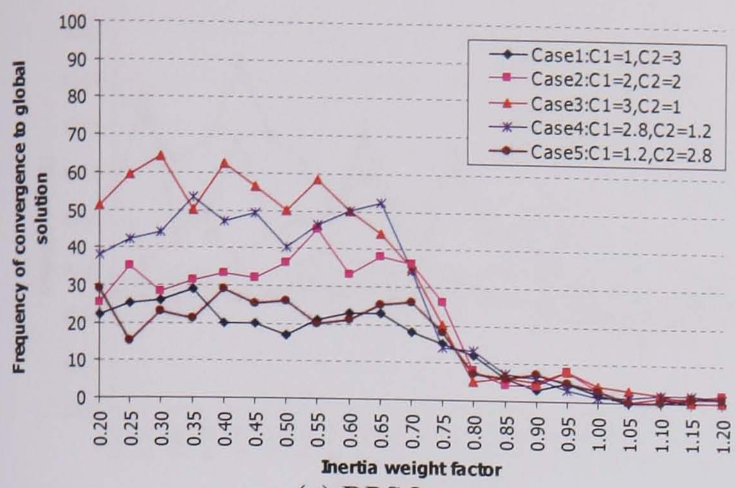
Method	Case	c_1	c_2	φ	K	w*	N_m**	Pop	Iter
BPSO	1	1	3	-	1	0.2-1.2	-	20	300
	2	2	2	-	1				
	3	3	1	-	1				
	4	2.8	1.2	-	1				
	5	1.2	2.8	-	1				
BPSO-RVM	1	1	3	-	1	0.2-1.2	1-3	20	300
	2	2	2	-	1				
	3	3	1	-	1				
	4	2.8	1.2	-	1				
	5	1.2	2.8	-	1				

* The different sub-cases of w that start from 0.2 to 1.2 with increment of 0.05.
** The different sub-cases of the number of particles that participate in mutation.

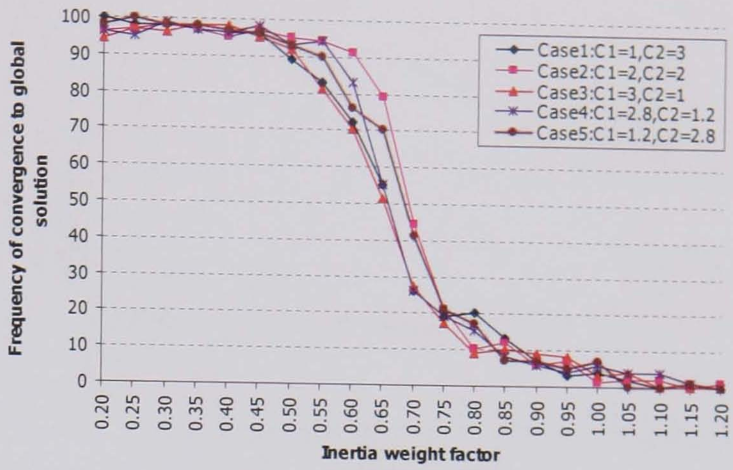
A.1 Comparison in terms of frequency of convergence to global solution

In this section, the discussions will also be made according to three different aspects. Firstly, the comparison between BPSO and BPSO-RVM will be discussed. Then, the comparison of the each of the BPSO itself, and the BPSO-RVM itself will be presented.

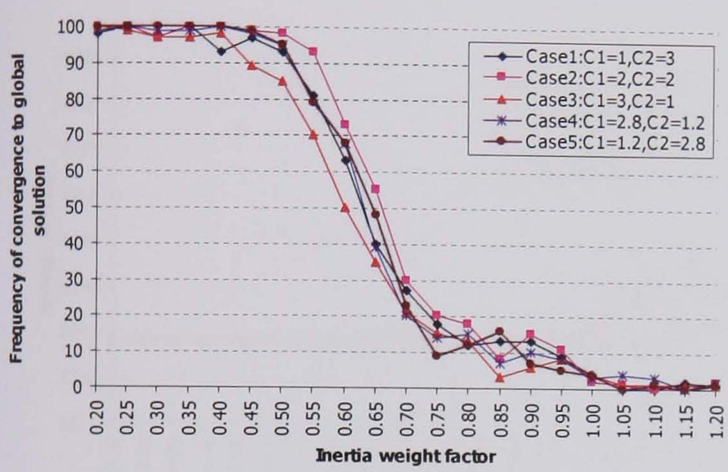
The comparison of frequency of convergence curves between BPSO and BPSO-RVM is demonstrated in Figure 6.2. It indicates that the change in all Cases 1-5 the variation of parameter w provides a better frequency of convergence for BPSO-RVM (Figure 6.2 (b)-(d)) than for BPSO (Figure 6.2(a)). In addition, although amongst BPSO case studies, Case 3, when c_1=3 and c_2=1, seems to be the best, its performance is still lower than the predefined criterion (70% of frequency of convergence [13]) . Lastly, it can be observed for all case studies BPSO-RVM satisfies the predefined criterion. Moreover, it is quite interesting to see the response of inertia weight factor to the variation of particles in mutation (N_m). Namely, increasing of the number of particles in mutation will lead to decreasing in the response to inertia weight factor. For example, BPSO-RVM (N_m = 1, 2 and 3) can respond to the maximum of inertia weight factor up to 0.65, 0.60, and 0.55, respectively.



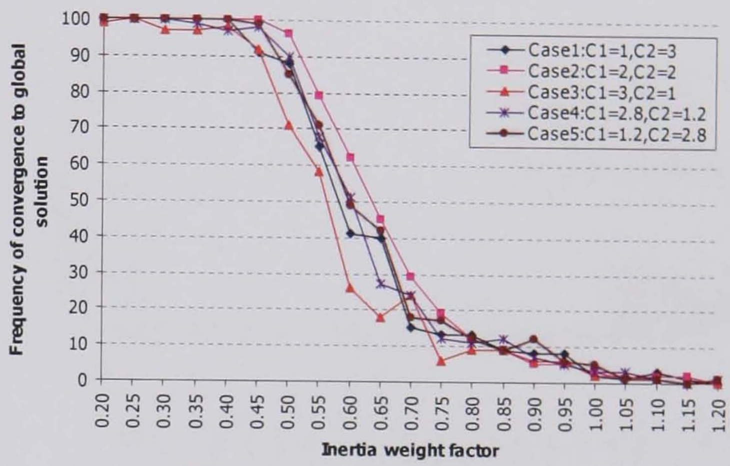
(a) BPSO



(b) BPSO-RVM ($N_m = 1$)



(c) BPSO-RVM ($N_m = 2$)

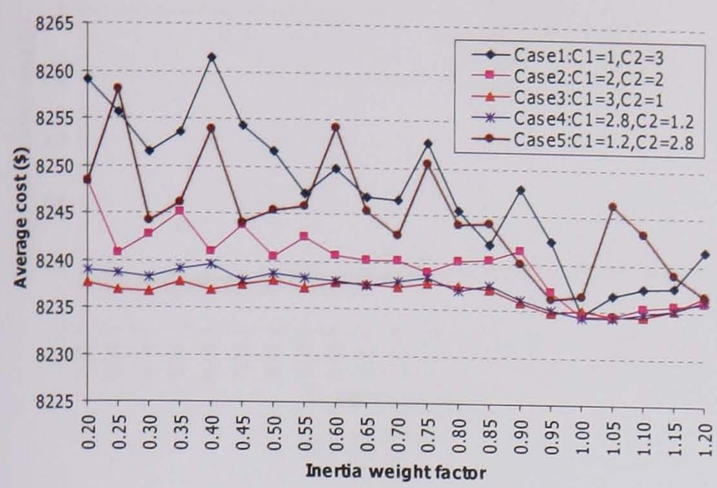


(d) BPSO-RVM ($N_m = 3$)

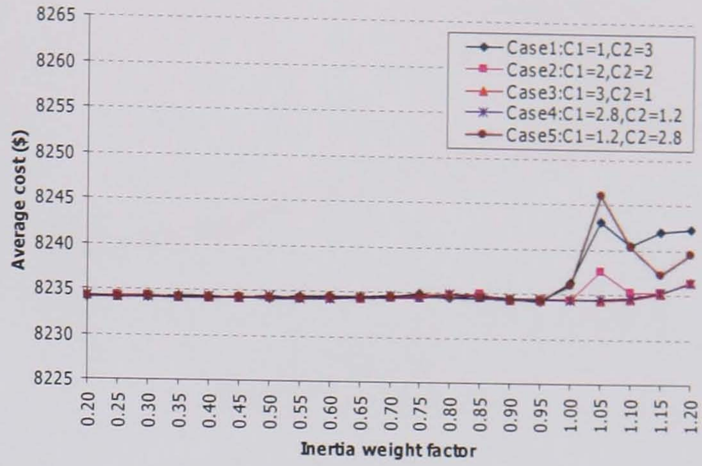
Figure 6.2 Comparison of frequency of convergence to global solution between BPSO and BPSO-RVM for ED problem

A.2 Comparison in terms of average cost

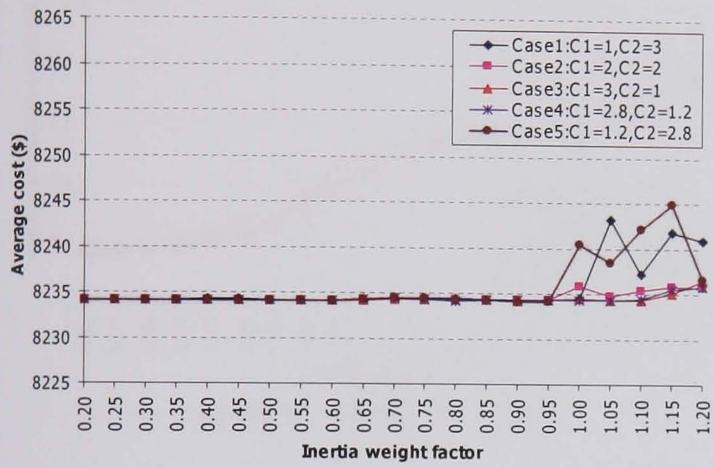
Figure 6.3 shows the comparison of average cost between BPSO and BPSO-RVM. From these graphs we can conclude that BPSO-RVM provides better average cost in all cases than its counterpart. For BPSO itself, Case3 (c_1=3 and c_2=1) yields better results than other cases. For BPSO-RVM itself, results show the superiority for the range of *w* from 0.2 to 0.95 in all 5 cases, and for various number of parameters that are mutated, i.e. N_m=1, 2 and 3.



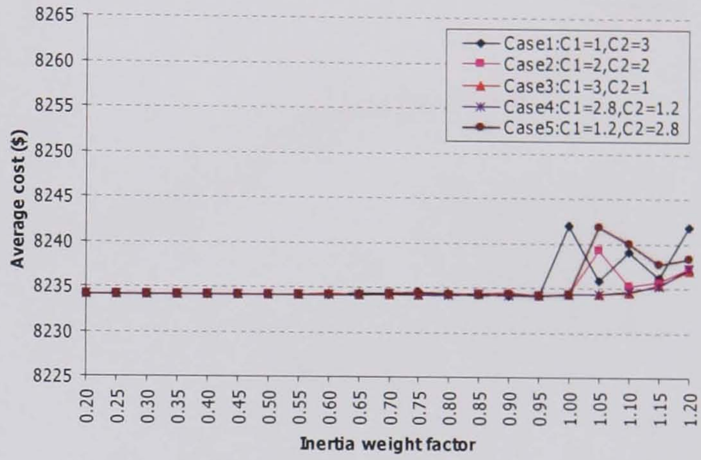
(a) BPSO



(b) BPSO-RVM ($N_m = 1$)



(c) BPSO-RVM ($N_m = 2$)

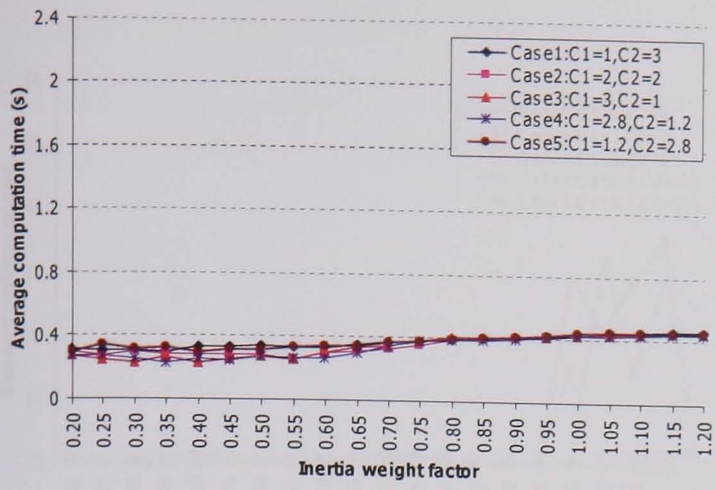


(d) BPSO-RVM ($N_m = 3$)

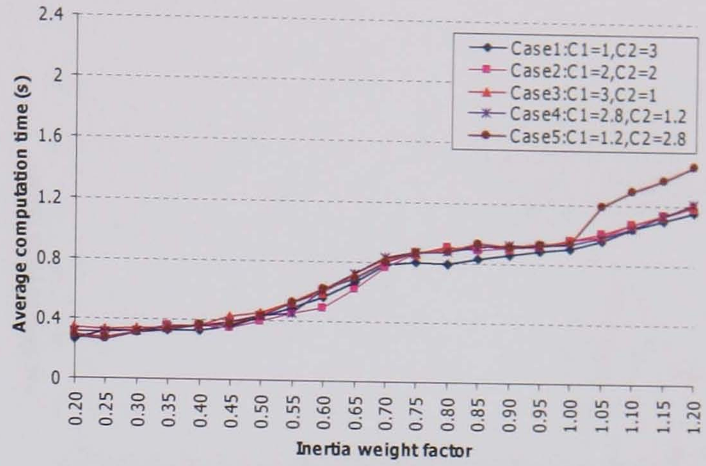
Figure 6.3 Comparison of average cost between BPSO and BPSO-RVM for ED problem

A.3 Comparison in terms of average computation time

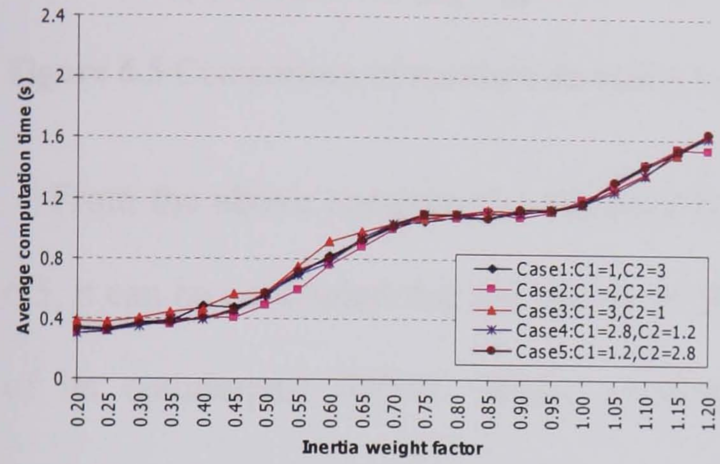
The comparison of average computation time between BPSO and BPSO-RVM is illustrated in Figure 6.4. The graph shows that BPSO takes less computation time than BPSO-RVM. Amongst BPSO-RVM itself, one can observe that an increase in the calculation time is due to changes in the number of particles in mutation. In addition, the significant changes in frequency of convergence (Figure 6.2 (b)-(d)) are related to a gradual increase in computation times (Figure 6.4 (b)-(d)).



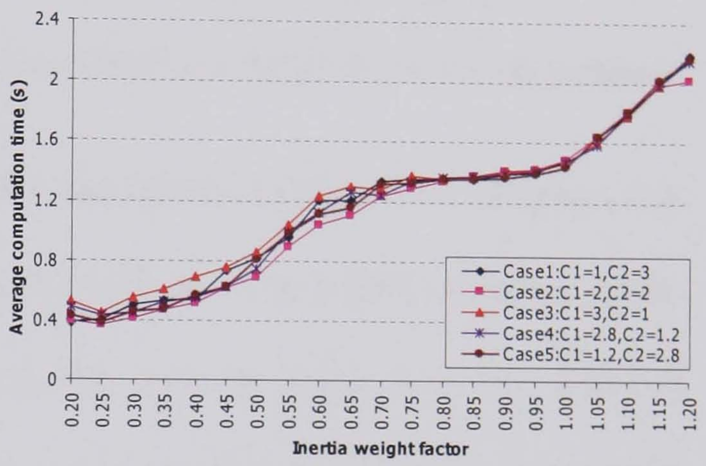
(a) BPSO



(b) BPSO-RVM (N_m = 1)



(c) BPSO-RVM (N_m = 2)

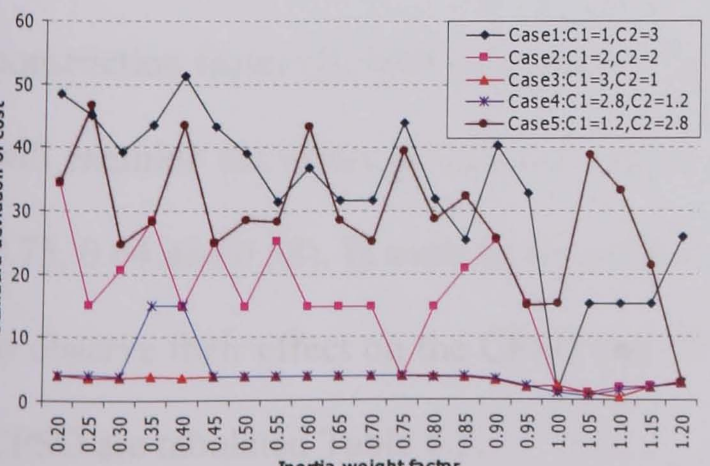


(d) BPSO-RVM (N_m = 3)

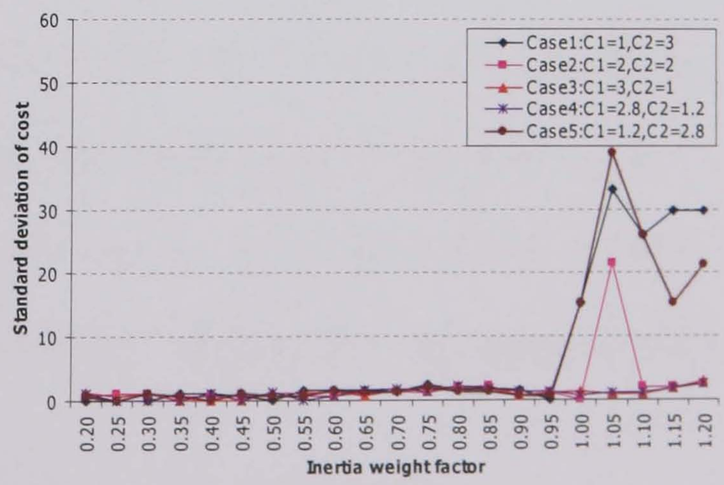
Figure 6.4 Comparison of average computation time between BPSO and BPSO-RVM for ED problem

A.4 Comparison in terms of standard deviation

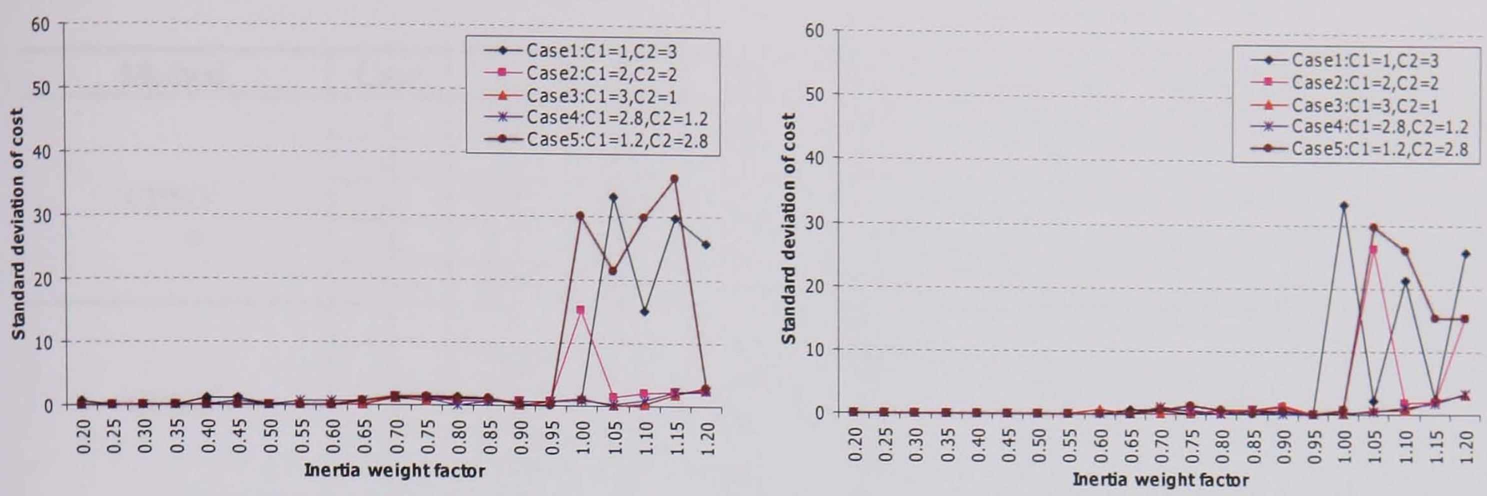
The plots that indicate the standard deviation between BPSO and BPSO-RVM are shown in Figure 6.5. It can be seen that the standard deviation of BPSO-RVM (Figure 6.5 (b)-(d)) is almost zero in the range of w = 0.2 to 0.95; in contrast, BPSO (Figure 6.5 (a)) suffers from the fluctuation problem, except from Case 3 and Case 4.



(a) BPSO



(b) BPSO-RVM (N_m = 1)



(c) BPSO-RVM (N_m = 2)

(d) BPSO-RVM (N_m = 3)

Figure 6.5 Comparison of standard deviation between BPSO and BPSO-RVM for ED problem

From the above comparisons between BPSO and BPSO-RVM given in Figures 6.2-6.5, it can be concluded that the results of BPSO-RVM are more reliable than the results of its counterpart BPSO. When considering the average costs and the standard deviations, the range of varied inertia weight factor (w) around 0.2-0.95 seems to be reliable, however the range of w between 0.2-0.6 is more preferable when the success rate (70% of frequency of convergence, discussed in section 6.2.2.A.1 and shown in Figure 6.2) is taken into consideration.

For the acceleration constants (c_1,c_2), the optimal case for BPSO-RVM is Case2 when $c_1 = c_2 = 2$.

B. Comparison of simulation results between CPSO and CPSO-RVM

For CPSO algorithm, which has been originally introduced by Clerc [31, 32, 86], constriction factor (K) will be taken into account rather than w. Hence, this experiment will examine the effect of varied c_1 and c_2 making three different values of K (i.e. K = 0.73, 0.64 and 0.58). In each K, five different patterns of c_1 and c_2 are assigned in order to observe their effect on the CPSO and CPSO-RVM. The setting of parameters of the CPSO are tabulated Table 6.3.

Table 6.3 Parameters used in the implementation of CPSO for ED problem

Method	Case	c_1	c_2	φ	K	w	N_m	Pop	Iter
CPSO[1]	1	1.95	2.15	4.1	0.73	-	-	20	300
	2	2	2.1	4.1	0.73				
	3	2.05	2.05	4.1	0.73				
	4	2.1	2	4.1	0.73				
	5	2.15	1.95	4.1	0.73				
CPSO[2]	1	2	2.2	4.2	0.64	-	-	20	300
	2	2.05	2.15	4.2	0.64				
	3	2.1	2.1	4.2	0.64				
	4	2.15	2.05	4.2	0.64				
	5	2.2	2	4.2	0.64				
CPSO[3]	1	2.05	2.25	4.3	0.58	-	-	20	300
	2	2.1	2.2	4.3	0.58				
	3	2.15	2.15	4.3	0.58				
	4	2.2	2.1	4.3	0.58				
	5	2.25	2.05	4.3	0.58				

[1-3] The different sub-cases of the variation of c_1, c_2, φ and K.

The parameters used for CPSO-RVM are somewhat similar to CPSO, except for the number of particles in mutation (N_m), which are varied from 1 to 3. Table 6.4 shows the setting of parameters used in the CPSO-RVM.

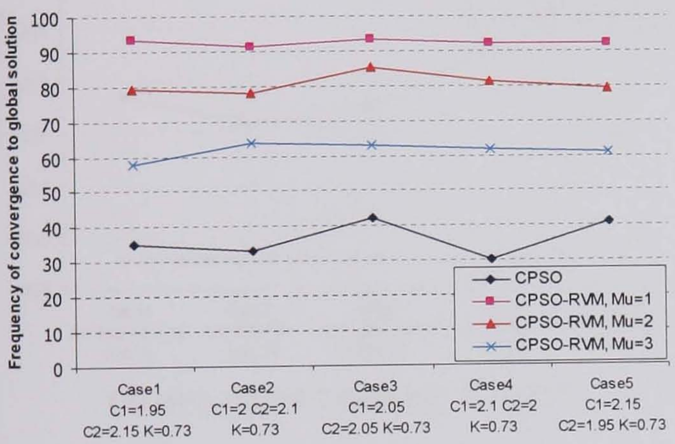
Table 6.4 Parameters used in the implementation of CPSO-RVM for ED problem

Method	Case	c_1	c_2	φ	K	w	N_m*	Pop	Iter
CPSO-RVM[1]	1	1.95	2.15	4.1	0.73	-	1-3	20	300
	2	2	2.1	4.1	0.73				
	3	2.05	2.05	4.1	0.73				
	4	2.1	2	4.1	0.73				
	5	2.15	1.95	4.1	0.73				
CPSO-RVM[2]	1	2	2.2	4.2	0.64	-	1-3	20	300
	2	2.05	2.15	4.2	0.64				
	3	2.1	2.1	4.2	0.64				
	4	2.15	2.05	4.2	0.64				
	5	2.2	2	4.2	0.64				
CPSO-RVM[3]	1	2.05	2.25	4.3	0.58	-	1-3	20	300
	2	2.1	2.2	4.3	0.58				
	3	2.15	2.15	4.3	0.58				
	4	2.2	2.1	4.3	0.58				
	5	2.25	2.05	4.3	0.58				

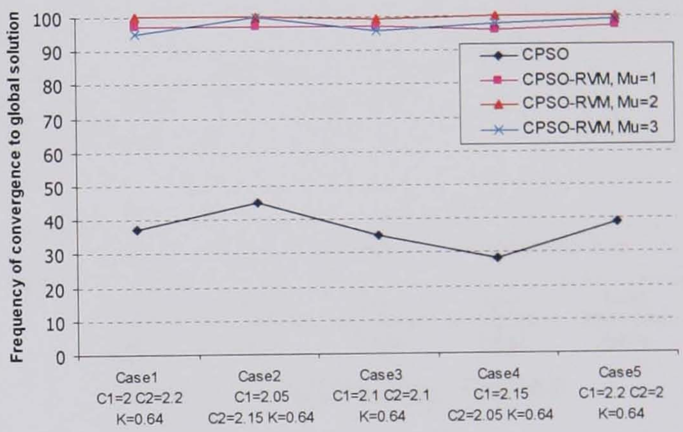
* The different sub-cases of the number of particles that participate in mutation.
[1-3] The different sub-cases of the variation of c_1, c_2, φ and K.

B.1 Comparison in terms of frequency of convergence to global solution

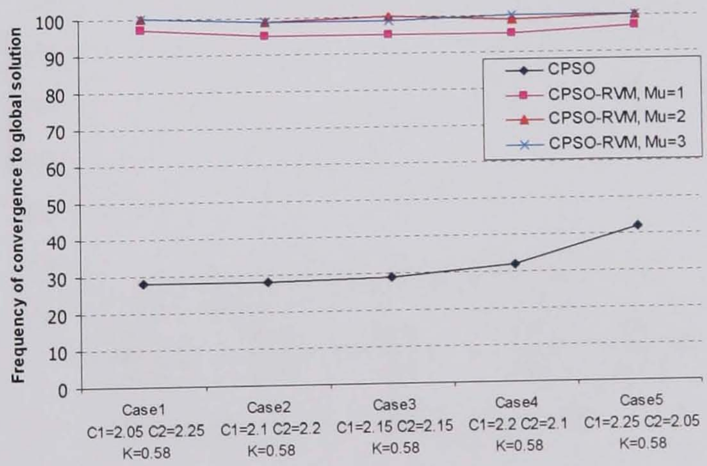
The comparison of frequency of convergence to global solution is shown in Figure 6.6. It shows that CPSO-RVM has higher rate of frequency of convergence to global solution, and therefore it outperforms CPSO for any case of parameter K variation. This is particularly highlighted in the case when K= 0.64 and 0.58. For CPSO itself, CPSO with K= 0.64 performs better than others; nevertheless, its performance is very poor when considering the previously discussed predefined criterion of 70% of frequency of convergence.



(a) $\varphi=c_1+c_2=4.1$ making K=0.73



(b) $\varphi= c_1+c_2=4.2$ making K=0.64



(c) $\varphi= c_1+c_2=4.3$ making K=0.58

Figure 6.6 Comparison of frequency of convergence to global solution between CPSO and CPSO-RVM for ED problem

B.2 Comparison in terms of average cost

Figure 6.7 illustrates the average cost of CPSO and CPSO-RVM versus the different values of K. The trend of CPSO-RVM is somewhat similar to the trend of BPSO-RVM with a change of *w* (Section 6.2.2.A.2), as for both algorithms their average costs are very close to global solution in every case. On the other hand, the average costs of CPSO fluctuate between \$8239 and \$8243 in the first two cases and the fluctuation increases in the last case.

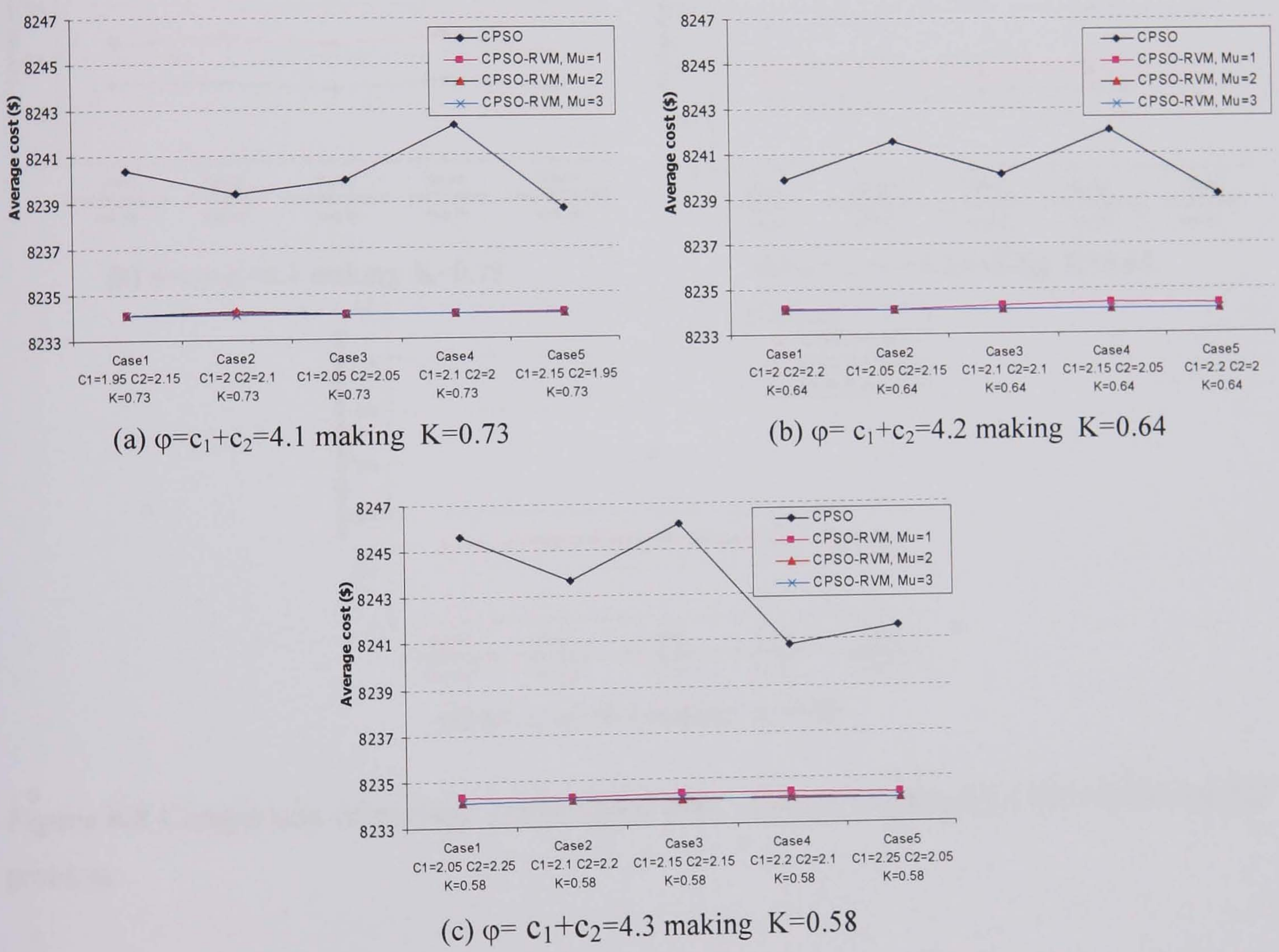
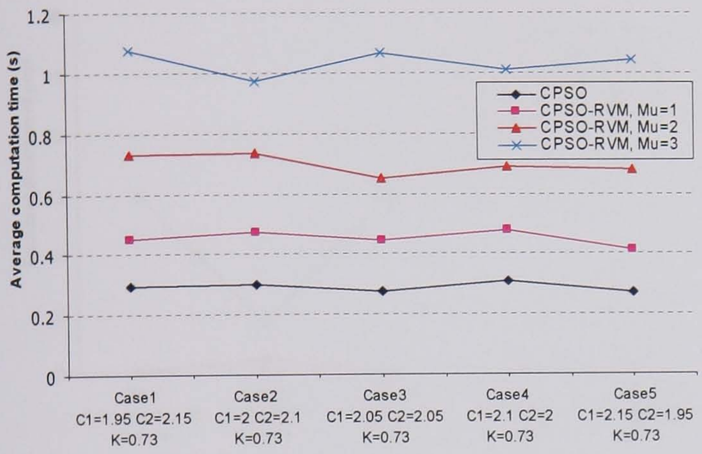


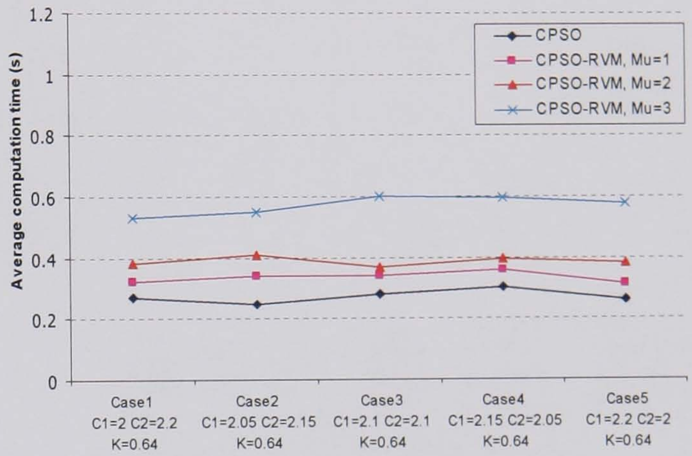
Figure 6.7 Comparison of average cost between CPSO and CPSO-RVM for ED problem

B.3 Comparison in terms of average computation time

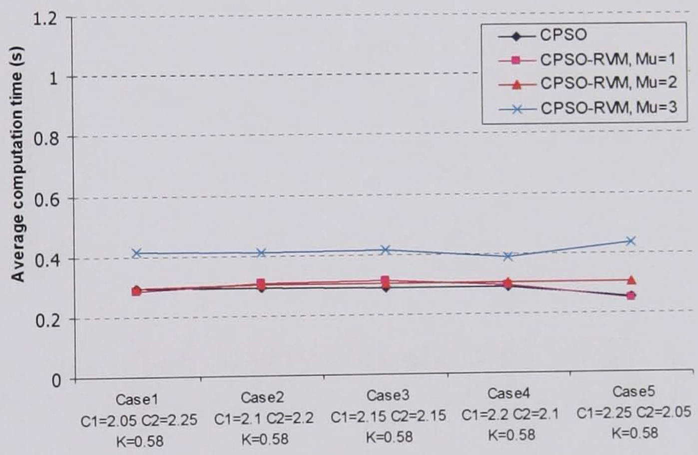
The comparison of average computation time between CPSO and CPSO-RVM is shown in Figure 6.8. The computation time of CPSO in this case is also less than CPSO-RVM. As expected, the rise in number of particles in mutation will contribute to the rise in computation time.



(a) $\varphi=c_1+c_2=4.1$ making K=0.73



(b) $\varphi= c_1+c_2=4.2$ making K=0.64

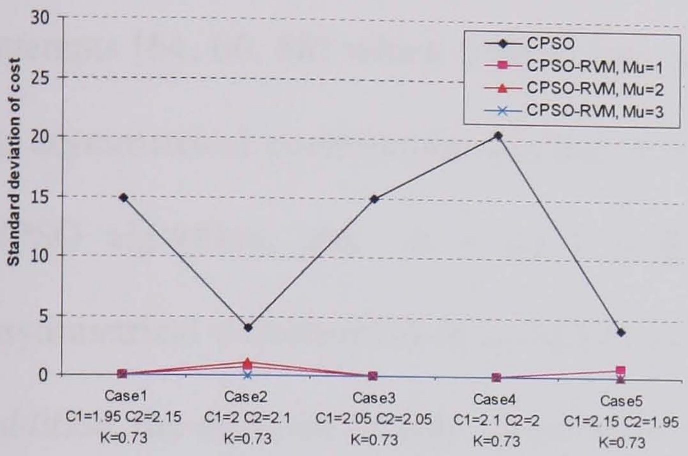


(c) $\varphi= c_1+c_2=4.3$ making K=0.58

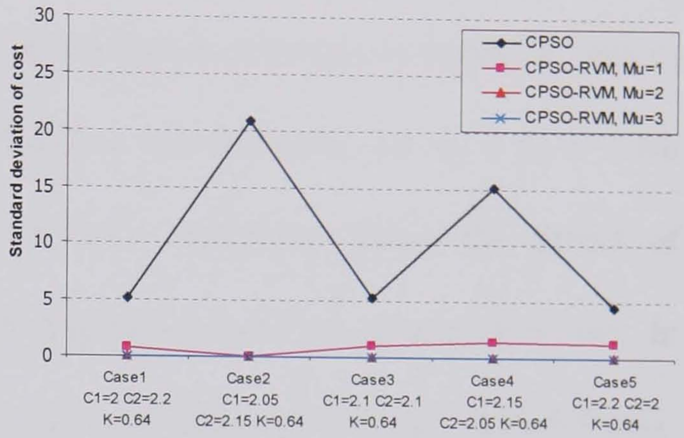
Figure 6.8 Comparison of average computation time between CPSO and CPSO-RVM for ED problem

B.4 Comparison in terms of standard deviation

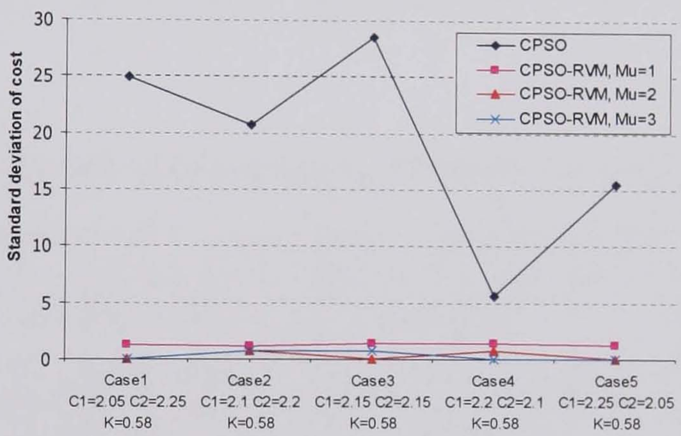
Figure 6.9 shows the comparison of the standard deviation between CPSO and CPSO-RVM. As a result of high quality solution of the average costs in CPSO-RVM, its standard deviations in all cases are considerably lower than CPSO. As for the CPSO, the fluctuations of its standard deviations result from the oscillation in its average costs.



(a) $\varphi = c_1+c_2=4.1$ making K=0.73



(b) $\varphi = c_1+c_2=4.2$ making K=0.64



(c) $\varphi = c_1+c_2=4.3$ making K=0.58

Figure 6.9 Comparison of standard deviation between CPSO and CPSO-RVM for ED problem

From the above comparisons between CPSO and CPSO-RVM, it is observed that that CPSO-RVM is rather less sensitive to the variations of c_1, c_2 and K for all cases compared with its counterpart. Concerning the reliability of the solutions, $c_1+c_2 = 4.2$ making K = 0.64 and $c_1+c_2 = 4.3$ making K = 0.58 seem to be the reasonable parameters setting.

C. Comparison of simulation results between CBPSO and CBPSO-RVM

Since CBPSO is the combination of the original PSO with both inertia weight (w) and constriction factor (K), this study will take the variation of both w and K into consideration. The summation of c_1 and c_2 is generally set to 4.1 when constriction method is employed as presented in [19, 23, 83, 86, 88]. However, there were many attempts [64, 66, 88] which tried to fine tune the optimum solution by paring c_1 and c_2 in asymmetrical combination instead of symmetrical combination, i.e. $c_1 = c_2 = 2$ for BPSO algorithm, and $c_1 = ,c_2 = 2.05$ for CPSO algorithm. Thus, the effect of asymmetrical combination of c_1 and c_2 on both algorithms will be investigated here. In addition, the different number of particles in mutation will be considered. Table 6.5 and Table 6.6 show the parameters used in evaluating performance of CBPSO and CBPSO-RVM, respectively.

Table 6.5 Parameters used in the implementation of CBPSO for ED problem

Method	Case	c_1	c_2	φ	K	w*	N_m	Pop	Iter
CBPSO	1	1.95	2.15	4.1	0.73	0.2-1.2	-	20	300
	2	2	2.1	4.1	0.73				
	3	2.05	2.05	4.1	0.73				
	4	2.1	2	4.1	0.73				
	5	2.15	1.95	4.1	0.73				
	6	2	2.2	4.2	0.64	0.2-1.2	-	20	300
	7	2.05	2.15	4.2	0.64				
	8	2.1	2.1	4.2	0.64				
	9	2.15	2.05	4.2	0.64				
	10	2.2	2	4.2	0.64				
	11	2.05	2.25	4.3	0.58	0.2-1.2	-	20	300
	12	2.1	2.2	4.3	0.58				
	13	2.15	2.15	4.3	0.58				
	14	2.2	2.1	4.3	0.58				
	15	2.25	2.05	4.3	0.58				

* The different sub-cases of w that start from 0.2 to 1.2 at 0.05 intervals.

Table 6.6 Parameters used in the implementation of CBPSO-RVM for ED problem

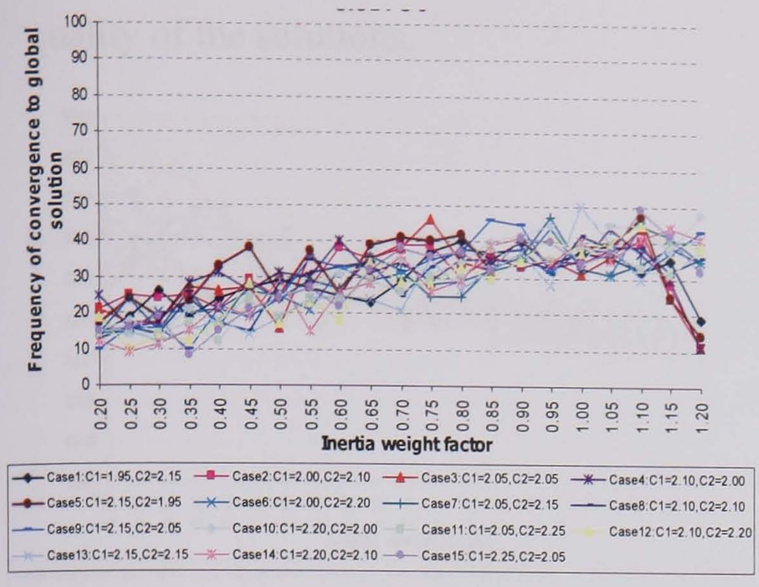
Method	Case	c_1	c_2	φ	K	w*	N_m**	Pop	Iter
CBPSO-RVM	1	1.95	2.15	4.1	0.73	0.2-1.2	1-3	20	300
	2	2	2.1	4.1	0.73				
	3	2.05	2.05	4.1	0.73				
	4	2.1	2	4.1	0.73				
	5	2.15	1.95	4.1	0.73				
	6	2	2.2	4.2	0.64	0.2-1.2	1-3	20	300

	7	2.05	2.15	4.2	0.64				
	8	2.1	2.1	4.2	0.64				
	9	2.15	2.05	4.2	0.64				
	10	2.2	2	4.2	0.64				
	11	2.05	2.25	4.3	0.58				
	12	2.1	2.2	4.3	0.58				
	13	2.15	2.15	4.3	0.58	0.2-1.2	1-3	20	300
	14	2.2	2.1	4.3	0.58				
	15	2.25	2.05	4.3	0.58				

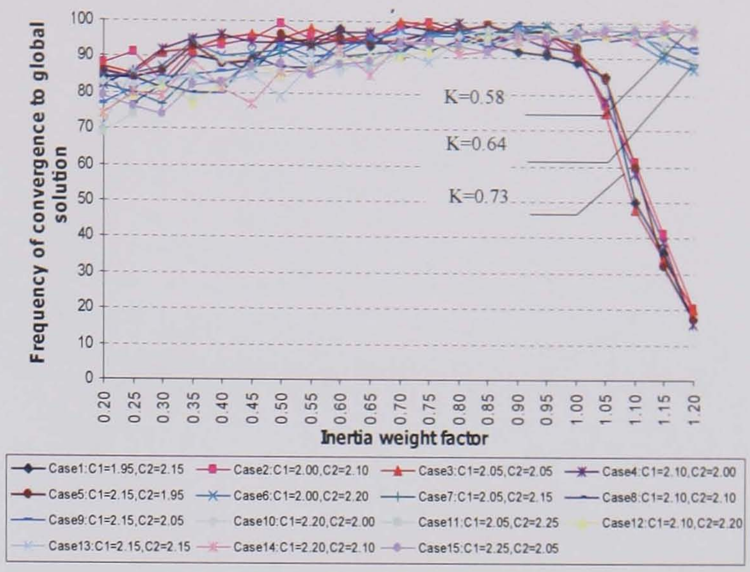
* The different sub-cases of *w* that start from 0.2 to 1.2 at 0.05 intervals.
** The different sub-cases of the number of particles that participate in mutation.

C.1 Comparison in terms of frequency of convergence to global solution

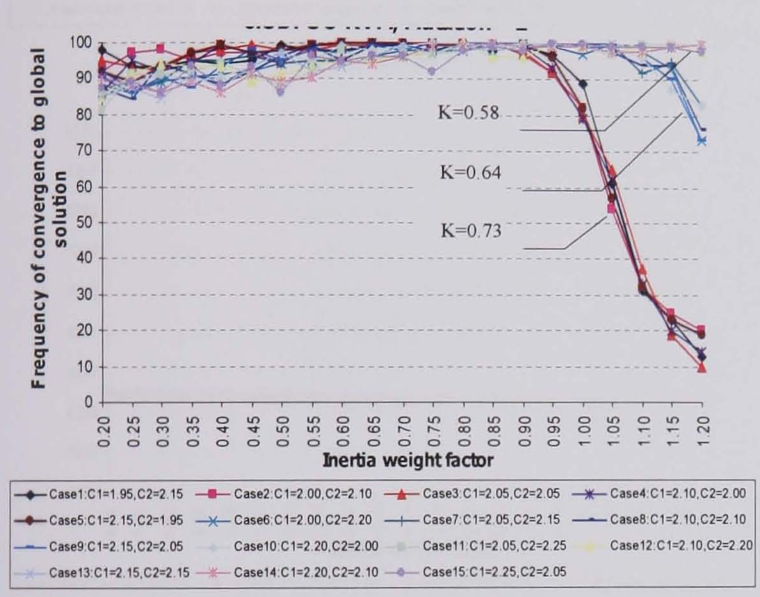
The comparison of frequency of convergence curves between CBPSO and CBPSO-RVM are given in Figure 6.10. Firstly, it can be seen that CBPSO-RVM responds very well to the varied inertia weight compared with CBPSO. In this case, it seems that CBPSO responds to the different values of *w* worse than BPSO and CPSO as shown in Figure 6.2 and Figure 6.6. For CBPSO-RVM itself, its response characteristics can be classified into three separate groups that are: (1) $c_1 + c_2 = 4.1$ or K = 0.73, (2) $c_1 + c_2 = 4.2$ or K = 0.64, and (3) $c_1 + c_2 = 4.3$ or K = 0.58 as illustrated in Figure 6.10 (b)-(d). From Figure 6.10 (b), both groups of K = 0.58 and K = 0.64 give the good response throughout the range of *w*, whilst the group of K = 0.73 does not. Moreover, it can be seen that a decrease in the response of frequency of convergence in Figure 6.10 (b)-(d) follows from the increase in the number of particles in mutation (N_m). Namely, a high inertia weight will cause changes in better global search [84]; in the same way, the large value of N_m will contribute to enhancing the global searching capability too. For the reasons just discussed, CBPSO-RVM will be employed for solving the lack of population diversity in global search, particularly in the case of K = 0.58.



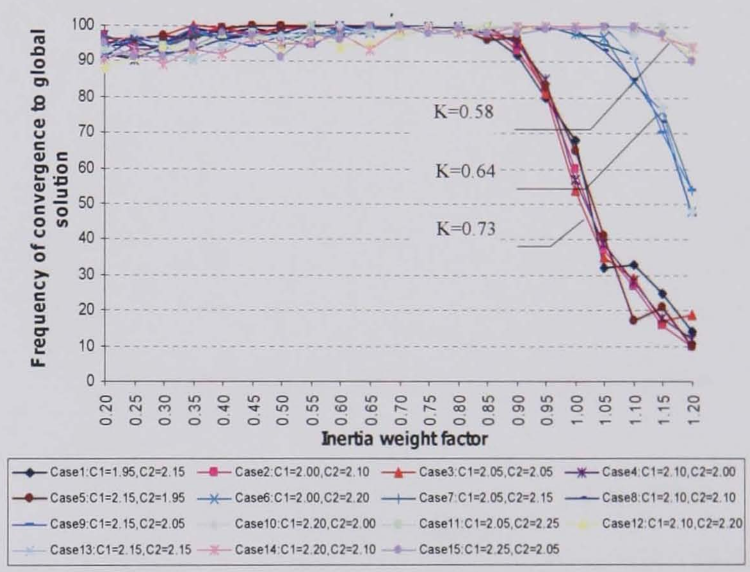
(a) CBPSO



(b) CBPSO-RVM (N_m = 1)



(c) CBPSO-RVM (N_m = 2)

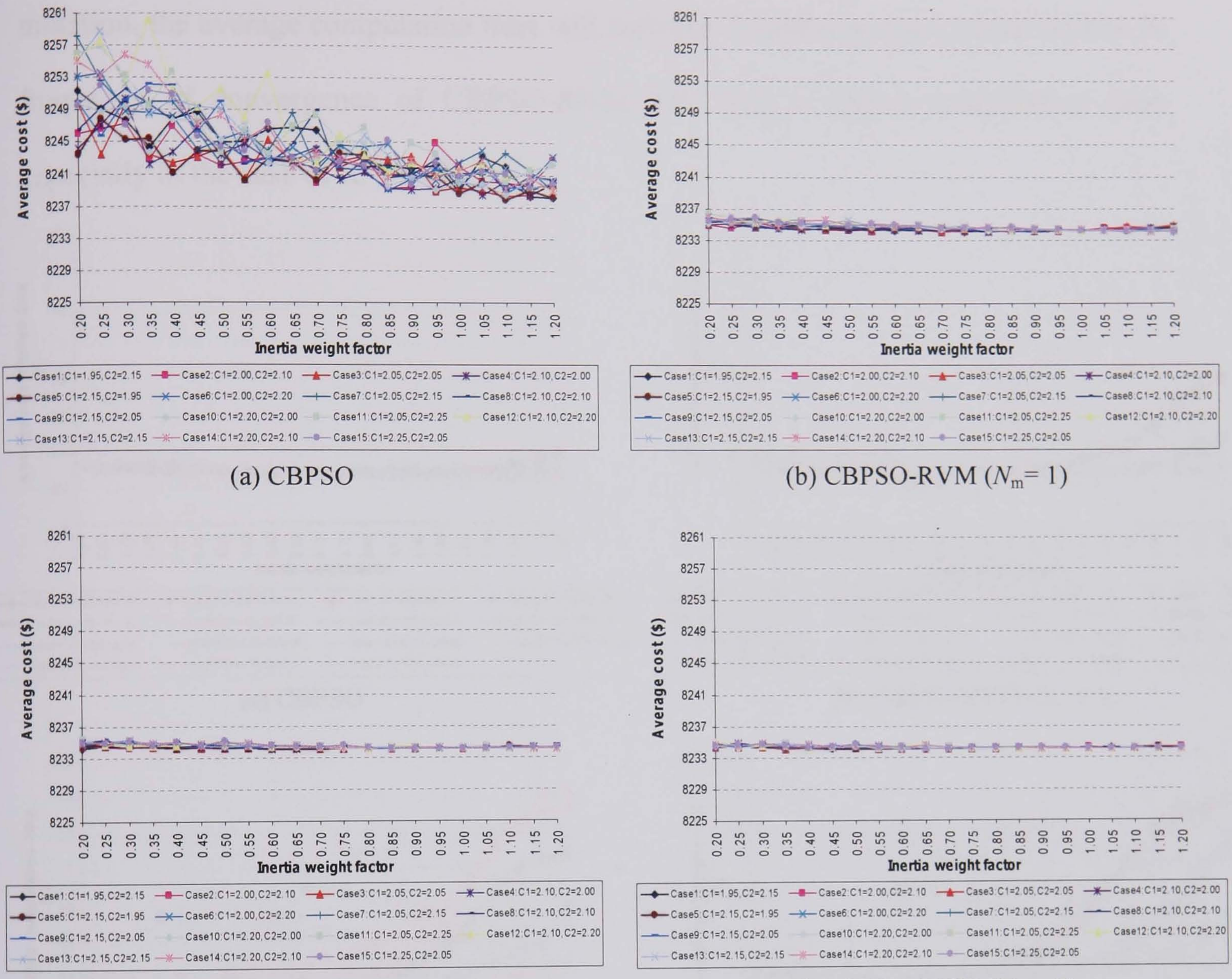


(d) CBPSO-RVM (N_m = 3)

Figure 6.10 Comparison of frequency of convergence to global solution between CBPSO and CBPSO-RVM for ED problem

C.2 Comparison in terms of average cost

The following discussion of average cost between CBPSO and CBPSO-RVM can be observed from Figure 6.11 (a)-(d). Whereas the average costs of CBPSO-RVM are close to the optimum solution ($8234.07) in all cases of different N_m, the average costs of CBPSO oscillate between $8261- $8238. Although the frequency of convergence of CBPSO-RVM with K = 0.73 (Figure 6.10 (b)-(d)) decrease significantly when w is more than 0.9, its average costs are still close to the optimum solution. Therefore, the simulation results show that applying the real-valued natural mutation (RVM) to CBPSO improves the searching ability of the PSO algorithms and also ensures the quality of the solutions.



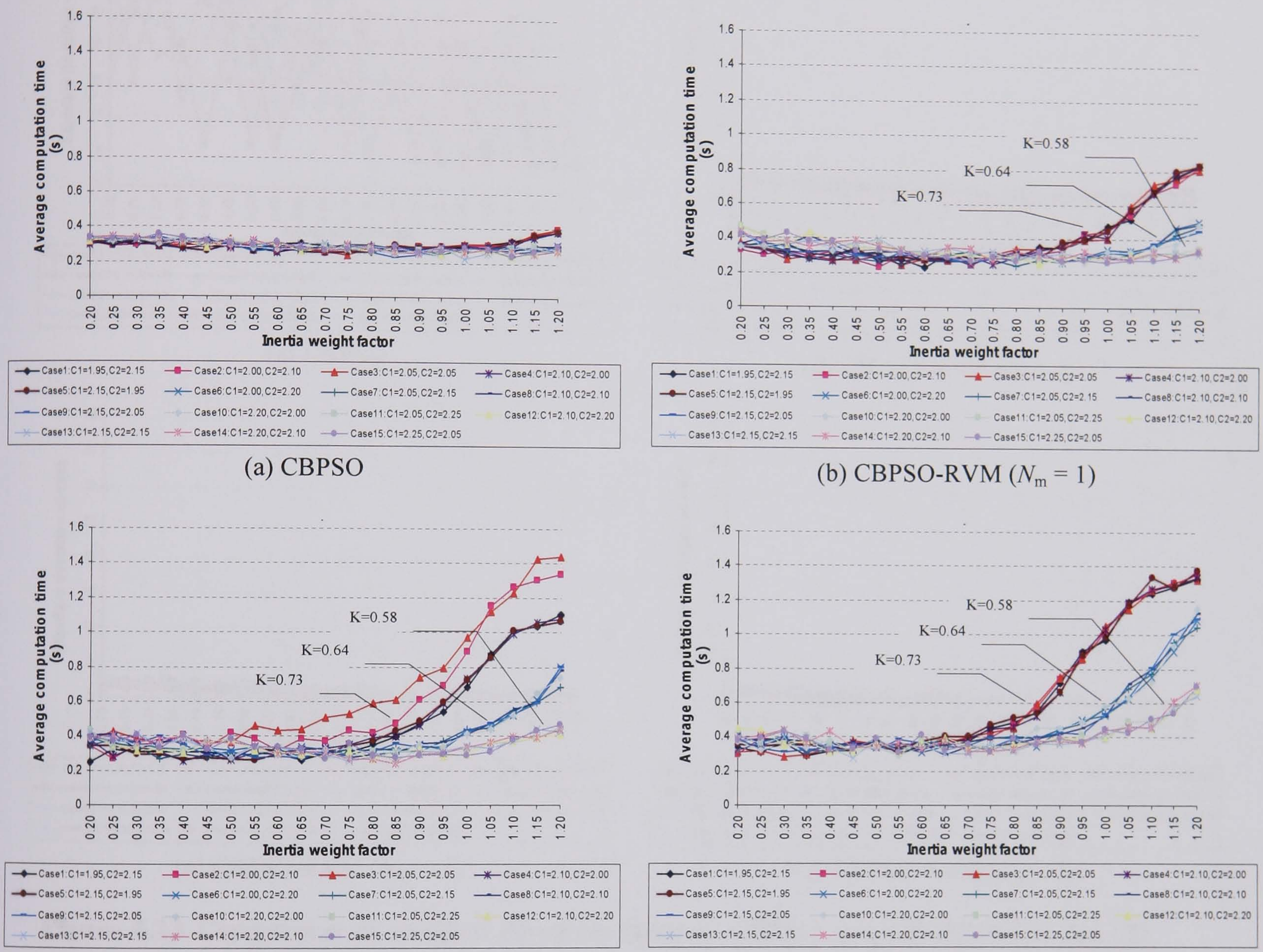
(a) CBPSO (b) CBPSO-RVM (N_m= 1)

(c) CBPSO-RVM (N_m= 2) (d) CBPSO-RVM (N_m = 3)

Figure 6.11 Comparison of average cost between CBPSO and CBPSO-RVM for ED problem

C.3 Comparison in terms of average computation time

The following discussion of average computation time between CBPSO and CBPSO-RVM can be observed from Figure 6.12 (a)-(d). The discussion of the average computation time here is somewhat similar to the comparison in both previous sections (BPSO vs. BPSO-RVM and CPSO vs. CPSO-RVM). From the simulation results, CBPSO still takes less computation time than CBPSO-RVM. The trend of average computation time of CBPSO-RVM is comparable with the trend of frequency of convergence to global solution in which it can be classified into three different groups of simulation results as well: (1) $c_1 + c_2 = 4.1$ or K = 0.73, (2) $c_1 + c_2 = 4.2$ or K = 0.64, and (3) $c_1 + c_2 = 4.3$ or K = 0.58. Because of the rise in the number of particles in mutation, the average computation time will increase. In addition, the gradual decline in frequency of convergence of CBPSO-RVM will lead to higher computation time especially in the case of K = 0.73.



(a) CBPSO

(b) CBPSO-RVM (N_m = 1)

(c) CBPSO-RVM (N_m = 2)

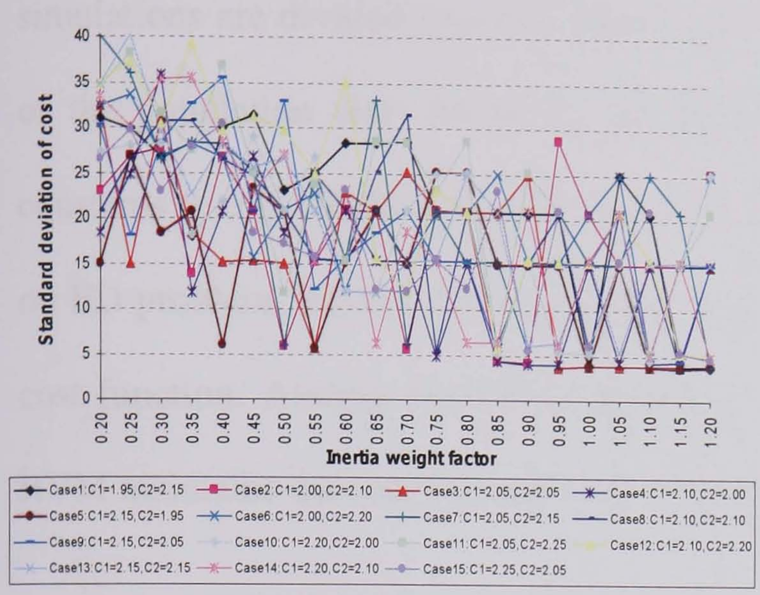
(d) CBPSO-RVM (N_m = 3)

Figure 6.12 Comparison of average computation time between CBPSO and CBPSO-RVM for ED problem

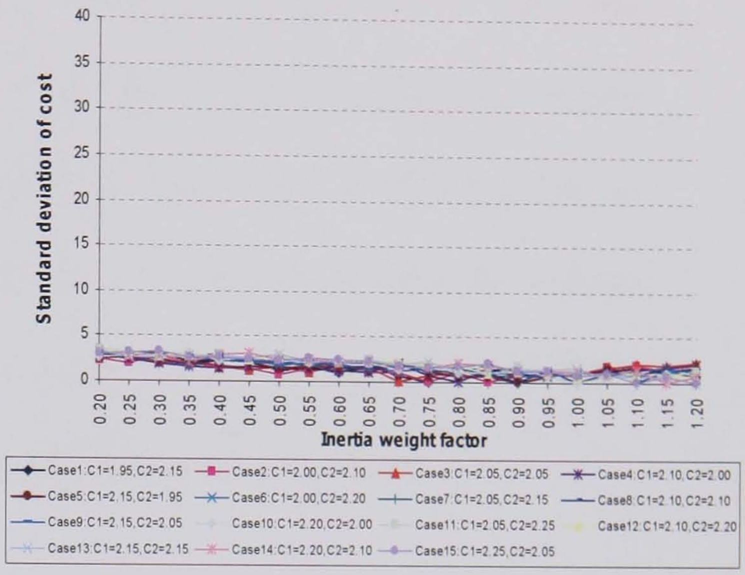
C.4 Comparison in terms of standard deviation

The following discussion of standard deviation between CBPSO and CBPSO-RVM can be observed from Figure 6.13 (a)-(d). Again, the trend of standard deviation of CBPSO and CBPSO-RVM is rather like the pair of CPSO vs. CPSO-RVM and BPSO vs. BPSO-RVM. The standard deviation of CBPSO differs from the standard deviation of

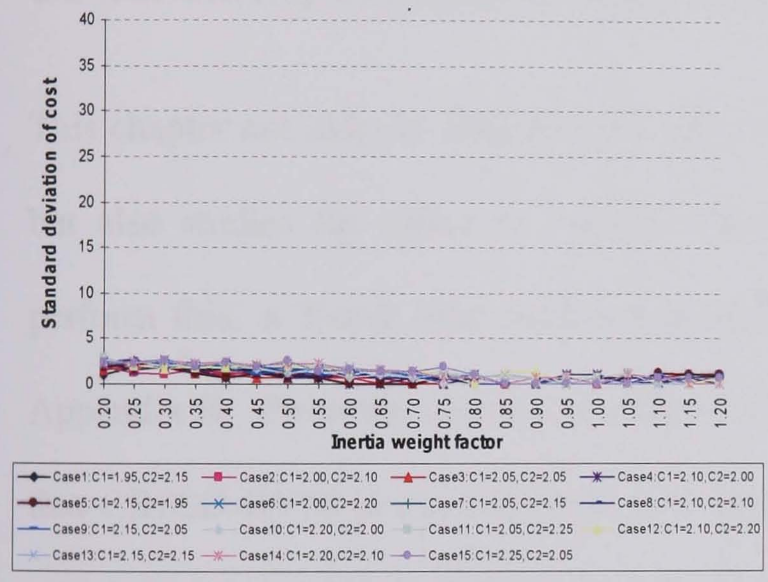
CBPSO-RVM in respect to its fluctuation. It can be seen that CBPSO-RVM is superior to its counterpart in that its standard deviation is rather stable.



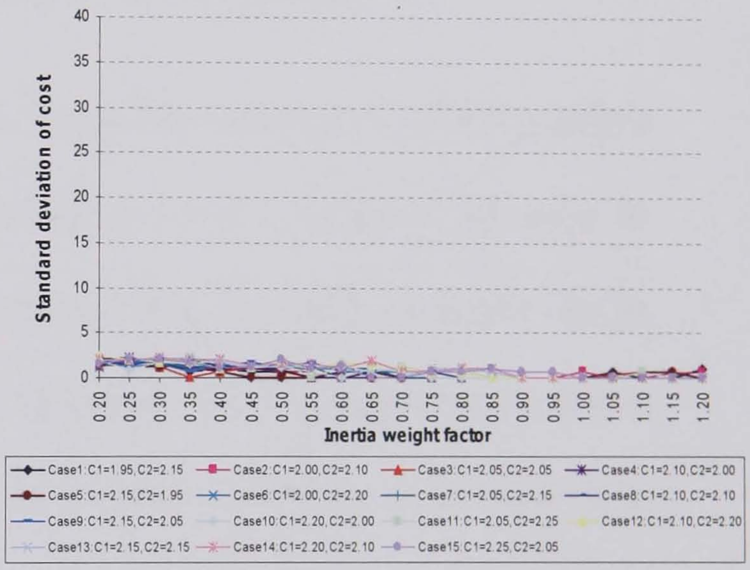
(a) CBPSO



(b) CBPSO-RVM (N_m= 1)



(c) CBPSO-RVM (N_m = 2)



(d) CBPSO-RVM (N_m = 3)

Figure 6.13 Comparison of standard deviation between CBPSO and CBPSO-RVM for ED problem

From the above comparisons between CBPSO and CBPSO-RVM, it apparently shows that CBPSO-RVM is shown to be less sensitive to the variation of parameters than CBPSO. For CBPSO-RVM, the group of c_1+c_2= 4.3 or K = 0.58 yields the reliable solutions in all cases, while it still represents the compromise between the solution quality and the calculation time.

6.2.3 Summary of sensitivity analysis of PSO parameters in ED problem

This section presents sensitivity analysis of PSO parameters in ED problem. The simulations are divided into two groups of study. The first group focuses on sensitivity of the population size. Secondly, the inertia weight factor (w) and the acceleration constants (c_1,c_2) will be taken into account. To illustrate their effect of PSO parameters on ED problem, the six PSO algorithms are tested on a 3-unit system with non-smooth cost function. Among the PSO algorithms, the simulation results show that CBPSO-RVM maintains the stability and reliability when the parameters are varied.

6.3 Sensitivity Analysis of PSO Parameters in UC Problem

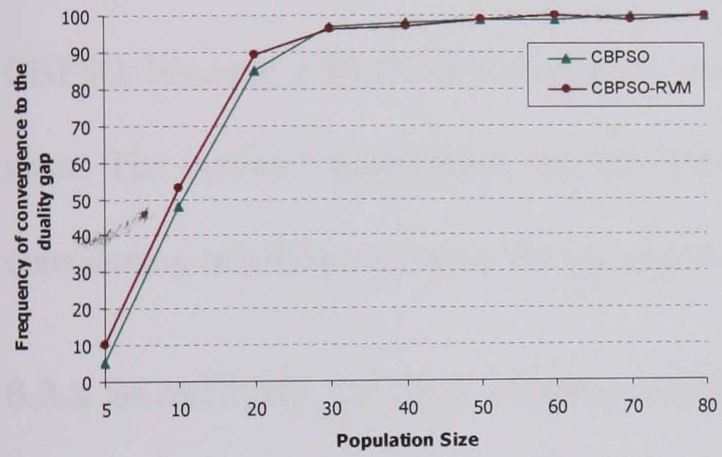
This chapter not only investigates the effect of varying PSO parameters on ED problem but also studies the effect of varying PSO parameters on UC problem. In order to perform this, a 3-unit 4-hr system has been adopted from [3] and the details are in Appendix B. From the sensitivity analysis in ED problem, the simulation results shows that CBPSO-RVM is more reliable and stable than other hybrid PSO algorithms (e.g. BPSO-RVM and CPSO-RVM); for this reason, only CBPSO-RVM and its counterpart will be considered to make this section concise. In addition, sensitivity of characteristics such as frequency of convergence to the duality gap, average Q*, average computation time, and standard deviation of the duality gap with respect to the population size and sensitivity of the inertia weight factor (w) and the acceleration constants (c_1,c_2) will also be investigated. Each plotted graph represents an average of the final solutions over 100 different initial runs in order to diminish the random effects. The predefined duality gap ($\varepsilon = 0.02$) is set to be the stopping criterion.

6.3.1 Sensitivity of the population size

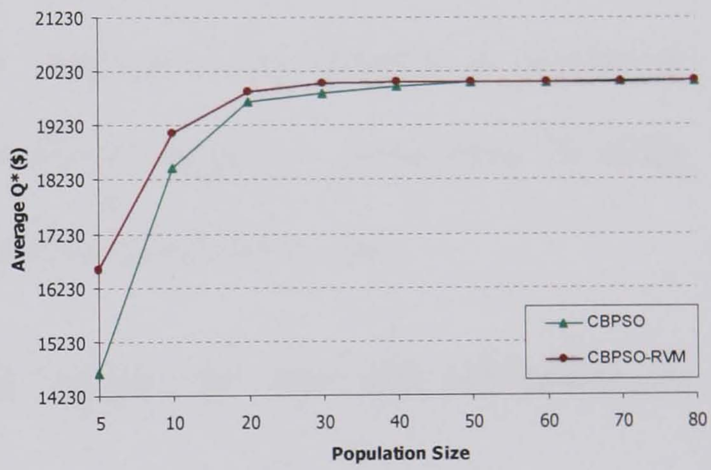
In this section, the parameters setting are almost the same as Section 6.2.1 (ED Section) except for the total number of iterations. The details of parameters setting are tabulated in Table 6.7. The simulation results will be presented and compared in four different patterns as shown in Figure 6.14 (a)-(d).

Table 6.7 Parameters used in the implementation of the CBPSO and CBPSO-RVM for UC problem

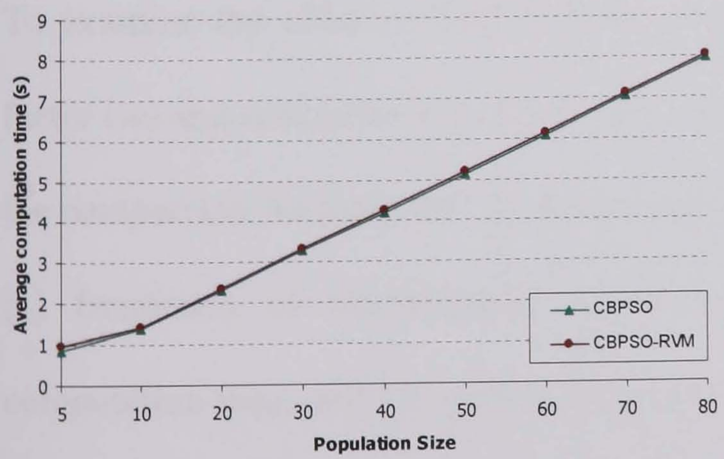
Methods	c_1/c_2	φ	K	w_{max}	w_{min}	N_m	Iter	Pop*
CBPSO	2.05	4.1	0.73	0.9	0.4	-	30	5-80
CBPSO-RVM	2.05	4.1	0.73	0.9	0.4	1	30	



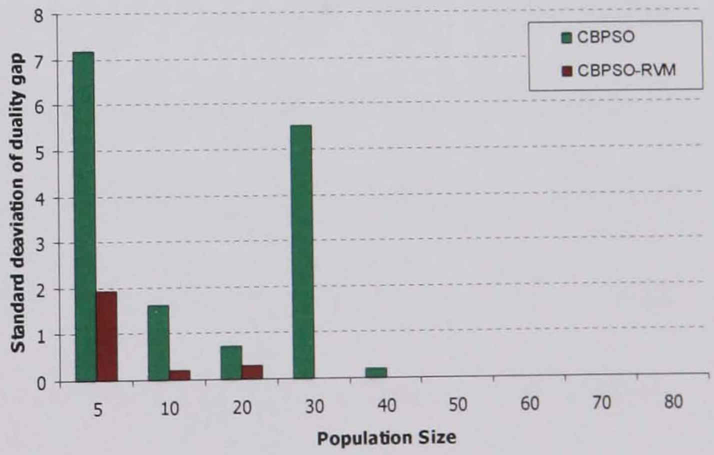
(a) Frequency of convergence to the duality gap



(b) Average cost



(c) Average computation time



(d) Standard deviation of cost

Figure 6.14 Comparison of various population sizes obtained by CBPSO and CBPSO-RVM for UC problem

Comparison of population size variation between CBPSO and CBPSO-RVM

From Figure 6.14, it follows that CBPSO-RVM performs better than CBPSO in terms of frequency of convergence to the duality gap. Regarding the average costs, the dual cost (Q*) has been chosen instead of the primary cost (J*) since the main concept of Largrange Relaxation (LR) for solving UC problem is to maximize Q*. From the simulation results, it can be seen that CBPSO-RVM can achieve the higher Q*, while the computation time of CBPSO-RVM is almost the same as CBPSO. It clearly shows that applying the mutation operator will not have affect on the computation time. Again, CBPSO-RVM shows its superiority over CBPSO with respect to the lower standard deviations. Furthermore, the standard deviations of CBPSO-RVM are almost zero when population size is greater than 20.

From above comparison, it can be concluded that CBPSO-RVM is superior to CBPSO because CBPSO-RVM shows better response to the variation of population sizes. The optimal population size of CBPSO-RVM should be greater than 20 while considering reliability of the solution quality and the computation time.

6.3.2 Sensitivity of the inertia weight factor (*w*) and the acceleration constants (c_1,c_2)

To examine the effects of other PSO parameters on UC problem, both inertia weight factor (w) and acceleration constants (c_1,c_2) will be considered. There are four aspects of the comparison between the simulation results of CBPSO and CBPSO-RVM as follows: (1) frequency of convergence to the duality gap, (2) average Q*, (3) average computation time, and (4) standard deviation of duality gap, respectively. In addition to the total number of iterations = 30, other parameters are still the same as ED section. Also, the different combinations of c_1 and c_2 will be taken into account, i.e. $c_1 + c_2 = 4.1$ making K = 0.73, $c_1 + c_2 = 4.2$ making K = 0.64, and $c_1 + c_2 = 4.3$ making K = 0.58. The parameter setting of CBPSO and CBPSO-RVM are tabulated in Table 6.8 and Table 6.9, respectively.

Table 6.8 Parameters used in the implementation of CBPSO for UC problem

Method	Case	c_1	c_2	φ	K	*W**	N_m	Pop	Iter
CBPSO	1	1.95	2.15	4.1	0.73	0.2-1.2	-	20	30
	2	2	2.1	4.1	0.73				
	3	2.05	2.05	4.1	0.73				
	4	2.1	2	4.1	0.73				
	5	2.15	1.95	4.1	0.73				
	6	2	2.2	4.2	0.64	0.2-1.2	-	20	30
	7	2.05	2.15	4.2	0.64				
	8	2.1	2.1	4.2	0.64				
	9	2.15	2.05	4.2	0.64				
	10	2.2	2	4.2	0.64				
	11	2.05	2.25	4.3	0.58	0.2-1.2	-	20	30
	12	2.1	2.2	4.3	0.58				
	13	2.15	2.15	4.3	0.58				
	14	2.2	2.1	4.3	0.58				
	15	2.25	2.05	4.3	0.58				

Table 6.9 Parameters used in the implementation of CBPSO-RVM for UC problem

Method	Case	c_1	c_2	φ	K	*W**	N_m**	Pop	Iter
CBPSO-RVM	1	1.95	2.15	4.1	0.73	0.2-1.2	1-3	20	30
	2	2	2.1	4.1	0.73				
	3	2.05	2.05	4.1	0.73				
	4	2.1	2	4.1	0.73				
	5	2.15	1.95	4.1	0.73				
	6	2	2.2	4.2	0.64	0.2-1.2	1-3	20	30
	7	2.05	2.15	4.2	0.64				
	8	2.1	2.1	4.2	0.64				
	9	2.15	2.05	4.2	0.64				
	10	2.2	2	4.2	0.64				
	11	2.05	2.25	4.3	0.58	0.2-1.2	1-3	20	30
	12	2.1	2.2	4.3	0.58				
	13	2.15	2.15	4.3	0.58				
	14	2.2	2.1	4.3	0.58				
	15	2.25	2.05	4.3	0.58				

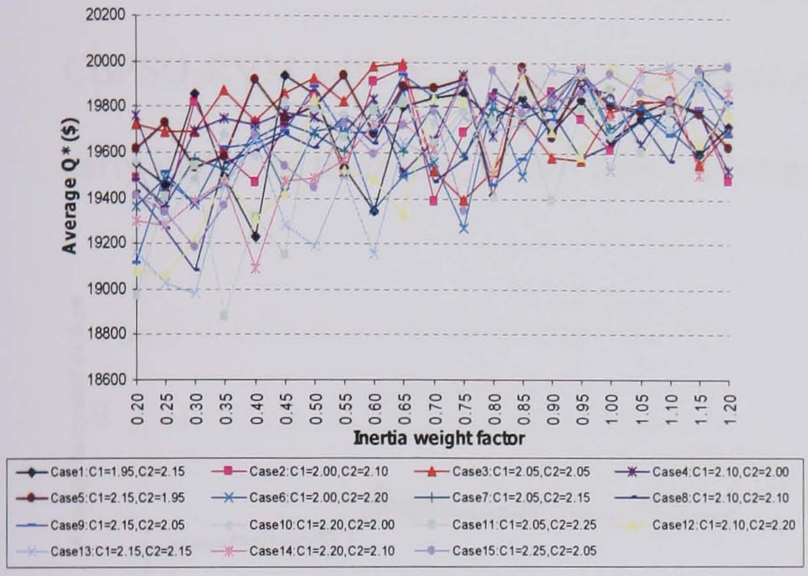
* The different sub-cases of *w* that start from 0.2 to 1.2 at 0.05 intervals.
** The different sub-cases of the number of particles that participate in mutation.

A.1 Comparison in terms of frequency of convergence to the duality gap

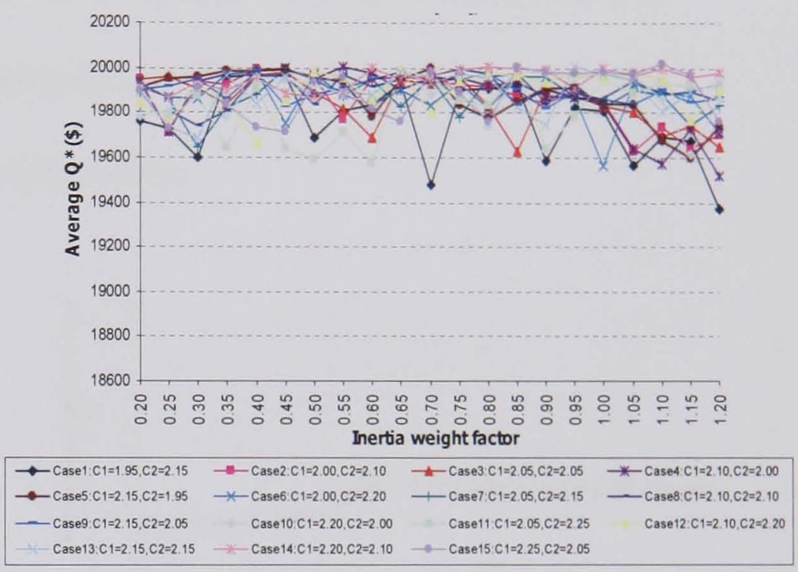
The following discussion of average computation time between CBPSO and CBPSO-RVM can be observed from Figure 6.15 (a)-(d). The response of CBPSO is worse than CBPSO-RVM, especially when the inertia weight factor (*w*) is approximately less than 0.55. For CBPSO itself, its simulation results can also be classified into three main categories as follows: (1) c_1+c_2 = 4.1 or K = 0.73, (2) c_1+c_2 = 4.2 or K= 0.64, and (3) c_1+c_2 = 4.3 or K = 0.58. The first category (K = 0.73) performs better than other categories except for *w* is greater than 1. Whereas the last category (K = 0.58) performs

*A.2 Comparison in terms of average Q**

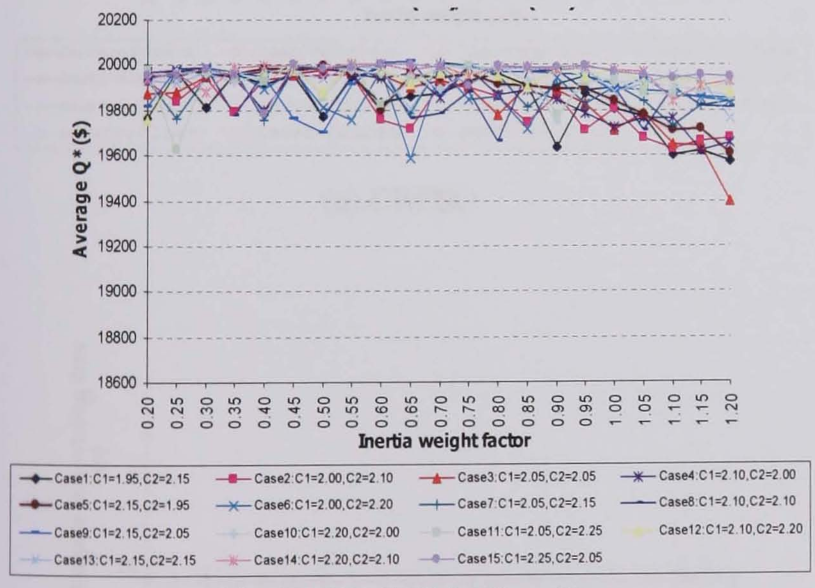
The discussion of average Q* between CBPSO and CBPSO-RVM can be observed from Figure 6.16 (a)-(d). From the comparison of these graphs, the average costs (Q*) of CBPSO confront with the oscillation problem, whilst the plotted Q* of CBPSO-RVM is reasonably stable.



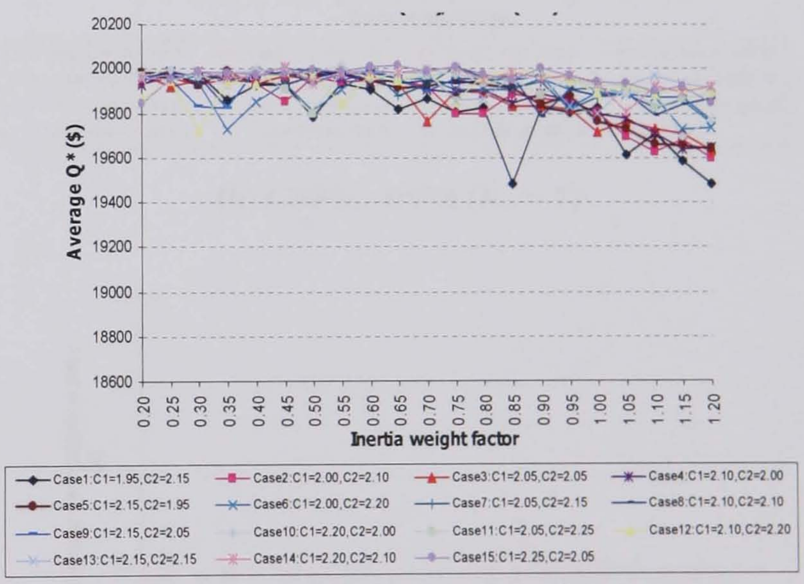
(a) CBPSO



(b) CBPSO-RVM (N_m = 1)



(c) CBPSO-RVM (N_m = 2)



(d) CBPSO-RVM (N_m = 3)

Figure 6.16 Comparison of average Q* between CBPSO and CBPSO-RVM for UC problem

A.3 Comparison in terms of average computation time

The following discussion of average computation time between CBPSO and CBPSO-RVM can be observed from Figure 6.17 (a)-(d). CBPSO-RVM normally takes more computation time than CBPSO. That is similar to the results obtained in ED section, except for N_m = 2 and 3. The difference in the graph of average computation time between N_m = 1 and N_m = 2, 3 is the smoothness of the graphs. It might be possible that CBPSO-RVM with N_m = 2 and 3 are more successful in closing the duality gap in the early iteration; therefore, it will take less computation time.

(a) CBPSO

(b) CBPSO-RVM (N_m = 1)

(c) CBPSO-RVM (N_m = 2)

(d) CBPSO-RVM (N_m = 3)

Figure 6.17 Comparison of average computation time between CBPSO and CBPSO-RVM for UC problem

A.4 Comparison in terms of standard deviation

The following discussion of standard deviation between CBPSO and CBPSO-RVM can be observed from Figure 6.18 (a)-(d).The trend of standard deviation in this section is quite similar to ED section in which the standard deviation of CBPSO-RVM is more reliable than CBPSO in terms of reaching a zero value of this characteristic.

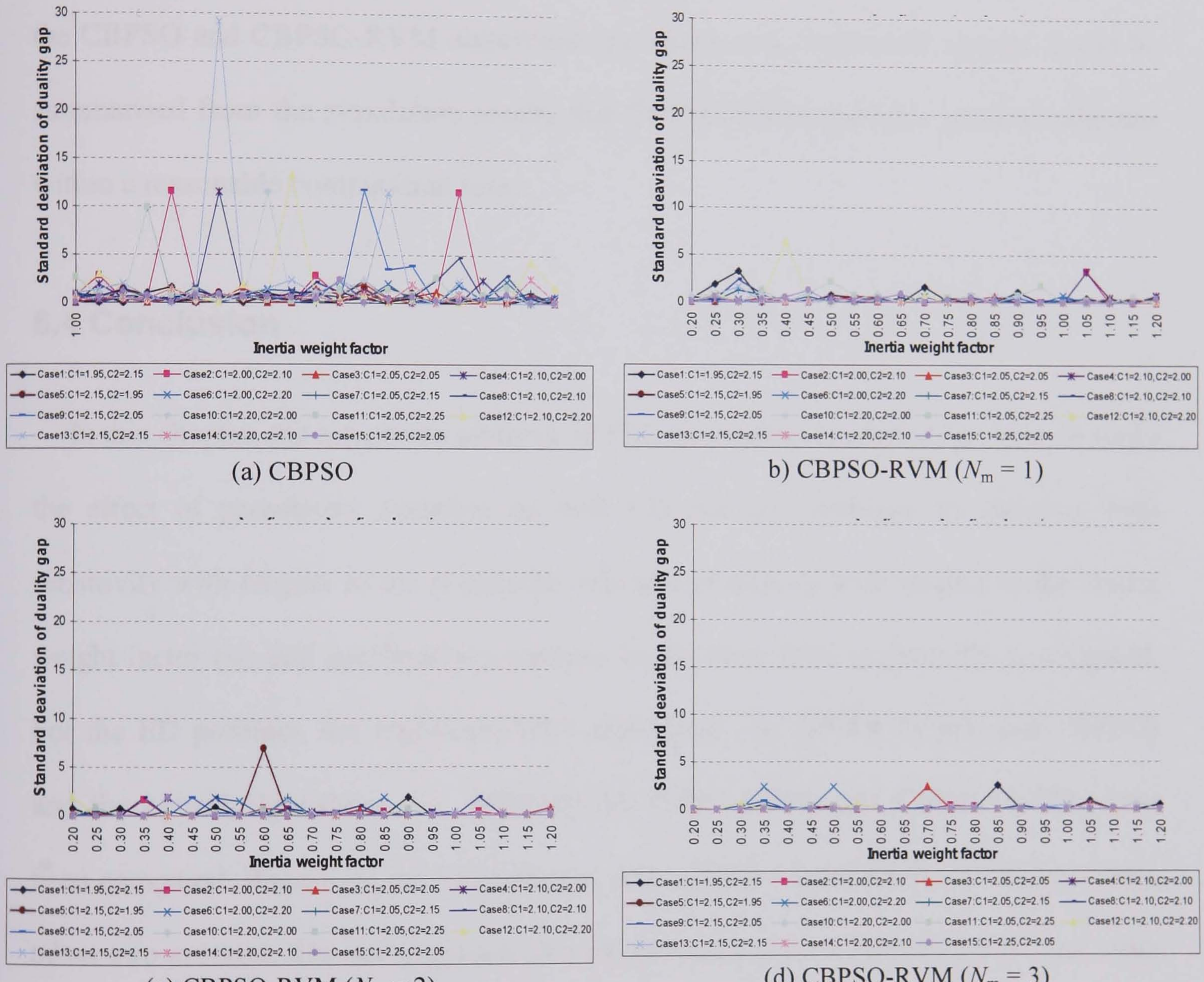


Figure 6.18 Comparison of standard deviation of duality gap between CBPSO and CBPSO-RVM for UC problem

From the above comparisons between CBPSO and CBPSO-RVM for UC problem, CBPSO-RVM is shown to be insensitive to the variations of inertia weight factor (w), and acceleration constants (c_1,c_2). It is rather similar to the section of ED in that the group of c_1+c_2= 4.3 or K = 0.58 is also found to be the best group in consideration of the reliability.

6.3.3 Summary of sensitivity analysis of PSO parameters in UC problem

This section presents sensitivity analysis of PSO parameters in UC problem by using CBPSO and CBPSO-RVM. The analysis is separated into two main categories. These are sensitivity with respect to the population size, as well as the inertia weight factor (w) including the acceleration constants (c_1,c_2). To illustrate their effects on an UC problem, the CBPSO and CBPSO-RVM algorithms are tested on a 3-unit 4-hr system. It can be summarised from the simulation results that CBPSO-RVM provides reliable solutions within a reasonable computation time.

6.4 Conclusion

In this chapter, the sensitivity analysis of PSO parameters is carried out so as to study the effect of parameters variation on both ED and UC problem. In addition, both sensitivity with respect to the population size and sensitivity with respect to the inertia weight factor (w) and acceleration constants (c_1,c_2) have been extensively investigated. For the ED problem, the traditional PSO algorithms (i.e. BPSO, CPSO, and CBPSO) and the hybrid algorithms (i.e. BPSO-RVM, CPSO-RVM, and CBPSO-RVM) have been compared. Regarding the UC problem, only CBPSO and CBPSO-RVM have been taken into account. The investigation of both problems can be classified into four main parts: (1) frequency of convergence, (2) average cost, (3) average computation time, and (4) standard deviation. It can be concluded from the comparisons that CBPSO-RVM provides efficiency and robustness for the ED and UC problems under consideration, since it is less sensitive to the PSO parameters. For choosing parameters of the CBPSO-RVM, the optimal group of $c_1+c_2= 4.3$ making $K = 0.58$ is recommended for both ED and UC problems.

Chapter 7: Conclusions and Future Work

7.1 Conclusions

This thesis proposes the application of PSO in ED and UC problems. Basically, it is found that PSO shows superiority to other evolutionary computation techniques in terms of less computation time, easy implementation with high quality solution, stable convergence characteristic, independent from initialisaion, etc. Unfortunately, PSO still has some drawbacks that are the problem of deficiency in searching diversity and the problem of its stability when tuning the parameters. Aiming at the drawbacks of the conventional PSO algorithms, this research proposes a hybrid method between the PSO algorithm and a real-valued mutation operator (CBPSO-RVM). The proposed method is subsequently tested on the mathematical benchmark problems. Moreover, it is also applied to solve the Economic Dispatch (ED) and Unit Commitment (UC) problems which are significant optimisation problems in power system operation. In addition, this research intends to study the influence of different parameters setting on the proposed method for ED and UC problems. From the empirical results, it can be concluded that CBPSO-RVM provides the global convergence property, the accurate solution, the efficiency and the feature of robust computation compared with the conventional PSO algorithms and other algorithms under consideration. However, further research directions are also proposed in order to improve the quality of this work.

7.2 Future work

This research can be further developed and extended in the two main categories as shown below:

7.2.1 Future work concerning the modified PSO algorithm itself

The directions of further research concerning the PSO algorithm itself are given as:

- Firstly, the direction of future work should extend the proposed method (CBPSO-RVM), which is the global-based version (Gbest model), to the local-based version of PSO (Lbest model) since the Lbest model has less possibility that the particles will plunge into a sub-optimal point.
- Secondly, the way to perform the real-valued natural mutation for CBPSO-RVM is to apply a constant mutation rate during the optimisation processes. That might be the reason for the lack of self-organised PSO dynamics. To reduce this impact, the mutations might be performed whenever the particles are stagnated in the search space.
- Finally, applying mutation operator to the standard PSO succeeds in overcoming the diversity problem; on the other hand, if there is too much mutation, it may lead to unnecessary amount of the diversity amongst the particles. The direction of the further work is to balance the diversity between global and local exploration. Therefore, it should be find out the new concept for dealing with this difficulty. A successful method, A Diversity-Guided Particle Swarm Optimizer [115], which has been successfully applied to solve

multimodal optimisation, could be one acceptable alternative.

7.2.2 Future work concerning the applications in power systems

The directions of further research concerning the applications in power systems are given as:

- The directions of the application in ED problem is to extend the proposed method to solve more complex ED problem that is Dynamic Economic Dispatch (DED) problem considering various constraints, i.e. voltage security constraints, environmental constraints, network constraints, transmission losses constraints, etc.
- The directions of the application in UC problem is to improve the calculation processes in the dynamic programming part where they heavily depend on the total number of generating units and the total number of hours. Therefore, the dynamic programming part should be modified and developed into the new on/off decision criteria [5] that reduces the stages of making decision by splitting the start-up cost into the value of minimum uptime for each unit.

References

[1] S. Sen and D. P. Kothari, "Optimal thermal generating unit commitment: a review," *International Journal of Electrical Power & Energy Systems*, vol. 20, pp. 443-451, 1998.

[2] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Trans. Power Syst*, vol. 11, pp. 83-92, Feb. 1996.

[3] A. J.Wood and B. F. Wollenberq, *Power Generation, Operation & Control*, 2 ed. New York: John Wiley, 1984.

[4] C.-P. Cheng, C.-W. Liu, and C.-C. Liu, "Unit commitment by Lagrangian relaxation and genetic algorithms," vol. 15, pp. 707 - 714, May. 2000.

[5] W. Ongsakul and N. Petcharaks, "Unit commitment by enhanced adaptive Lagrangian relaxation," *IEEE Trans. Power Syst*, vol. 19, pp. 620 - 628, Feb. 2004.

[6] Z.-L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst*, vol. 18, pp. 1187 - 1195, Aug. 2003.

[7] J.-B. Park, Y.-W. Jeong, W.-N. Lee, and J.-R. Shin, "An improved particle swarm optimization for economic dispatch problems with non-smooth cost functions," *In Conf. of IEEE Power Engineering General Meeting*, pp. 1-7, June, 2006.

[8] K. Y. Lee and M. A. El-Sharkawa, *A Tutorial Course on Evolutionary Computation Techniques for Power System Optimization*. Seoul, Korea: IFAC Symposium on Power Plants and Power, Sep. 2003.

[9] Y.-H. Song and M. R. Irving, "Optimisation techniques for electrical power systems. II. Heuristic optimisation methods," *Power Engineering Journal*, vol. 15, pp. 151-160, June, 2001.

[10] Y.-H. Song, "Introduction " in *Modern Optimisation Techniques in Power Ststems*, Y.-H. Song, Ed.: Kluwer Academic Publishers, 1999, pp. 1-13.

[11] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*: John Wiley & Sons, 1989.

[12] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function," *IEEE Trans. Power Syst.*, vol. 17, pp. 411 - 416, May. 2002.

[13] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO-SQP for economic dispatch with valve-point effect," *Electric Power Systems Research*, vol. 71, pp. 51-59, 2004.

[14] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 83 - 94, Feb. 2003.

[15] B. Zhao, C. X. Guo, and Y. J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," *IEEE Trans. Power Syst*, vol. 20, pp. 1070 - 1078, May. 2005

[16] J.-B. Park, K.-S. Lee, J.-R. Shin, and Kwang Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Trans. Power Syst*, vol. 20, pp. 34-42, Feb. 2005

[17] Z.-L. Gaing, "Discrete particle swarm optimization algorithm for unit commitment," *IEEE Power Eng. Soc. General Meeting*, vol. 1, pp. 418-424,

Jul. 2003

[18] M. A. Abido, "Optimal power flow using particle swarm optimization," *International Journal of Electrical Power & Energy Systems*, vol. 24, pp. 563-571, 2002.

[19] A. Stacey, M. Jancic, and I. Grundy, "Particle swarm optimization with mutation," *in Proc. Congr. Evol. Compt.*, vol. 2, pp. 1425 - 1430, Dec. 2003.

[20] R. C. Eberhart and Y. Shi, "Guest Editorial Special Issue on Particle Swarm Optimization," *IEEE Trans. on Evolutionary Computation*, vol. 8, pp. 201 - 203, June, 2004.

[21] M. R. AlRashidi and M. E. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Electric Power Systems," *IEEE Trans. on Evolutionary Computation: Accepted for future publication, Object Identifier 10.1109/TEVC.2006.880326*, pp. 1 - 1, 2006.

[22] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *in Proc. of IEEE Inte. Conf. on Systems, Man, and Cybernetics*, vol. 5, pp. 4104 - 4108, Oct. 1997.

[23] P. S. Andrews, "An Investigation into Mutation Operators for Particle Swarm Optimization," *in Proc. Congr. Evol. Compt.*, pp. 1044 - 1051, July 2006.

[24] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," *In Proc. Congr. Evol. Compt.(CEC 99)*, vol. 3, pp. 1931-1938, July, 1999.

[25] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," *In Proc. Congr. Evol. Compt.(CEC 2002)*, vol. 2, pp. 1671 - 1676, May, 2002.

[26] J. Kennedy and R. Mendes, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms," *In Proc. Conf. IEEE International Workshop on Soft Computing in Industrial Applications (SMCia/03)*, pp. 45 - 50 June, 2003.

[27] E. S. Peer, F. v. d. Bergh, and A. P. Engelbrecht, "Using neighbourhoods with the guaranteed convergence PSO," *In Proc. of Conf. on Swarm Intelligence Symposium (SIS 2003)*, pp. 235 - 242, Apr. 2003.

[28] M.-P. Song and G.-C. Gu, " Research on particle swarm optimization: A review," *In Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 4, pp. 2236 - 2241 Aug. 2004.

[29] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," *In Proc. Congr. Evol. Compt.*, pp. 69 - 73, May, 1998.

[30] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization," *In Proc. of the 1998 Annual Conference on Evolutionary programming, San Diego*, 1998.

[31] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," *in Proc. Congr. Evol. Compt.*, vol. 3, pp. 1951-1957, July. 1999

[32] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. on Evolutionary Computation*, vol. 6, pp. 58 - 73, Feb. 2002

[33] P. J. Angeline, "Using selection to improve particle swarm optimization," *in Conf. of IEEE World Congress on Computational Intelligence*, pp. 84 - 89 May, 1998.

[34] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," *in Proc. IEEE Swarm Intelligence Symposium (SIS'03)*, pp. 72 - 79 Apr. 2003

[35] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid Particle Swarm Optimiser with Breeding and Subpopulations," *In Proc. the Genetic and Evolutionary Comunication Conference*, 2001.

[36] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. on Evolutionary Computation*, vol. 8, pp. 240 - 255, June. 2004.

[37] T.-O. Ting, M. V. C. Rao, C. K. Loo, and S.-S. Ngu, "A new class of operators to accelerate particle swarm optimization," *In Proc. Congr. Evol. Compt.(CEC '03)*, vol. 4, pp. 2406 - 2410, Dec. 2003.

[38] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, "A Dissipative Particle Swarm Optimization," *In Proc. Congr. Evol. Compt.(CEC 2002)*, vol. 2, pp. 1456 - 1461, May, 2002.

[39] W.-J. Zhang and X.-F. Xie, "DEPSO: hybrid particle swarm with differential evolution operator," *In Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 3816 - 3821, Oct. 2003

[40] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, 3rd edition ed: Springer-Verlag, 1996.

[41] N. Li, Y.-Q. Qin, D.-B. Sun, and T. Zou, "Particle swarm optimization with mutation operator," *in Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 4, pp. 2251 - 2256, Aug. 2004.

[42] G. Zhang and H. Lu, "Hybrid Real-Coded Genetic Algorithm with Quasi-Simplex Technique," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 6, Oct. 2006.

[43] X. Hu, R. C. Eberhart, and Y. Shi, "Engineering optimization with particle swarm," *in Proc. IEEE Swarm Intelligence Symposium(SIS'03)*, pp. 53-57, Apr. 2003.

[44] M. Salazar-Lechuga and J. E. Rowe, "Particle swarm optimization and fitness sharing to solve multi-objective optimization problems," *In Proc. Congr. Evol. Compt.*, vol. 2, pp. 1204 - 1211, Sept. 2005

[45] H. Bai and B. Zhao, "A Survey on Application of Swarm Intelligence Computation to Electric Power System," *In Proc. of the 6th World Congress on Intelligent Control and Automation (WCICA)*, vol. 2, pp. 7587 - 7591, June. 2006

[46] 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," *In Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 5, pp. 2885 - 2889, Aug. 2005.

[47] Z.-l. Gaing, "Constrained optimal power flow by mixed-integer particle swarm optimization," *In Conf. of IEEE Power Engineering General Meeting*, vol. 1, pp. 243 - 250, June, 2005.

[48] D.-X. Niu, Y.-P. Ling, Q. Zhao, and Q.-Y. Zhao, "An Improved Particle Swarm Optimization Method Based on Borderline Search Strategy for Transmission Network Expansion Planning," *In Proc. Int. Conf. Machine Learning and Cybernetics*, pp. 2846 - 2850, Aug. 2006.

[49] P. Ren, L.-Q. Gao, N. Li, Y. Li, and Z.-L. Lin, "Transmission network optimal planning using the particle swarm optimization method," *In Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 7, pp. 4006 - 4011, Aug. 2005.

[50] H. Yoshida, K. Kawata, Y. Fukuyama, S.Takayama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control

considering voltage security assessment," *IEEE Trans. on Power Systems*, vol. 15, pp. 1232 - 1239 Nov. 2000.

[51] A.A.A.Esmin, G. Lambert-Torres, and A. C. Z. d. Souza, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. on Power Systems*, vol. 20, pp. 859 - 866 May, 2005.

[52] C.-M. Huang, C.-J. Huang, and M.-L. Wang, "A particle swarm optimization to identifying the ARMAX model for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, pp. 1126 - 1133, May 2005.

[53] S. P. Ghoshal, "Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control," *Electric Power Systems Research*, vol. 72, pp. 203-212, 2004.

[54] M. A. Abido, "Optimal design of power-system stabilizers using particle swarm optimization," *IEEE Trans. on Energy Conversion*, vol. 17, pp. 406 - 413 Sept. 2002.

[55] S. Kannan, S. M. R. Slochanal, P. Subbaraj, and N. P. Padhy, "Application of particle swarm optimization technique and its variants to generation expansion planning problem," *Electric Power Systems Research*, vol. 70, pp. 203-210, 2004.

[56] S. Kannan, S. M. R. Slochanal, and N. P. Padhy, "Application and comparison of metaheuristic techniques to generation expansion planning problem," *IEEE Trans. on Power Systems*, vol. 20, pp. 466 - 475, Feb. 2005.

[57] Z.-L. Gaing, "Constrained dynamic economic dispatch solution using particle swarm optimization," *In Conf. of IEEE Power Engineering General Meeting*, vol. 1, pp. 153 - 158, June, 2004.

[58] B. Zhao, C. Guo, and Y. Cao, "Dynamic economic dispatch in electricity market using particle swarm optimization algorithm," *In Proc. of the 5th World Congress on Intelligent Control and Automation (WCICA)*, vol. 6, pp. 5050 - 5054, June, 2004.

[59] D. N. Jeyakumar, T. Jayabarathi, and T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," *International Journal of Electrical Power & Energy Systems*, vol. In Press, Corrected Proof.

[60] T. A. A. Victoire and A. E. Jeyakumar, "Reserve Constrained Dynamic Dispatch of Units With Valve-Point Effects," *IEEE Trans. Power Syst*, vol. 20, pp. 1273 - 1282, Aug. 2005.

[61] K. S. Swarup and P. R. Kumar, "A new evolutionary computation technique for economic dispatch with security constraints," *International Journal of Electrical Power & Energy Systems*, vol. 28, pp. 273-283, 2006.

[62] A. I. Selvakumar and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems," *IEEE Trans. Power Syst.*, vol. 22, pp. 42 - 51, Feb. 2007.

[63] T.-O. Ting, M. V. C. Rao, C. K. Loo, and S. S. Ngu, "Solving Unitcommitment Problem Using Hybrid Particle Swarm Optimization," *Journal of Heuristics*, vol. 9, pp. 507-520, 2003.

[64] H. H. Balci and J. F. Valenzuela, "Schedualing electric power generations using particle swarm optimization combined with the lagrangian relaxation method," *Int. J. Appl. Math. Comput. sci.*, vol. 14, pp. 411-421, 2004.

[65] T. A. A. Victoire and A. E. Jeyakumar, "Unit commitment by a tabu-search-based hybrid-optimisation technique," *IEE Proceedings Generation, Transmission and Distribution*, vol. 152, pp. 563 - 574, July. 2005.

[66] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A novel approach for unit

commitment problem via an effective hybrid particle swarm optimization," *IEEE Trans. Power Syst*, vol. 21, pp. 411 - 418, Feb. 2006.

[67] B. Zhao, C. X. Guo, B. R. Bai, and Y. J. Cao, "An improved particle swarm optimization algorithm for unit commitment," *International Journal of Electrical Power & Energy Systems*, vol. 28, pp. 482-490, 2006.

[68] L. Cagnina, S. Esquivel, and R. Gallard, "Particle swarm optimization for sequencing problems: a case study," *In Proc. Congr. Evol. Compt. (CEC '04)*, vol. 1, pp. 536 - 541, June, 2004.

[69] C. Wang, J. Zhang, J. Yang, C. Hu, and J. Liu, "A Modified Particle Swarm Optimization Algorithm and its Application For Solving Traveling Salesman Problem," *In Proc. Int. Conf. Neural Networks and Brain (ICNN&B '05)*, pp. 689 - 694, Oct. 2005.

[70] Y.-Q. Qin, D.-B. Sun, N. Li, and Y.-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator," *In Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 4, pp. 2473 - 2478, Aug. 2004.

[71] J. Chen, Z. Ren, and X. Fan, "Particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters," *1st International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA 2006)*, pp. 990-994, Jan. 2006.

[72] J. Zhu and X. Gu, "A New Particle Swarm Optimization Algorithm for Short-Term Scheduling of Single-Stage Batch Plants with Parallel Lines," *In Proc. Sixth International Conference on Intelligent Systems Design and Applications (ISDA '06)* vol. 2, pp. 673 - 678 Oct. 2006.

[73] Z.-F. Hao, H. Huang, and X.-W. Yang, "A Novel Particle Swarm Optimization Algorithm for Solving Transportation Problem," *In Proc. Int. Conf. Machine Learning and Cybernetics*, pp. 2178 - 2183, Aug. 2006.

[74] D. S. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "On Solving Multiobjective Bin Packing Problems Using Particle Swarm Optimization," *In Proc. Congr. Evol. Compt. (CEC '06)*, pp. 2095 - 2102 July, 2006.

[75] N. Sinha and B. Purkayastha, "PSO embedded evolutionary programming technique for nonconvex economic load dispatch," *In Proc. of IEEE PES Power Systems Conference and Exposition*, vol. 1, pp. 66 - 71, Oct. 2004.

[76] Y.-H. Hou, L.-J. Lu, X.-Y. Xiong, and Y.-W. Wu, " Economic Dispatch of Power Systems Based on the Modified Particle Swarm Optimization Algorithm," *In Proc. IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific*, pp. 1 - 6, Aug. 2005.

[77] C. Yuehui, C. Haiyan, C. Jinfu, and D. Xianzhong, "An Improved Particle Swarm Optimization Algorithm for Multistage and Coordinated Planning of Transmission Systems," *In Proc. IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific*, pp. 1 - 6, Aug. 2005.

[78] D. Li, L. Gao, J. Zhang, and Y. Li, "Power System Reactive Power Optimization Based on Adaptive Particle Swarm Optimization Algorithm," *In Proc. of the 6th World Congress on Intelligent Control and Automation (WCICA)*, vol. 2, pp. 7572 - 7576, June. 2006.

[79] Y. Xiaohui, Y. Yanbi, W. Cheng, and Z. Xiaopan, "An Improved PSO Approach for Profit-based Unit Commitment in Electricity Market," *In Proc. IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific*, pp. 1 - 4 Aug. 2005.

[80] S. C. Esquivel and C. A. C. Coello, "On the use of particle swarm optimization with multimodal functions," *In Proc. Congr. Evol. Compt.(CEC 2003)*, vol. 2,

pp. 1130 - 1136, Dec. 2003.

[81] W. Jian, Y.-C. Xue, and J.-X. Qian, "An improved particle swarm optimization algorithm with neighborhoods topologies," *In Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 4, pp. 2332 - 2337, Aug. 2004.

[82] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in Proc. IEEE Int. Conf. Neural Networks*, vol. 4, pp. 1942 - 1948, Nov. 1995.

[83] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," *in Proc. Congr. Evol. Comput.*, vol. 1, pp. 81 - 86, May. 2001

[84] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," *in Proc. Congr. Evol. Compt.*, vol. 3, pp. 1945-1950, Jul. 1999.

[85] Y. Fukuyama, "Particle Swarm Optimization Techniques with applications in Power System," in *Evolutionary Computation Techniques for Power System Optimization*, K. Y. Lee and M. A. El-Sharkawa, Eds. Seoul, Korea: Tutorial given at The IFAC Symposium on Power Plants and Power, Sep. 2003, pp. 45-62.

[86] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *in Proc. Congr. Evol. Compt.*, vol. 1, pp. 84 - 88, Jul. 2000.

[87] R. A. Krohling and L. d. S. Coelho, "PSO-E: Particle Swarm with Exponential Distribution," *in Proc. Congr. Evol. Compt.*, pp. 1428 -1433, July 2006.

[88] Q.-L. Zhang, X. Li, and Q.-A. Tran, "A modified particle swarm optimization algorithm," *In Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 5, pp. 2993 - 2995, Aug. 2005.

[89] J.H.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, pp. 1030-1038, Aug. 1993

[90] D. Liu and Y. Cai, "Taguchi method for solving the economic dispatch problem with nonsmooth cost functions," *IEEE Trans. Power Syst*, vol. 20, pp. 2006-2014, Nov. 2005.

[91] R. E. Perez-Guerrero and J.R Cedeno-Maldonado, "Economic power dispatch with non-smooth cost functions using differential evolution," *In Proc. of the 37th Annual North American on Power Symposium*, pp. 183 -190, Oct. 2005.

[92] 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, pp. 1690 - 1699, Nov. 2005.

[93] C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," *IEEE Trans. Power App. Syst*, vol. PAS-103, pp. 1170-1175, Jun. 1984.

[94] Y.-M. Park, J. R. Won, and J. B. Park, "A new approach to economic load dispatch based on improved evolutionary programming," *Eng. Intell. Syst. Elect. Eng. Commu.*, vol. 6, pp. 103-110, Jun. 1998.

[95] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Syst*, vol. 8, pp. 1325 - 1332, Aug. 1993.

[96] W.-M. Lin, F.-S. Cheng, and M.-T. Tsay, "An improved tabu search for economic dispatch with multiple minima," *IEEE Trans. Power Syst*, vol. 17, pp. 108 - 112, Feb. 2002

[97] A. Pereira-Neto, C. Unsihuay, and O. R. Saavedra, "Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch

problem with generator constraints," *Proc. Inst. Elect. Eng. Gen. Trans. Distrib.*, vol. 152, pp. 653 - 660, Sep. 2005.
[98] K. A. Juste, H. Kita, E. Tunaka, and J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," *IEEE Trans. Power Syst*, vol. 14, pp. 1452 - 1459 Nov. 1999
[99] S. O. Orero and M. R. Irving, "A combination of the genetic algorithm and lagrangian relaxation decomposition techniques for the generation unit commitment problem," *Electric Power Systems Research*, vol. 43, pp. 149-156, 1997.
[100] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming : Theory and Algorithms*. New york: John Wiley & Sons, 1979.
[101] M. L. Fisher, "Optimal Solution of Scheduling Problem Using Lagrange Multipliers: Part I," *Operations Research*, vol. 21, pp. 1114-1127, 1973.
[102] S. Dekrajangpetch, G. B. Sheble, and A. J. Conejo, "Auction implementation problems using Lagrangian relaxation," *IEEE Trans. Power Syst.*, vol. 14, pp. 82 - 88 Feb. 1999.
[103] A. Merlin and P. Sandrin, "A New Method for Unit Commitment at Electricite De France," *IEEE Trans. Power Syst. App. Syst.*, vol. PAS-102, pp. 1218 - 1225, May 1983.
[104] M. L. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, vol. 27, pp. 1-18, Jan. 1981.
[105] P. B. Luh, D. Zhang, and R. N. Tomastik, "An algorithm for solving the dual problem of hydrothermal scheduling," *IEEE Trans. Power Syst.*, vol. 13, pp. 593 - 600, May 1998.
[106] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid LR-EP for solving new profit-based UC problem under competitive environment," *IEEE Trans. Power Syst*, vol. 18, pp. 229 - 237 Feb. 2003.
[107] C.-L. Tseng, S. S. Oren, A. J. Svoboda, and R. B. Johnson, "A unit decommitment method in power system scheduling," *International Journal of Electrical Power & Energy Systems*, vol. 19, pp. 357-365, 1997.
[108] C. W. Richter and G. B. Sheble, "A profit-based unit commitment GA for the competitive environment," *IEEE Trans. Power Syst.*, vol. 15, pp. 715 - 721 May, 2000.
[109] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A new profit-based unit commitment considering power and reserve generating," *in Conf. of IEEE Power Engineering Society Winter Meeting*, vol. 2, pp. 1311 - 1316, Jan. 2002.
[110] M. Shahidehpour, H. Yamin, and Z. Y. Li, "Price-Based Unit Commitment," in *Market Operations in Electric Power Systems*. New York: John Wiley & Sons, Inc., 2002.
[111] E. H. Allen and M. D. Ilic, "Reserve markets for power systems reliability," *IEEE Trans. Power Syst*, vol. 15, pp. 228 - 233 Feb. 2000.
[112] J. Yu, J. Zhou, W. Wu, J. Yang, and W. Yu, "Solution of the profit-based unit commitment problem by using multi-agent system," *In Proc. of the 5th World Congress on Intelligent Control and Automation (WCICA)*, vol. 6, pp. 5079 - 5083, June. 2004.
[113] J. F. Schutte and A. A. Groenwold, "A study of global optimization using particle swarm," *Journal of Global Optimization*, vol. 31, pp. 93-108, 2005.
[114] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, pp.

317-325, 2003.

[115] J. Riget and J. S. Vesterstroem, "A diversity-guided particle swarm optimizer - the ARPSO," *Technical Report 2002-02, Department of Computer Science, University of Aarhus*, 2002.

Publications Related to the Thesis

[1] P. Sriyanyong, Y. H. Song, and P. J. Turner, "Particle Swarm Optimisation for Operational Planning: Unit Commitment and Economic Dispatch," in Evolutionary Scheduling (Studies in Computational Intelligence), vol. 49, K. Dahal, K. C. Tan, and P. I. Cowling, Eds.: Springer-Verlag, Feb. 2007, pp. 628.

[2] P. Sriyanyong and Y. H. Song, "Unit commitment using particle swarm optimization combined with Lagrange relaxation," in Conf. of IEEE Power Engineering General Meeting, vol. 3, pp. 2752 - 2759, June, 2005.

[3] P. Sriyanyong, H. Y. Lu, and Y.H. Song, "Hybrid PSO with Real-Valued Natural Mutation and Application in Economic Dispatch with Nonsmooth Cost Function." to be submitted to International Journal of Electrical Power & Energy Systems for review.

Appendix A: Economic Dispatch

A.1 Units data for test Case A (3-generator system) where a, b, c are cost coefficients in the production cost function [3].

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c
1	150	600	0.001562	7.92	561
2	100	400	0.001940	7.85	310
3	50	200	0.004820	7.97	78

A.2 Units data with multiple fuels for test Case B (10-generator system) [93]

U	Generation							F	Cost Coefficients		
	Min	F_1	P_1	F_2	P_2	F_3	Max		a	b	c
1	100		196		250			1	.2176e-2	-.3975e0	.2697e2
		1		2				2	.1861e-2	-.3059e0	.2113e2
2	50		114		157		230	1	.4194e-2	-.1269e1	.1184e3
		2		3		1		2	.1138e-2	-.3988e-1	.1865e1
								3	.1620e-2	-.1980e0	.1365e2
3	200		332		388		500	1	.1457e-2	-.3116e0	.3979e2
		1		3		2		2	.1176e-4	.4864e0	-.5914e2
								3	.8035e-3	.3389e-1	-.2876e1
4	99		138		200		265	1	.1049e-2	-.3114e-1	.1983e1
		1		2		3		2	.2758e-2	-.6348e0	.5285e2
								3	.5935e-2	-.2338e1	.2668e3
5	190		338		407		490	1	.1066e-2	-.8733e-1	.1392e2
		1		2		3		2	.1597e-2	-.5206e0	.9976e2
								3	.1498e-3	.4462e0	-.5399e2
6	85		138		200		265	1	.2758e-2	-.6348e0	.5285e2
		2		1		3		2	.1049e-2	-.3114e-1	.1983e1
								3	.5935e-2	-.2338e1	.2668e3
7	200		331		391		500	1	.1107e-2	-.1325e0	.1893e2
		1		2		3		2	.1165e-2	-.2267e0	.4377e2
								3	.2454e-3	.3559e0	-.4335e2
8	99		138		200		265	1	.1049e-2	-.3114e-1	.1983e1
		1		2		3		2	.2758e-2	-.6348e0	.5285e2
								3	.5935e-2	-.2338e1	.2668e3
9	130		213		370		440	1	.1554e-2	-.5675e0	.8853e2
		3		1		3		2	.7033e-2	-.4514e-1	.1530e2
								3	.6121e-3	-.1817e-1	.1423e2
10	200		362		407		490	1	.1102e-2	-.9938e-1	.1397e2
		1		3		2		2	.4164e-4	.5084e0	-.6113e2
								3	.1137e-2	-.2024e0	.4671e2

A.3 Units data for test Case C (3-generator system) where a, b, c, e and f are cost coefficients in the production cost function [95].

Generator	P_{min}(MW)	P_{max} (MW)	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	100	400	0.001940	7.85	310	200	0.042
3	50	200	0.004820	7.97	78	150	0.063

A.4 Units data for test Case D (40-generator system) [14].

Generator	P_{min}(MW)	P_{max}(MW)	a	b	c	e	f
1	36	114	0.00690	6.73	94.705	100	0.084
2	36	114	0.00690	6.73	94.705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.20	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.40	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

A.5 Units data considering multiple fuels and valve-point loading for test Case E (10-generator system)[92]

U	Min	F_1	P_1	F_2	P_2	F_3	Max	F	a	b	c	e	f
	Generation								Cost Coefficients				
1	100	1	196	2	250			1	.2176e-2	-.3975e0	.2697e2	.2697e-1	-.3975e1
								2	.1861e-2	-.3059e0	.2113e2	.2113e-1	-.3059e1
2	50	2	114	3	157	1	230	1	.4194e-2	-.1269e1	.1184e3	.1184e0	-.1269e2
								2	.1138e-2	-.3988e-1	.1865e1	.1865e-2	-.3988e0
								3	.1620e-2	-.1980e0	.1365e2	.1365e-1	-.1980e1
3	200	1	332	3	388	2	500	1	.1457e-2	-.3116e0	.3979e2	.3979e-1	-.3116e1
								2	.1176e-4	.4864e0	-.5914e2	-.5914e-1	.4864e1
								3	.8035e-3	.3389e-1	-.2876e1	-.2876e-2	.3389e0
4	99	1	138	2	200	3	265	1	.1049e-2	-.3114e-1	.1983e1	.1983e-2	-.3114e0
								2	.2758e-2	-.6348e0	.5285e2	.5285e-1	-.6348e1
								3	.5935e-2	-.2338e1	.2668e3	.2668e0	-.2338e2
5	190	1	338	2	407	3	490	1	.1066e-2	-.8733e-1	.1392e2	.1392e-1	-.8733e0
								2	.1597e-2	-.5206e0	.9976e2	.9976e-1	-.5206e1
								3	.1498e-3	.4462e0	-.5399e2	-.5399e-1	-.4462e1
6	85	2	138	1	200	3	265	1	.2758e-2	-.6348e0	.5285e2	.5285e-1	-.6348e1
								2	.1049e-2	-.3114e-1	.1983e1	.1983e-2	-.3114e0
								3	.5935e-2	-.2338e1	.2668e3	.2668e0	-.2338e2
7	200	1	331	2	391	3	500	1	.1107e-2	-.1325e0	.1893e2	.1893e-1	-.1325e1
								2	.1165e-2	-.2267e0	.4377e2	.4377e-1	-.2267e1
								3	.2454e-3	.3559e0	-.4335e2	-.4335e-1	.3559e1
8	99	1	138	2	200	3	265	1	.1049e-2	-.3114e-1	.1983e1	.1983e-2	-.3114e0
								2	.2758e-2	-.6348e0	.5285e2	.5285e-1	-.6348e1
								3	.5935e-2	-.2338e1	.2668e3	.2668e0	-.2338e2
9	130	3	213	1	370	3	440	1	.1554e-2	-.5675e0	.8853e2	.8853e-1	-.5675e1
								2	.7033e-2	-.4514e-1	.1530e2	.1423e-1	-.1871e0
								3	.6121e-3	-.1817e-1	.1423e2	.1423e-1	-.1871e0
10	200	1	362	3	407	2	490	1	.1102e-2	-.9938e-1	.1397e2	.1397e-1	-.9938e0
								2	.4164e-4	.5084e0	-.6113e2	-.6113e-1	.5084e1
								3	.1137e-2	-.2024e0	.4671e2	.4671e-1	-.2024e1

Appendix B: Unit Commitment

B.1 Unit data of the 3-unit 4-hour system for the traditional UC [3]

	Unit 1	Unit 2	Unit 3
P_{max} (MW)	600	400	200
P_{min} (MW)	100	100	50
a ($/MW^2 h)	0.002	0.0025	0.005
b ($/MW h)	10	8	6
c ($/h)	500	300	100

B.2 Load demand of 3-unit 4-hour system for the traditional UC [3]

Hour	1	2	3	4
Load (MW)	170	520	1100	330

B.3 Unit data of the 10-unit 24-hour system for the traditional UC and the profit-based UC [106], [2]

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5
P_{max} (MW)	455	455	130	130	162
P_{min} (MW)	150	150	20	20	25
a ($/MW^2 h)	0.00048	0.00031	0.002	0.00211	0.00398
b ($/MW h)	16.19	17.26	16.60	16.50	19.70
c ($/h)	1000	970	700	680	450
min up (h)	8	8	5	5	6
min down (h)	8	8	5	5	6
hot start cost ($)	4500	5000	550	560	900
Cold start cost ($)	9000	10000	1100	1120	1800
Cold start hrs (h)	5	5	4	4	4
Initial status (h)	8	8	-5	-5	-6

	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
P_{max} (MW)	80	85	55	55	55
P_{min} (MW)	20	25	10	10	10
a ($/MW^2 h)	0.00712	0.00079	0.00413	0.00222	0.00173
b ($/MW h)	22.26	27.74	25.92	27.27	27.79
c ($/h)	370	480	660	665	670
min up (h)	3	3	1	1	1
min down (h)	3	3	1	1	1
hot start cost ($)	170	260	30	30	30
Cold start cost ($)	340	520	60	60	60
Cold start hrs (h)	2	2	0	0	0
Initial status (h)	-3	-3	-1	-1	-1

B.4 Load demand of 10-unit 24-hour system for the traditional UC [2]

Hour	Load (MW)	Hour	Load (MW)
1	700	13	1400
2	750	14	1300
3	850	15	1200
4	950	16	1050
5	1000	17	1000
6	1100	18	1100
7	1150	19	1200
8	1200	20	1400
9	1300	21	1300
10	1400	22	1100
11	1450	23	900
12	1500	24	800

B.5 Unit data of 3-unit 12-hour system for profit-based UC [106]

	Unit 1	Unit 2	Unit 3
Pmax (MW)	600	400	200
Pmin (MW)	100	100	50
a ($/h)	500	300	100
b ($/MW h)	10	8	6
c ($/MW^2 h)	0.002	0.0025	0.005
min up (h)	3	3	3
min down (h)	3	3	3
Startup cost (h)	450	400	300
Initial status (h)	-3	3	3

B.6 Load demand of 3-unit 12-hour system for profit-based UC [106]

Hour	Forecasted Demand (MW)	Forecasted Reserve (MW)
1	170	20
2	250	25
3	400	40
4	520	55
5	700	70
6	1050	95
7	1100	100
8	800	80
9	650	65
10	330	35
11	400	40
12	550	55

B.7 Forecasted spot prices and reserve prices of 3-unit 12-hour system for profit-based UC [106]

Hour	Forecasted Spot Price ($/MW-H)	Forecasted Reserve Price ($/MW-H)
1	10.55	0.4220
2	10.35	0.4140
3	9.00	0.3600
4	9.45	0.3780
5	10.00	0.4000
6	11.25	0.4500
7	11.30	0.4520
8	10.65	0.4260
9	10.35	0.4140
10	11.20	0.4480
11	10.75	0.4300
12	10.60	0.4240

Note: Forecasted reserve price = (0.04* Forecasted spot price)

B.8 Forecasted load demand and spinning reserve of 10-unit 24-hour for the profit-based UC [106]

Hour	Forecasted Demand (MW)	Forecasted Reserve (MW)	Hour	Forecasted Demand (MW)	Forecasted Reserve (MW)
1	700	70	13	1400	140
2	750	75	14	1300	130
3	850	85	15	1200	120
4	950	95	16	1050	105
5	1000	100	17	1000	100
6	1100	110	18	1100	110
7	1150	115	19	1200	120
8	1200	120	20	1400	140
9	1300	130	21	1300	130
10	1400	140	22	1100	110
11	1450	145	23	900	90
12	1500	150	24	800	80

B.9 Forecasted spot prices and reserve prices for 10-unit 24-hour for profit-based UC [106]

Hour	Forecasted Spot Price ($/MW-H)	Forecasted Reserve Price ($/MW-H)	Hour	Forecasted Spot Price ($/MW-H)	Forecasted Reserve Price ($/MW-H)
1	22.15	0.2215	13	24.6	0.246
2	22	0.22	14	24.5	0.245
3	23.1	0.231	15	22.5	0.225
4	22.65	0.2265	16	22.3	0.223
5	23.25	0.2325	17	22.25	0.2225
6	22.95	0.2295	18	22.05	0.2205
7	22.5	0.225	19	22.2	0.222
8	22.15	0.2215	20	22.65	0.2265
9	22.8	0.228	21	23.1	0.231
10	29.35	0.2935	22	22.95	0.2295
11	30.15	0.3015	23	22.75	0.2275
12	31.65	0.3165	24	22.55	0.2255

Note: Forecasted reserve price = (0.01* Forecasted spot price)