APPLICATION OF IMPROVED PARTICLE SWARM OPTIMIZATION IN ECONOMIC DISPATCH OF POWER SYSTEMS

By

GHISLAIN YANICK GNINKEU TCHAPDA

Submitted in accordance with the requirements
For the degree of

MAGISTER TECHNOLOGIAE

In the subject

ELECTRICAL ENGINEERING

At the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: PROF Z WANG

March 2018

DECLARATION

Name: GhislainYanick GNINKEU TCHAPDA

Student Number: 54107342

Degree: Magister Technologiae (MTech) Engineering (Electrical)

Title: APPLICATION OF IMPROVED PARTICLE SWARM OPTIMIZATION IN ECONOMIC DISPATCH OF POWER SYSTEMS

I declare that the above dissertationis my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

.... 24/04/2018

SIGNATURE DATE

ii

ABSTRACT

Economic dispatch is an important optimization challenge in power systems. It helps to find the optimal output power of a number of generating units that satisfy the system load demand at the cheapest cost, considering equality and inequality constraints. Many nature inspired algorithms have been broadly applied to tackle it such as particle swarm optimization. In this dissertation, two improved particle swarm optimization techniques are proposed to solve economic dispatch problems. The first is a hybrid technique with Bat algorithm. Particle swarm optimization as the main optimizer integrates bat algorithm in order to boost its velocity and to adjust the improved solution. The second proposed approach is based on Cuckoo operations. Cuckoo search algorithm is a robust and powerful technique to solve optimization problems. The study investigates the effect of levy flight and random search operation in Cuckoo search in order to ameliorate the performance of the particle swarm optimization algorithm. The two improved particle swarm algorithms are firstly tested on a range of 10 standard benchmark functions and then applied to five different cases of economic dispatch problems comprising 6, 13, 15, 40 and 140 generating units.

Key terms: Particle swarm optimization, economic dispatch, swarm intelligence, genetic algorithm, evolutionary algorithm, bat algorithm, Cuckoo search algorithm, power systems, levy flight, random search, and thermal power plant.

To my mothers, maman

Rosaline NANA, Denise TODA and Olive SEDJOUNG

For their unconditional love

ACKNOWLEDGEMENTS

I would like to use this opportunity to thank all those who have contributed to the success of this project:

- My supervisor, Professor Zenghui WANG for his guidance, encouragement, constructive criticism, suggestions, mentoring and supervision through every stage of the research

- The TCHAPDA family, The Young Adults Family and all my friends for their constant support

- My father Joseph TCHAPDA for his protection, forgiveness, love and prayers

Above all I thank the almighty GOD for his blessing and favour in my life.

LIST OF PUBLICATIONS

[1] G.Y. GNINKEU TCHAPDA[1], Z. WANG[1], Y. SUN[2], "application of improved particle swarm optimization in economic dispatch of power system*", 10th International Symposium on computational Intelligence and design(ISCID)*, Zhejiang China, Vol.2, pp. 500-503, 9-10 December 2017.

[2] G.Y. GNINKEU TCHAPDA, Z. WANG, "Improved particle swarm optimization based on cuckoo search operations and its application*", 2nd International conference on Robotics and Automation Engineering (ICRAE)* Shanghai China, December 29-31, 2017.

[3] G.Y. GNINKEU TCHAPDA, Z. WANG," Cuckoo search operation and Bat algorithm to improve particle swarm optimization and its application", *Manuscript submitted for review and consideration for publication.*

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS, ACRONYMS AND NOMENCLATURE

PSO: Particle Swarm Optimization

BA: Bat Algorithm

CSA: Cuckoo Search Algorithm

ED: Economic Dispatch

ELD: Economic Load Dispatch

DED: Dynamic Economic Dispatch

GA: Genetic Algorithm

DE: Differential Evolution

POZ: Prohibited Operating Zone

SI: Swarm Intelligence

Rad: Radian

MW: Megawatt

Btu/hr: British thermal unit per hour

LIST OF NOMENCLATURE

$\mathbf{CO_2}$**:** Carbon Dioxide

$: US Dollars

Rad: radian

Rand: random number between (0, 1)

$\rightarrow \mathbf{0}$: Very close to 0

R: the set of real number

K: iteration index

$\boldsymbol{N_g}$: Total number of generating units

$\boldsymbol{F_j}$ $(\boldsymbol{P_j})$**:** fuel cost of the j^{th} generating unit in $/hr

$\boldsymbol{P_j}$: power generated by the j^{th} generating unit in MW

$\boldsymbol{a_j}$**,** $\boldsymbol{b_j}$**,** $\boldsymbol{c_j}$: cost coefficients of j^{th} generating unit in $\$/MW^2$, $/MW and $

$\boldsymbol{e_j}$ **and** $\boldsymbol{f_j}$: constants of the valve point effect of generators

$\boldsymbol{P_j}$**,** $\boldsymbol{P_k}$: real power injection at the j^{th} and k^{th} buses, respectively

$\boldsymbol{B_{jk}}$: loss coefficients or B coefficients.

$\boldsymbol{P_j^{min}}$ **and** $\boldsymbol{P_j^{max}}$: minimum and maximum power generation limits of the j^{th} generating unit.

$\boldsymbol{P_j}$**,** $\boldsymbol{P_j^o}$: current and the previous power generation of unit j respectively

$\boldsymbol{DR_j}$ **,** $\boldsymbol{UR_j}$: down-ramp and up-ramp limits of the jth generator in MW/h

$\boldsymbol{P_j^{min}}$, $\boldsymbol{P_j^{max}}$:minimal and maximal power generation of unit j

$\boldsymbol{P_{j,k}^{lower}}$, $\boldsymbol{P_{j,k}^{upper}}$:upper and lower limit of the power generation corresponding to the k prohibited operating zone.

$\boldsymbol{c_1}$, $\boldsymbol{c_2}$: accélération coefficients

W: inertia weight between $w_{min} = 0.4$ and w_{max} =0.9.

$\boldsymbol{r_1}$, $\boldsymbol{r_2}$: randomly generated numbers with a range of [0,1];

$\boldsymbol{p\_b_i^k}$: best position particle i has achieved based on its own experience;

$\boldsymbol{g\_b^k}$: best particle position based on overall swarm's experience;

$\boldsymbol{f^i}$: Frequency of i bat

CHAPTER 1: INTRODUCTION

In power system, Economic dispatch helps to find the optimal output power of a number of generating units that satisfy the system load demand, at the cheapest possible cost, for a short period considering equality and inequality constraints [1, 2]. This is an essential optimization challenge in electric power system. It aims to minimize as much as possible the total cost of production of electricity in consideration of the physical constraints of their generators [3]. The research project is divided into two main points which are: Particle Swarm Optimization (PSO) and Economic Dispatch (ED).

This chapter highlights all the different parts that will be involved in the research. The chapter is structured as follows: the first section gives the background of the study and state the problem that need to be addressed, then the next section develops the objectives of the research followed by the different questions that need to be answered at the end of the study and the research approach. Finally the last section briefly presents the different chapters that constitute the dissertation.

1.1. Background

Every country that wants to boost its economy and its industries needs sufficient energy and lower electricity cost. Due to the shortage, it is important to well manage the power generated by the power station. That means to reduce the costs related to the production and the transmission of electricity from the generating stations to the end users: it is called economic dispatch of power systems [3].Thus ED distributes the real power produced by different power stations in the most economical way possible. So the objective is to optimize the cost of production. There are two types of economic dispatch corresponding to the type of constraints: static and dynamic. The static economic dispatch problem is about reducing the fuel cost of the power generators for a specific period of operation usually one hour while the Dynamic Economic Dispatch (DED) problem follows the characteristics of the hourly dispatch problem. As the power demand varies with each hour and the power generation scheduled for 24 hours is to be determined, therefore the dimension of DED problem can be considered 24 times that of the static ED problem [20].

Economic dispatch is an optimization challenge. There are two groups of optimization methods. The first are the conventional methods also called classical methods. They include linear programming, base- point and participation factors, lambda iteration method [4], gradient method [5], branch and bound [6], quadratic programming, optimal power flow and other mathematical expression with loss. These methods are for continuous and smooth objective functions [7].The second are non- classical, also called metaheuristic methods. They are popular for their capabilities to deal with the nonsmooth and discontinuous objective functions that practical ED problems encounter. They can be classified in many categories such as evolutionary algorithms like Differential Evolution (DE) [8]; immune algorithms like Genetic Algorithm (GA) [9]; and Swarm Intelligence (SI) algorithms [10]. SI techniques have emerged as popular tools to solve ED. They mimic a collective behaviour of nature such as insects, birds, and fishes which normally are decentralized and self-organized [12]. SI techniques include Particle Swarm Optimization (PSO), firefly algorithm [11], bat algorithm [7], differential search [12], artificial bee colony algorithm [13], and cuckoo search algorithm [14],

Kennedy and Eberhart firstly presented PSO as a new heuristic method in 1995 [14]. They studied the social behaviours of bird flock and fish schools [15, 16]. PSO has many advantages as an optimization technique. Firstly, it does not derivate from any other algorithm unlike many classical techniques [1]. Secondly, its flexibility allows it to be associated with other optimization techniques to form hybrid tools. Moreover, PSO is a simple algorithm that easily adapts with different types of optimization problems thus is effective for optimizing a wide range of functions [15]. It is less sensitive to the nature of the objective function that is convexity or continuity. Furthermore, PSO is[a] simple concept, easy to implement with few lines of computer codes. It is programmed with basic mathematical operations. It requires only primitive mathematical operators and is computationally inexpensive in term of both memory requirement and speed [16]. Beside, PSO can generate high quality solutions within shorter calculation time and stable convergence characteristic[s] than other stochastic methods [17]. It has memory. PSO is effective in performing difficult optimization tasks at a cheap computational cost [18]. It has less parameter[s] to adjust unlike many other competing evolutionary techniques. It is able to escape local minima. Another advantage is its ability to handle objective functions with stochastic nature, like in the case of representing one of the optimization variables as random. Finally, PSO does not require a good initial solution to start its iteration process.

1.2. Problem Statement

Despite all the assets enumerated, PSO technique has some drawbacks. For example it has a high time of calculation and it is difficult to set the parameters of the method when it comes to ED problems [3]. Premature convergence and poor fine tuning of the final solutions are also some disadvantages that PSO has to tackle [17- 19]. Above all, PSO is less competitive. That is, some techniques such as genetic algorithm and differential evolution have been improved these recent years and have become more powerful than the PSO algorithm. The main problem that the research project intends to solve is therefore to try to correct those drawbacks of PSO in order to improve its performance and make it a more competitive optimization algorithm applied to the electrical economic load dispatch. Improving the performance of PSO means to correct some drawbacks such as high time calculation, fine tuning of the final solution and premature convergence in order to have better and more competitive solution when applied to the ED problems.

1.3. Research objectives

1.3.1. Improve the performance of the PSO algorithm

There are many ways by which the performance of an optimization algorithm such as particle swarm optimization can be ameliorated. The improvement can be done through the analysis of the topological structure of the algorithm. It can also be done via the modification of its parameters. Moreover the studies of the different constraints that influence the objective function. Another possibility is by associating the optimization technique with other techniques in the way of boosting its performance.

An advantage of PSO is its ability to be combined with other optimization techniques to form hybrid tools. The study will look at the way to change the formulas of PSO through combining it with recently developed SI algorithms such as Bat algorithm [7] and Cuckoo algorithm [36]. So the research is not limited only on combining PSO with other optimization techniques to improve its performance but also to look at possibilities to analyse its structures or to modify the parameters or else to solve the constraints related to the economic dispatch problem.

1.3.2. Use the improved PSO to solve the ED problem.

The second stage of the research is the application of the improved PSO to solve ED problems. The successful theories obtained to enhance the performance of the PSO algorithm will be applied to five cases on ED problems according to the number of generating units and the constraints included. The study will test the algorithm on 6, 13, 15, 40 and 140 generating units.

1.4. Research questions

The main research question for this study is:

How to improve the optimization performance of the PSO?

In order to answer that main question, two sub-research questions need to be answered. The primary research questions are:

- Which technical approach can be used to improve the PSO performance?
- Can PSO combined with another optimization technique be more efficient than improving it alone?

The secondary research question is:

- How to make the improved PSOs more suitable for solving the ED problems?

1.5. Delimitation of the study

The research is limited to the static economic dispatch problem. Thus, economic load dispatch here refers to the static economic dispatch. Pollution of the environment is also an important constraint which affects the production cost when included. It would not be taken into account here since it is not related to the power generation cost. It could have been included if pollutant emission objective function[s] were considered. The power generation considered is the thermal power plant. All the constraints and parameters considered for the ED problems are those provided by the CEC 2011 competition [20]. The experimental research will be tested through standard benchmark functions internationally accepted and broadly used for optimization algorithms. The improvement of the performance of the algorithm will be verified through five characteristics values obtained: best, mean, median, worst, and standard deviation.

1.6. Benefits of the study

The entire study is based on the improvement of PSO algorithm in order to find a better solution to the electrical economic dispatch which is of great importance considering the high price of electricity in the world in general and in South Africa in particular. Solving the ED problem means lowering the cost of power generation and thus the price of electricity to the end users.

Moreover**,** PSO belongs to the field of artificial intelligence which is seen as the fourth revolution. With the development of more powerful computers and the easy collection of data, evolutionary algorithms such as PSO will find a broader area of application and today they are already applied in many areas of science, engineering and technologies. Its improvement might have a remarkable impact on people’s daily life.

Furthermore, PSO is a general purpose optimizer, which is able to tackle wide ranges of optimization problems because it can easily adapt to suit various types of optimization problems with minor modification.

1.7. Structure of the dissertation

This section gives a brief description of the content of all the different chapters that constitute this dissertation report.

Chapter 1: Introduction

The background and problem statement of the project are described in this chapter as well as, the research questions, the aim and objectives, the scope of the project, the approach and methodology to be followed.

Chapter 2: Literature study

This chapter reviews the literatures around economic dispatch of power system, the formulation of the objective function as well as the description of the different constraints that constitute a practical system. Another literature is done on particle swarm optimization from its first introduction to the different improvement until now.

Chapter 3: Research design and methodology

This chapter describes the methodology used to carry out the research. It defines aims and objectives, and develops the research designs with the data used for the analysis as well as the software and the benchmark functions applied for the experimentation.

Chapter 4: Hybrid PSO-BA applied to the economic dispatch problem

The chapter presents a new idea based on the association of PSO with bat algorithm in order to improve the PSO performance. The new algorithm obtained is tested first on some standard benchmark functions and then applied to the economic dispatch problem.

Chapter 5: Improved particle swarm optimization based on cuckoo search operations

The chapter studies the impact of the levy flight and random research operations in cuckoo search and the advantages that they may have in the improvement of the PSO algorithm. The new idea is also tested on benchmark functions and then applied to economic dispatch problems.

Chapter 6: Conclusion and future work

This chapter draws up the conclusion of the dissertation and the future works that can be done. Recommendations are also proposed for further studies and researches in the field.

CHAPTER 2: LITERATURE REVIEW

To realize the sustainable developments, the plants must have efficient management/schedule plan and control methods from the aspects of economy, environment and social community. Optimization algorithms are necessary to obtain the optimal or sub-optimal solutions for the plants. As one of the most important systems, the power systems are becoming more and more complicated and there are many optimization problems in power system. Most of the generators consume the natural resources and it is important that the generators do not waste the natural resources and meet the needs of the society. Hence it is important to get an optimal or acceptable sub-optimal Economic Dispatch (ED). This chapter studies and presents the literature on an important optimization technique: Particle Swarm Optimization (PSO). The first section describes the ED considering a thermal power plant. And the next one presents PSO algorithm as a tool to solve ED problems.

2.1. Economic Load Dispatch

There are different types of systems to generate electricity such as hydroelectric power systems which change the hydraulic energy into electricity; thermal power systems which use the heat from nuclear, charcoal or gas ; and also renewable energy system also called green energy: this is the energy from the sun, the wind or animals residues to produce electricity. The research logic considers thermal power plant as the power generating systems. This section describes a thermal power plant and the economic dispatch problem formulation.

2.1.1. Description of a Thermal power plant

Electrical power is produced in a power station through alternators which drive generators. It is commonly generated through thermal power plant which constitutes about 80% of power plants in the world before hydroelectric plant, and nowadays through renewable energy such as solar panel, wind turbine, and biomasses. It is also the case with South Africa which generates most of its energy via thermal power plants using coal fired.

A thermal power plant is a plant where steam is used to drive a steam turbine. It uses water to generate power [4]. Water is heated at the boiler and circulated with energy to be expanded at the steam turbine to give work to the rotor shaft of the generator. After it passes through the turbine, the steam is condensed in a condenser using an ambient heat sink. The water is then pumped to feed the boiler where it was heated. Steam boiler has the advantage that it can be built for almost any type of combustible fuel.

For an easy understanding, a thermal power plant can be modelled as a transfer function of energy conversion from fuel usually nuclear, charcoal, fuel oils, gas, and diesel in a small level, to produce electricity. The thermal unit system usually consists of the boiler. Fuel is the input and the output is the steam, the steam turbine, and the generator .The input of the turbine-generator unit is the volume of steam and the output is the electrical power, the overall input-output characteristic of the entire generating unit is the direct continuity from the input-output characteristic of the boiler to the input-output characteristic of the turbine-generator unit. The figure 2.1 describes the energy conversion of a thermal power plant.

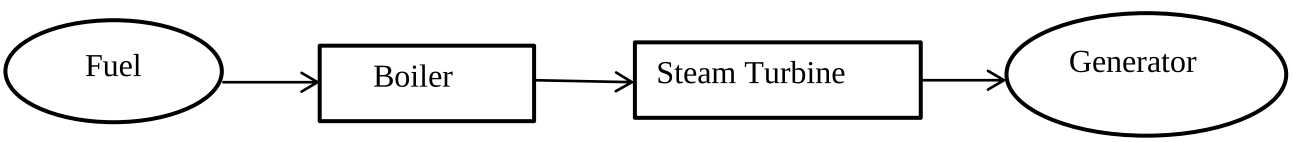

Figure 2.1: Thermal plant energy generation process

A thermal plant input power is generally measured in British thermal unit per hour (Btu/hr.) while the output which is the active power, is in megawatt (MW). The purpose is to find the generation of every plant such that total operating cost is minimum. That is the economic dispatch of power systems.

2.1.2. Economic dispatch: Problem Formulation

The Economy Load Dispatch (ELD) problem studies how the real power output of each controlled generating unit in an area is chosen to meet a given load demand and to minimize the total operating cost in that area. In power system, ELD in general terms deals with the minimization of the total generation cost while meeting the equality and inequality constraints imposed by the power generating units [8]. The generation cost makes the load dispatch problem non- linear and constrained optimization problem. Thus, ELD attempts to schedule the generation with capability of minimizing the total operating cost and operate under unit operating limits [20]. It has therefore the objective to minimize the total cost of generation while honouring the operational constraints of the available generation resources. ED includes a cost function employed with various Power balance and inequality constraints. The cost function considered and constraints which are taken into account for the purpose of formulating the ELD problem. The study considers thermal power plant generating units. All the constraints included are those that are found in a practical thermal power plant generating units.

2.1.2.1. Objective function

The objective function that minimizes the total cost function of the generating units is given by [7]:

$$\min_{\mathrm{P} \in \mathrm{R}^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g} (a_j\, P_j^2 + b_j P_j + c_j) \tag{2.1}$$

where R is the set of real number and N_g is the total number of generating units,

$F_j\ (P_j)$ is the fuel cost of the j^{th} generating unit in \$/hr,

P_j is the power generated by the j^{th} generating unit in MW,

a_j, $b_j$and c_j are cost coefficients of j^{th} generating unit in \$/MW^2, \$/MW and \$ respectively.

An important constraint that influences the total cost is the valve-point effect. The valve point effect is due to the fact that the actual cost curve function of a large thermal plant is discontinuous and nonlinear [9]. This is because the valve-opening process creates a wave- like effect in the heat rate curve of the generators. This opening will increase the fuel cost exponentially due to the wire drawing effect [11, 13]. When the valve- point effect is considered, sinusoidal term is included to the quadratic cost function. The objective function therefore becomes:

$$\min_{\mathrm{P} \in \mathrm{R}^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g} (a_j P_j^2 + b_j P_j + c_j)$$

$$= \sum_{j=1}^{N_g} (a_j P_j^2 + b_j P_j + c_j) + \left| e_j \sin(f_j (P_j^{min} - P_j)) \right| \qquad (2.2)$$

where e_j and f_j are constants of the valve point effect of generators in $ and rad/MW respectively.

This ELD problem is subjected to different type of constraints: optimization constraints and practical operating constraints of generators.

2.1.2.2. Optimization constraints

Optimization constraints are constraints related to the production of power. They include the real power balance constraints and the generator limits.

- **Power balance constraints**

They balance the power between total system generation (P_D) and total system losses (P_L). When the transmission losses of a unit are high, the ED solution may require the unit to decrease its output, while other units with lower transmission losses increase their outputs.

When line or transmission losses are included in the ED problem, the formulation of the power generation is as follow,

$$\sum_{j=1}^{N_g} P_j = P_D + P_L \qquad (2.3)$$

where P_L is calculated using B-coefficients, given by

$$P_L = \sum_{j=1}^{N_g} \sum_{k=1}^{N_g} P_j \; B_{jk} \, P_k + \sum_{j=1}^{N_g} B_{oj} \, P_j + B_{oo} \qquad (2.4)$$

Here P_j and P_k are the real power injection at the j^{th} and k^{th} buses, respectively, and B_{jk} is the loss coefficients or B coefficients. The B coefficients are not really constant, but vary with unit loading. However they are presumed to be constant under standard operating conditions.

- **Generator constraints**

The output power of each generating unit should lie in between its lower and upper bounds. Each generating unit must not operate above its rating or below some minimum capacity. The generator constraint is represented as follows.

$$P_j^{min} \leq P_j \leq P_j^{max} \qquad j= 1, 2, 3 \ldots N_g \qquad (2.5)$$

where P_j^{min} and P_j^{max} are the minimum and maximum power generation limits of the j^{th} generating unit.

Other inequality constraints can also be included in the ED problem such as the restriction of some unit output so that other apparatus are not overloaded. Another example is under adverse weather conditions, generation at some units may be limited to reduce emissions.

2.1.2.3. Practical operating constraints of generators

The Economy Dispatch problem is usually expressed as a quadratic programming function that can easily be solved with classical or gradient based methods. However, realistic ED problems are based on the nonlinear characteristics of generators which comprise ramp rate limits, the discontinuous prohibited operating zones, and valve point effect which are nonconvex [17]. Metaheuristic methods such as PSO are used to solve this type of problem.

❖ Ramp rate limits

Ramp rate limits are due to the physical limitation of generators to start up and shut down. The ramp rate limits constraint the rate at which the output power level of a given generator can be modified from one step t_{z-1} to another step t_z. The power generated may increase or decrease depending on the upper and downward ramp rate limits [18]. The model is as follow,

If power generation increases, $P_j - P_j^o \leq UR_j$ (2.6)

If power generation decreases, $P_j^o - P_j \leq DR_j$ (2.7)

P_j And P_j^o are the current and the previous power generation of unit j respectively. DR_j and UR_j are respectively the down-ramp and up-ramp limits of the jth generator in MW/h. P_j^{min} and P_j^{max} are the minimal and maximal power generation of unit j. Therefore, the operating constraints of generators with the ramp rate limit can be formulated as:

$$\max(P_j^{min}, UR_j - P_j^o) \leq P_j \leq \min(P_j^{max}, P_j^o - DR_j) \quad (2.8)$$

❖ Prohibited operating zones

The generator may have particular area where operations are limited due to physical failure of machine components or instability. Steam valve operation or vibration in shaft bearings causes the Prohibited Operating Zones (POZ) in the input – output curve of the generating unit [17, 18]. Actual performance testing or operating records are not efficient to find the POZ, thus the best economy is realized by avoiding operations in areas that are in actual operation. The feasible operating zones of j^{th} generator are described in the equation below,

$$\begin{cases} P_j^{min} \leq P_j \leq P_{j,1}^{lower} \\ P_{j,k-1}^{upper} \leq P_j \leq P_{j,k}^{lower} \\ P_{j,PZ_j}^{upper} \leq P_j \leq P_j^{max} \end{cases} \quad , k= 2, 3,\ldots, \; N_{pz} \quad (2.9)$$

P_j^{min} and P_j^{max} are the minimal and maximal power generation of unit j. $P_{j,k}^{lower}$ and $P_{j,k}^{upper}$ are the upper and lower limit of the power generation corresponding to the k prohibited operating zone.

Most constraints enumerated above need to be taken into consideration to solve a realistic ED problem. When they are all combined, the objective function of the ED problem becomes:

$$\min_{P \in R^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g} (a_j P_j^2 + b_j P_j + C_j)$$

$$= \sum_{j=1}^{N_g} (a_j P_j^2 + b_j P_j + C_j) + \left| e_j \sin(f_j (P_j^{min} - P_j) \right| \quad (2.10)$$

Subject to. $\sum_{j=1}^{N_g} P_j = P_D + P_L$

$$\max(P_j^{min}, P_j^0 - DR_j) \leq P_j \leq P_{j,1}^l$$

$$P_{j,k-1}^U \leq P_j \leq P_{j,k}^l, k = 2,3, \dots, n_j, j = 1,2, \dots N_g \quad (2.11)$$

$$P_{j,n_j}^U \leq P_j \leq \min(P_j^{max}, P_j^0 + UR_j)$$

Particle swarm optimization is a nature based optimization algorithm that has been broadly used to solve nonlinear and nonconvex functions. It has been applied in many areas of engineering including economic dispatch of power system. The next section introduces and describes the PSO algorithm as the metaheuristic technique to solve ED problems.

2.2. Other optimization techniques successfully applied in Economy Dispatch

To solve the ED problem, PSO is challenged by strong evolutionary methods such as Differential Evolution (DE) and Genetic Algorithm (GA) which use a completely different approach than PSO do. DE instead uses a greedy and less stochastic method [25, 26]. GA on the other hand belongs to the largest class of evolutionary algorithm (EA), which generates solution to optimization problems using techniques inspired by natural evolution such as inheritance, mutation, selection and crossover [9, 27]. GA is known to be slow and sometimes it cannot converge to the global optimal. GA with multi parent crossover (GA-MPC) introduces a new crossover with a randomized operator to replace mutation. The applied crossover uses three parents to generate three new offsprings. Two of them are to help exploitation. And the third offspring is for promoting exploitation. The randomized operator helps to escape from local optima and premature convergence [44].

Apart from them, they are also some recently developed swarm intelligences (SI) algorithms that have been successfully applied to ED problems such as Bat algorithm [7], Cuckoo algorithm [36], firefly algorithm [11], differential search algorithm [12], and artificial bee colony [13]. These SI can be combined with the PSO to improve its performance.

PSO has many advantages as an optimization technique. Firstly, it does not derivate from any other algorithm unlike many classical techniques [1]. Secondly, its flexibility allows it to be associated with other optimization techniques to form hybrid tools. Moreover, PSO is a simple algorithm that easily adapt with different type of optimization problems thus is effective for optimizing a wide range of functions [15]. It is programmed with basic mathematical operations. Beside, PSO can generate high quality solutions within shorter calculation time and stable convergence characteristic than other stochastic methods [17]. It has memory. PSO is effective in performing difficult optimization tasks at a cheap computational cost [18]. It has less parameter to adjust unlike many other competing evolutionary techniques.

2.3. Particle swarm optimization

Since its first development in 1995, Particle Swarm Optimization (PSO) has become one of the most popular nature inspired metaheuristic optimization algorithm [20]. The algorithm uses the behaviour of the bird flock in their search for space or for food. The bird relies on two types of information, its personal position and the position of its neighbour which is the closest to the food. Each particle during the search, adjusts its speed and position according to its own information, and the information received from other particle in the swarm. It thus uses the best position obtained by itself and its neighbours [17]. That is, in the search space, the next position of the particle is guided by its personal experience (pbest), the global experience (gbest) and the present movement of the particles. Pbest is the best position that the particle has achieved so far. It can be seen as the particle’s memory [20]. The particle in PSO is an intelligent element that searches space based on its own experience and the experiences of peer particles: That is the simple concept around the PSO algorithm [21, 22].

2.3.1. PSO formula equation

The particle swarm optimization formula involves only two models equations: position and velocity. The modified velocity and position of each particle can be calculated using the current velocity and distance from pbest to gbest as shown:

$$v_i^{k+1} = wv_i^k + c_1 r_1 \left(p\_b_i^k - x_i^k\right) + c_2 r_2 (g\_b^k - x_i^k) \tag{2.12}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2.13}$$

c_1 and c_2 are two positive constants also called acceleration coefficients; they are usually between 2 and 2.05. In the case of the experimentation, they will be chosen c_1 = 2 and c_2 = 2.

k is the iteration index; w is the inertia weight between $w_{min} = 0.4$ and w_{max} =0.9. The inertia weight improves consequently the PSO performance.

r_1 and r_2 are two randomly generated numbers with a range of [0,1];

$p\_b_i^k$ is the best position particle i has achieved based on its own experience;

$$p\_b_i^k = \left[x_{i1}^{p\_b},\ x_{i2}^{p\_b}, \dots, x_{iN}^{p\_b}\right];$$

g_b^k is the best particle position based on overall swarm's experience;

$$g\_b^k = \left[x_1^{g\_b}, x_2^{g\_b}, \dots, x_N^{g\_b}\right];$$

PSO is a simple algorithm and easy to apply; this does not prevent it to be a robust tool in solving optimization problems. Since its first introduction in 1995, lots of researches have been done in other to improve the PSO performance. It also has been combined with other optimization technique to form a hybrid tool. This has results in many variant of PSO algorithms such as: Standard PSO, Modified PSO [25], Evolutionary PSO [22], Chaotic PSO [29], Self-organizing hierarchical PSO [34], PSO with time-varying acceleration coefficients [37], Random drift PSO [18], Quantum-inspired PSO [38], Anti-predatory PSO [39], multi swarm PSO[59], etc. Figure 2.2 shows the search mechanism of PSO in multidimensional search space. It helps to understand how the different elements that constitute the formula are combined to update the position and the velocity of the particle [27].

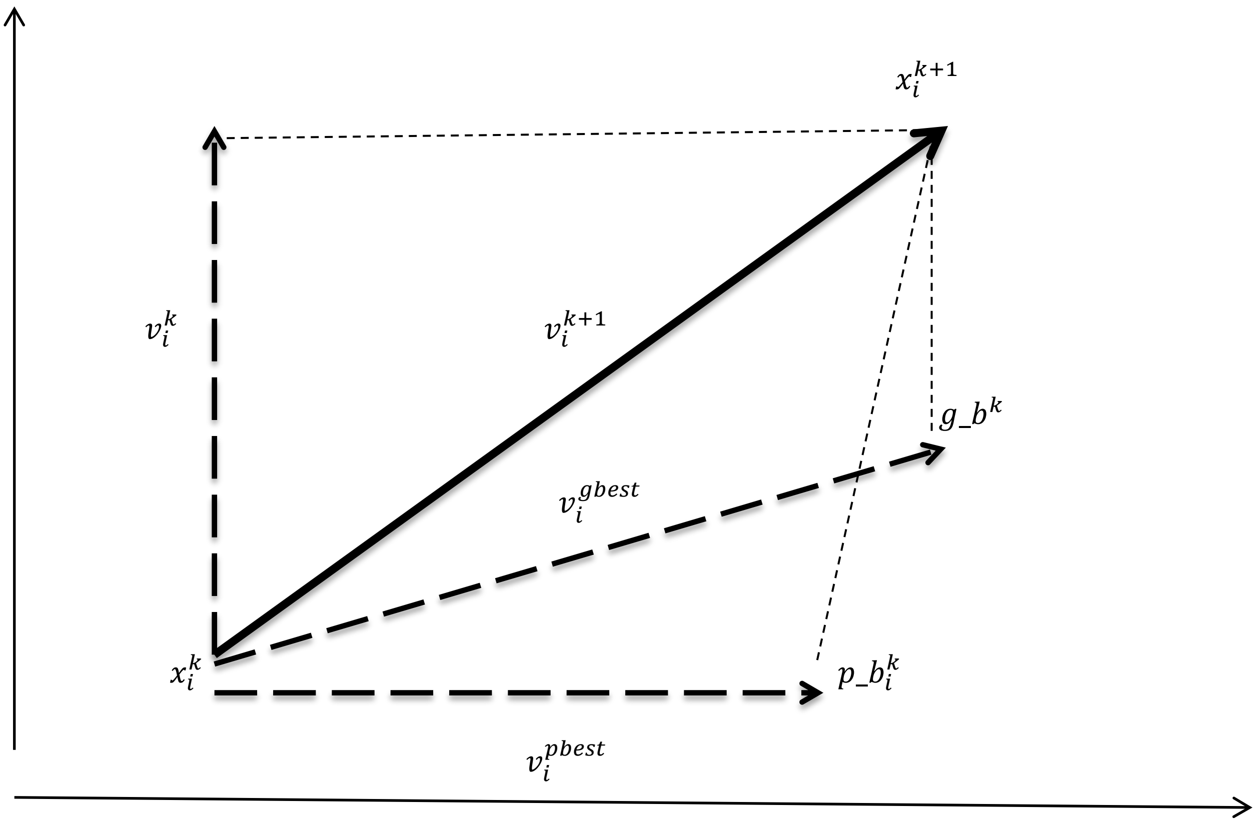

Figure 2.2: PSO search mechanism in multidimensional search space

2.3.2. PSO Algorithm

It is to be noted that the equation of PSO given above is called equation of the "modified PSO" which includes the inertia weight w, on the velocity equation (equation 2.12). Initially, PSO algorithm does not include it. Research has shown that the use of the inertia weight improves consequently the PSO performance [25]. PSO algorithm is generally presented as follows [1, 2]:

1) For each particle, initialize randomly the position and velocity vectors with the same size as the problem dimension

2) Measure the fitness of each particle (pbest) and store the particle with the best fitness value (g_b)

3) Update velocity and position vectors according to (2.12) and (2.13) for each particle

4) Repeat steps 2-3 until a termination criterion is satisfied

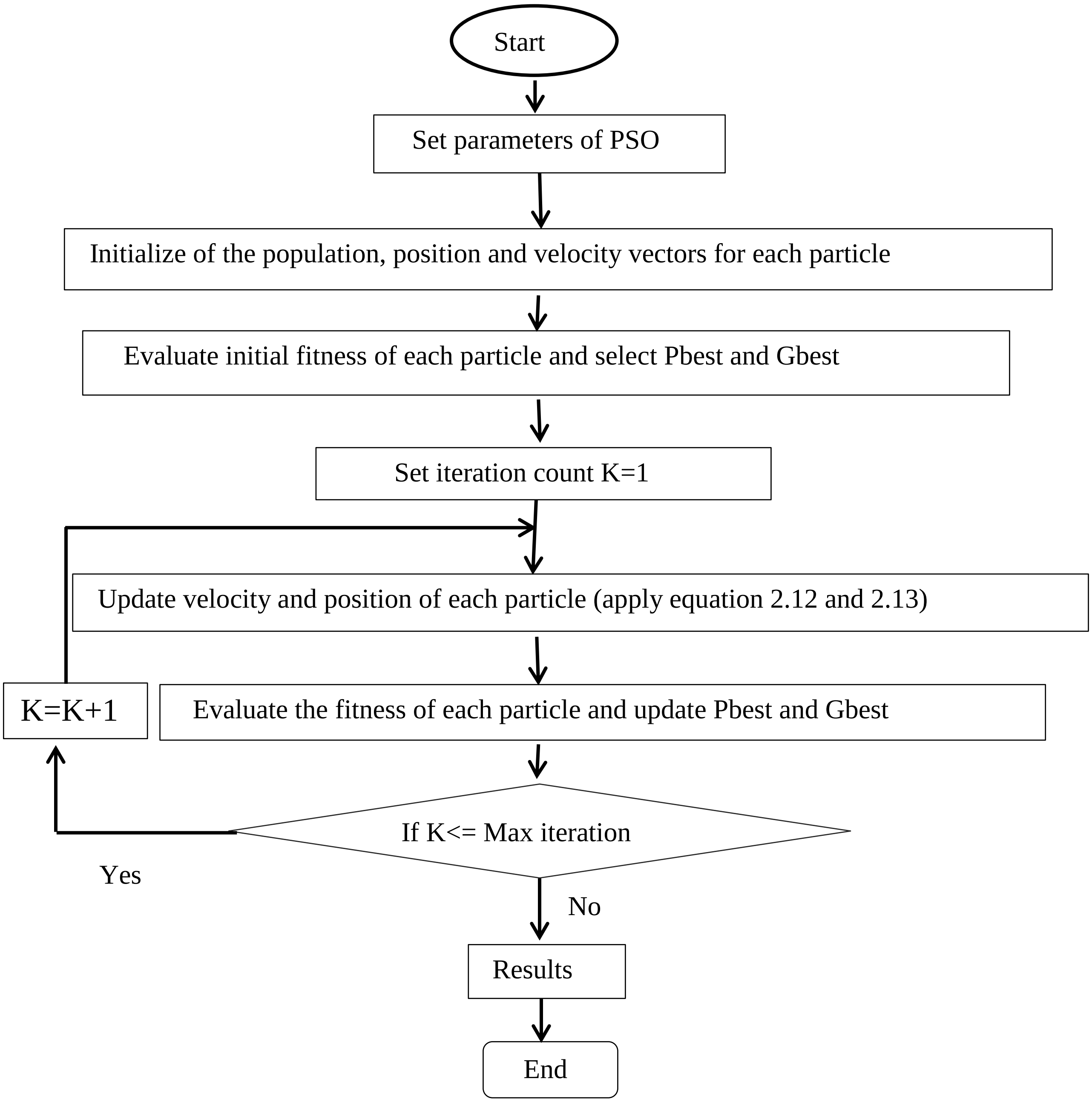

Figure 2.3: Flow chart for PSO algorithm

The detailed flow chart helps to better understand the PSO algorithm. The study aims to improve the algorithm in the purpose of applying it to solve the ED of power system. With the always increasing need of power, the competitive market and the growing of the commodity material, added to the fact that power cannot be stored at a high level, it is essential that the power produced must be used in its optimal capability: thus the importance of economic dispatch in power system. The PSO code modified and adapted to address this issue is proposed in [27].

In thermal power generating firm which is the case study, ED means to minimize the fuel cost for power generation through proper dispatch schedule. That is the firm need to reduce fuel cost, for better profit at the same time it should satisfy system load demand, reactive power limit, voltage limit, power transmission limit and other limitations [26]. To achieve that, the firm has to spend money which affects the power generating profit [28, 29]. Therefore the two opposite objectives should be compromised to find the firm's optimal profit [30, 31]. So the objective of an economic load dispatch here is to find the optimal combination of power generation that minimizes the total generation cost while satisfying quality and inequality constraints [29, 30]. Furthermore, DED involves more parameters than the static one. This is a complex optimization situation. Its importance may increase with the growth in power generation competition [32- 35].

2.3.3. Particle swarm optimization parameters

The choice of the parameters should be in accordance with those of the CEC 2011 since they provide data that will be used for experimentation. The results should provide the best, mean, median, worst and Standard Deviation (S.D) of the objective function values obtained over 25 independent runs. The algorithm should be runs for 150,000 times [20]. Parameters should also be initialized randomly. PSO parameters are as followed:

- number of variables is the number of generating units for each case that is 6, 13, 15, 40 and 140
- population size is 100
- inertia weight max is 0.9, min is 0.4

- acceleration factor c1=2 and c2= 2

2.4. Research design

This section describes the approach used to carry out the research. It then presents the data and the software used and finally the benchmark functions applied during experimentations.

2.4.1. Research approach

A research design is a master plan which specifies the methods and procedures used to guide and carry out a research. It is a strategic plan for a research project. It ensures that the evidence obtained allows the researcher to effectively address the research problem, obtains relevant information to the research problem, specifies the type of evidences needed to test a theory, evaluates a program to accurately describe and assess data.

The research design is generally divided into qualitative and quantitative approaches. Qualitative research consists of a series of differently developed techniques used to address questions of a particular interest by employing a methodology which includes observation, interviews and data analysis. On the other hand, quantitative approach is a formal, objective and systematic process to obtain quantifiable information about the research, presented in numerical form and analysed through the use of statistics [21]. Quantitative analysis is primarily a measuring process since it provides essential connection between empirical observation and mathematical expression of quantitative relationship. The research project is a mixture of qualitative and quantitative approach project, based on the improvement of an optimization algorithm in order to apply it to different economic dispatch problems.

2.4.2. Data availability

There are many sources that can provide reliable data to test the results of the study. The algorithm can be tested through the Institute of Electrical and Electronic Engineers (IEEE) database or from ESKOM database, ESKOM is the South African public provider and distributor of power. There are also some famous international events such as CEC 2011 competition on testing evolutionary algorithms on real world optimization problem that provide data system to simulate different economy dispatch problems [20]. In order for the research to be more practical, and reduce the difficulties to obtain data from the different organizations mentioned, the algorithm would be tested using data from the CEC 2011 competition.

Moreover, it is good to note that reliable and secure computer software is important in the economic dispatch for the quick response to the system changes and to maintain power system reliability, when choosing the lowest cost of power to dispatch [32]. Matlab is a robust tool with easy ability to plot graph. It is able to do multiple graphs in one. This will be helpful since the research will result in many simulations and comparison of the different plots obtained.

2.4.3. Benchmark functions

When the test involves optimization functions, researches usually compare different algorithms on a large test set such as benchmark functions. Benchmark functions also called test functions are functions that are used to evaluate and to compare the performance of optimization algorithms [46]. They are used in experimental research to evaluate the improvement of the PSO algorithm. Benchmark functions are usually continuous and multi or unimodal. Their functions also are two or multidimensional. Table 2.1 shows a list of 10 common standard benchmark functions with their formula and their initial range that will be applied to analyse the different findings.

Table 2.1: Benchmark functions

Function		**Formula and Initial range**
$f_1(x)$	Sphere	F(x) = $\sum_{i=1}^{D} x_i^2$ -10 ≤ xi ≤ 10
$f_2(x)$	Rastrigin	F(x) = 10n + $\sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$ 10 ≤ xi ≤ 10.
f_3(x)	Ackley 1	F(x)=$-20e^{-0.02}\sqrt{D^{-1}\sum_{i=1}^{D} x_i^2}$ $-20e^{D^{-1}}\sum_{i=1}^{D}\cos(2\pi x_i) + 20 + e$ −35 ≤ xi ≤ 35

Function		Formula and Initial range
f_4(x)	Salomon	F(x) =1cos (2π$\sqrt{\sum_{i=1}^{D} x_i^2}$) +0.1$\sqrt{\sum_{i=1}^{D} x_i^2}$ -100≤xi ≤100
f_5(x)	Rotated hyper-ellipsoid	F(x) = $\sum_{i=1}^{n} \sum_{j=1}^{i} \left[x_j^2\right]$ - 65:536 < xi <65:536
f_6(x)	Alpine 1	F(x) = $\sum_{i=1}^{D} \lvert x_i \ \sin(x_i) + 0.1\, x_i \rvert$ −10 ≤ xi ≤ 10
f_7(x)	Branin	F(x)=$(x_2 - \frac{5.1}{4\pi^2} x_1{}^2 + \frac{5}{\pi}\ x_1 - 6)^2 + 10\left(1 \frac{1}{8\pi}\right)\cos(x_1) + 10$ –π<x_1 <12.275 π<x_2 <2.275
f_8(x)	Griewank	F(x) = $\sum_{i=1}^{n} \frac{x_i^2}{4000} - \Pi \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ −100 ≤ xi ≤ 100.
f_9(x)	Rosenbrock	F(x) = $\sum_{i=1}^{D} [100(x_{i+1-}\, x_i^2)^2 + (x_i - 1)^2$ −30 ≤ xi ≤ 30
f_{10}(x)	Quartic function	sum1 = sum1 + i × x(i)^4 -10 ≤ xi ≤ 10

2.5. Conclusion

This chapter has presented the literature on economic dispatch of power system considering a thermal power plant. Then it has given details about all the parameters and constraints that need to be considered to formulate a practical economic dispatch problem. Thereafter, it has presented the particle swarm optimization as the optimization technique used to solve the economic dispatch problem. Parameters and algorithms have been described and a brief description of other techniques applied to solve the ED problem has been presented. Finally, the research design has highlighted all the different aspect that will intervene in the researches and experimentations in order to improve the particle swarm optimization algorithm. It has presented all the parameters, the source of data used and the software applied to do the simulation. The upcoming chapters focus on the improvement of PSO algorithm. The next chapter first introduces a hybrid technique combining PSO and bat algorithm in order to improve PSO performance. Then chapter four investigates the effects of Levy flight and random search operations in Cuckoo Search algorithm to improve PSO algorithm. Every improved solution will be applied to economic dispatch problems. And finally the two proposed ideas will be compared in order to evaluate the level of improvement of each technique

CHAPTER 3: HYBRID PSO-BA APPLIED TO THE ECONOMIC DISPATCH PROBLEM

This chapter presents a new idea to improve PSO performance based on an association with Bat Algorithm (BA). The new algorithm obtained is tested first on 8 benchmark functions and then applied to the economic dispatch problem. The research methodology to improve the PSO algorithm is first presented then the rest of the chapter is divided as followed: section two presents the Bat algorithm. Then the following section describes and analyses the proposed hybrid PSO-BA. Finally the last section shows the experimental results obtained for the simulation with the benchmark functions first and then with economic dispatch problems.

3.1. Methodology to improve the PSO algorithm

Deep literature around evolutionary algorithms in general and particle swarm optimization in particular has helped to understand that there are many theories on improving the performance of an optimization algorithm. These theories are divided into two categories: standalone improvement and improvement through hybridisation with other algorithm.

Published papers have proposed different theories to improve PSO without combining it with other algorithms. They include: modification of the PSO formulas; solving different constraints; analyse the topological structure of the PSO; and then develop the multi swarm PSO. Likewise, lots of researches have also been published on PSO combined with other algorithms. Hybrid theories have been successful when applied to improve PSO algorithm. Theories on the improvement will be then experimented through simulation using famous benchmark functions considering the best, mean, median, worst and standard deviation criteria. To clearly define each criterion, it is important to situate the context. Optimization mean to find the optimal solution that is the highest or the lowest value possible. In the case of the research study, optimization implies to find the smallest value possible. Thus, "best" means the smallest solution possible while "worst" means the highest solution found. In the same way, "mean" refers to the middle of the two extremes (that are "best" and "worst").While "median" represents the value in the middle of a series of value that is the value in the middle of all the value obtained during simulation. And finally the standard deviation measure the level of change in value of the other criteria numerated. Therefore, the improvement will be appreciated when at least three of the five criteria show a result that has lower values than the current one.

3.1.1. Improved PSO applied to the economic dispatch

The particle swarm optimization has been applied in many areas of science and engineering. In power system, many papers have been published around the application of PSO to solve economic dispatch problems. All the improved algorithms obtained will be applied to ED problems. Five cases will be considered according to the number of generating units and the constraints included (6, 13, 15, 40 and 140 generating units). Comparative analysis will be performed to confirm the improvement of the algorithm. The test will consist to find the minimum value possible for the systems. This value will be observed also through the plot of graphs representing the system.

3.2. The Bat algorithm

Bats are captivating animal. They are the only mammals with wings. There are two types of Bats depending on their size: mega bats and micro bats which use extensively echolocation to detect prey [50]. The bat algorithm is a swarm intelligence algorithm proposed by Xin-She Yang in 2010 inspired by the echolocation behaviour of micro bats [47]. Bats use echolocation to navigate and locate prey. Echolocation works as a type of sonar. Bats, mainly micro bats emit a loud and short pulse of sound, wait for the sound wave to hit into an object and after a fraction of time, the echo returns back to their ears. Thus bats can compute how far they are from the object [40]. Hence, bats are characterized by echolocation behaviour, frequency and loudness. The frequency is sent by the micro-bats with fixed frequency, f_{min} and with variable wavelength, λ. The loudness is used to hunt [48]. In addition, these mechanisms make bats able to distinguish between an obstacle and a prey, allowing them to search even in the darkness [48]. The bat algorithm with its other characteristics is generally described as follow [44]:

Step 1: Calculate the fitness values of all the bats using the objective function F,

Step 2: For i^{th} bat, define pulse frequency f

$$f^{i} = f^{min} + r_1\,(f^{max}\text{-}f^{min}) \tag{4.1}$$

Step 3: Update the velocity and position (which is a vector of generation values)

$$V^{i}(t+1) = V^{i}(t) + f^{i}(X^{i}(t) - X^{best}(t)\,) \tag{4.2}$$

$$X^{i}(t+1) = X^{i}(t) - V^{i}(t+1) \tag{4.3}$$

Step 4: Generate a new solution by random walk

$$X^{i,new}(t) = \begin{cases} X^{best}(t) + r_3 A^{i}(t) & if\ r_2 > R^{i}(t) \\ X^{r}(t) + r_3 A^{i}(t) & else \end{cases} \tag{4.4}$$

Step 5: Select the fitter of the old and new solutions, with a probability $A^{i}(t)$

$$X^{i}(t) = X^{i,new}(t),\ \text{if f}\ (X^{i,new}(t)) < \ \text{f}\ (X^{i}(t))\ \text{and}\ r_4 < A^{i}(t) \tag{4.5}$$

Step 6: Update the values of R^{i} and A^{i} using the respective equation.

$$A^{i}(t+1) = \alpha A^{i}(t) \tag{4.6}$$

$$R^{i}(t+1) = R^{i}(0)[1\text{- exp }(\text{-}\gamma t)] \tag{4.7}$$

Step 7: Check if there is no constraints violation

Step 8: Repeat steps 1 to 7 until the maximum number of iterations is reached.

As PSO, bat algorithm has found many applications in different area of sciences and engineering [40, 45, and 50]. It is also a strong hybrid tools easy to combine with other optimization algorithm.

3.3. Hybrid PSO-BA: proposed idea

The improvement of the PSO algorithm is based on the integration of the BA into the modified version of PSO in order to form a hybrid PSO-BA. The modified PSO uses the inertia weight. The inertia weight introduced at the velocity updating step is responsible for the momentum of each particle. It was proposed by Shi and Eberhart [25] and it works by weighting the involvement of the previous velocity for the purpose of eliminating the requirement for velocity clamping. It impacts positively on the PSO performance.

The idea comes from the consideration of a particle as a micro bat in order to use its technique of echolocation to search for prey. Only two characteristics of micro bat are applied. The first characteristic is the frequency which is sent by the micro bat with a fixed value f_{min} and with a variable wavelength λ [26]. The PSO as the main optimizer uses frequency characteristic of BA in order to boost its velocity update by associating the two. The inertia weight is added on the velocity update equation of BA in order to maintain the homogeneity of the algorithm. Thus the velocity update of the new algorithm is an addition of the velocity update of the PSO and the modified velocity update of BA.

The second characteristic applied is the loudness. Loudness is used to search for preys. PSO uses the loudness of BA in order to handle the boundary violation in case the solution improves. That is to adjust the improved solution by making sure that the sound is not too loud. Loudness also helps to tackle premature convergence which is another drawback of PSO. The enhanced PSO algorithm is as follow:

1) For each particle, initialize randomly the position and velocity vectors
2) Measure the fitness of each particle (pbest) and store the particle with the best fitness (gbest) value
3) Update the ameliorated velocity vectors using the following equations:

$$f^{i} = f^{min} + r_1\,(f^{max} - f^{min})$$

$$v1_i^{k+1} = wv_i^k + f^i\left(x_i^k - p\_b_i^k\right)$$

$$v2_i^{k+1} = wv_i^k + c_1 r_1\left(p\_b_i^k - x_i^k\right) + c_2 r_2\left(g\_b^k - x_i^k\right)$$

$$v_i^{k+1} = (v1_i^{k+1} + v2_i^{k+1})/2 \quad (4.8)$$

4) update the position vector using the PSO equation:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (4.9)$$

5) Evaluate fitness , update pbest if solution improves and not too loud
6) Update Gbest of population.

7) Repeat steps2-6 until a termination criterion is satisfied.

The next paragraph shows the simulation of the improved PSO and the comparison with PSO and BA that have been used to form the hybrid algorithm.

3.4. Experimental results

In order to confirm its improvement, the algorithm has been tested first on nine benchmark functions and then on five different cases of economic dispatch problems. Matlab software version 7.10.0.499 (R2010a), installed on a personal computer with a 2.6GHz processor and 8GB RAM, running on windows 7, is used to program and test the algorithm.

3.4.1. Simulations and analysis

Best, mean, median, worst, and Standard Deviation (S.D.) are the five criteria considered to evaluate and compare the three algorithms. Optimization performance can only be appreciated after a certain number of iterations. In this case study, the simulation is made of 10 independent runs of 100 iterations each. That makes it 1000 iterations which is enough to appreciate any improvement of an algorithm. Figures 4.1-9 depict the convergence characteristic of the PSO (represented in blue colour) and the improved PSO algorithm (represented in red colour). The figures depict a faster convergence characteristic of the hybrid PSO-BA. That is in each of those figures, the red curve is quicker to reach its optimal value than the blue do. That faster convergence of the improved PSO also let see that with a smaller number of iteration, the gap between the two algorithms could have been bigger. Moreover, five figures out of the 9 reach their optimal value in about 50% of the total iteration count. Furthermore, it is good to observe that the curves of the improved PSO are smoother because the direction of the curves has fewer changes in direction during their convergence.

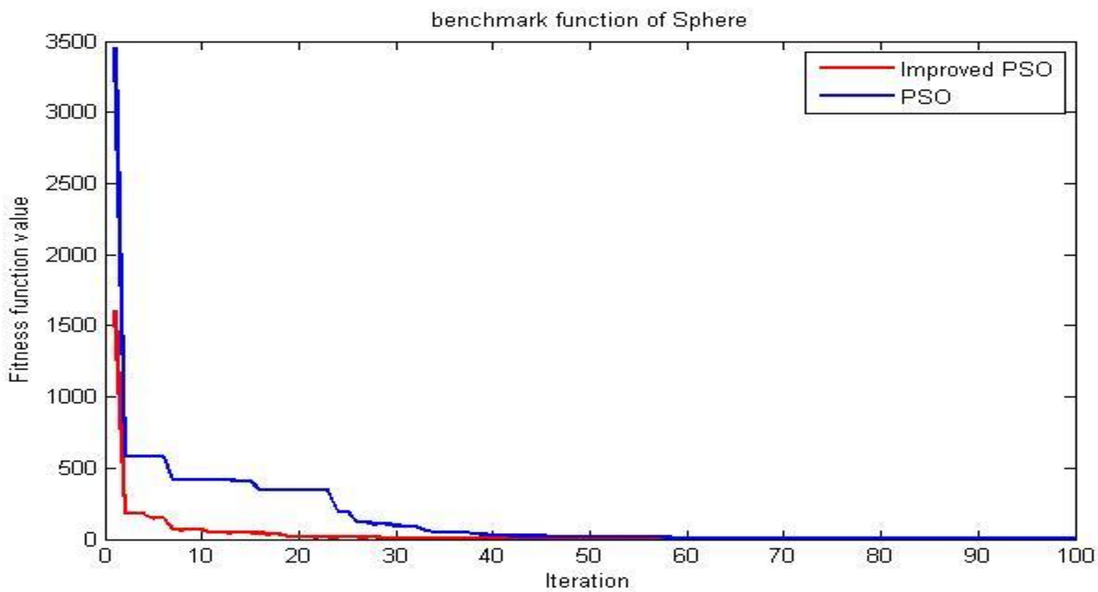

Figure 3.1: Convergence characteristic of PSO and hybrid PSO-BA for Sphere function

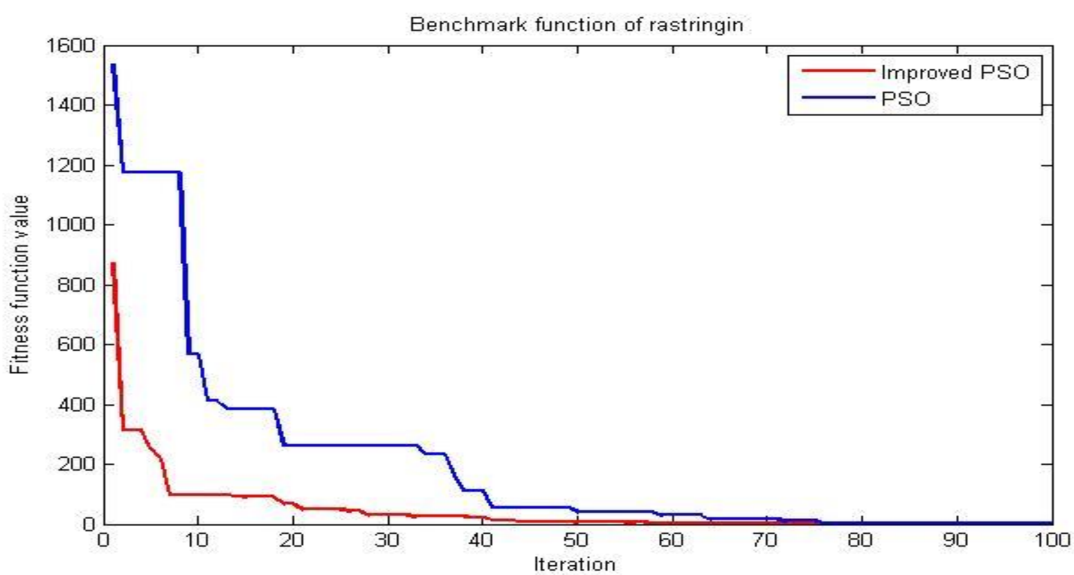

Figure 3.2: Convergence characteristic of PSO and hybrid PSO-BA for Rastringin function

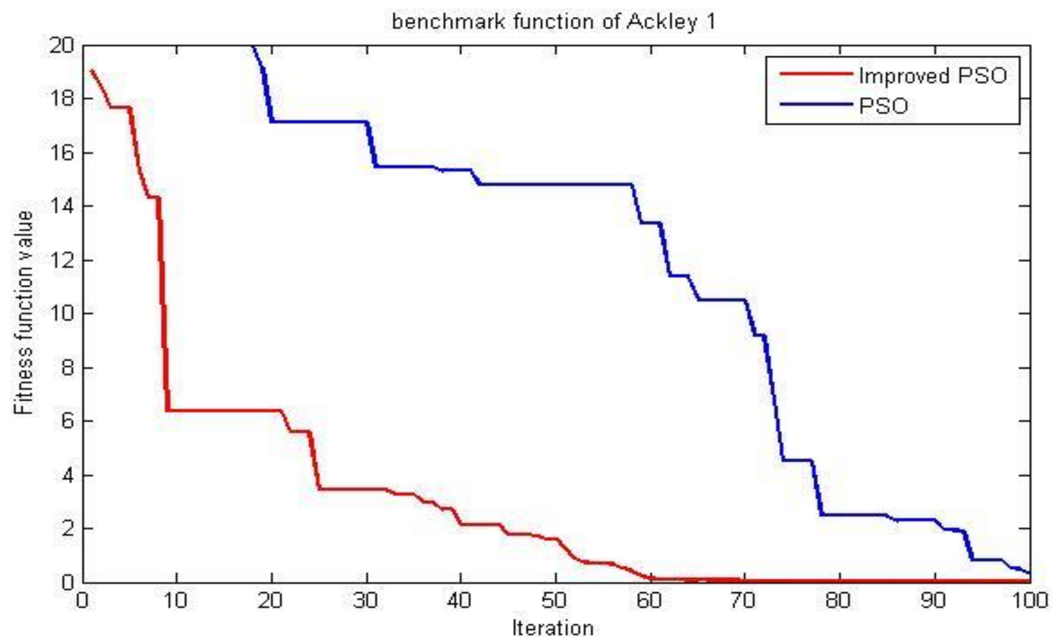

Figure 3.3: Convergence characteristic of PSO and hybrid PSO-BA for Ackley 1 function

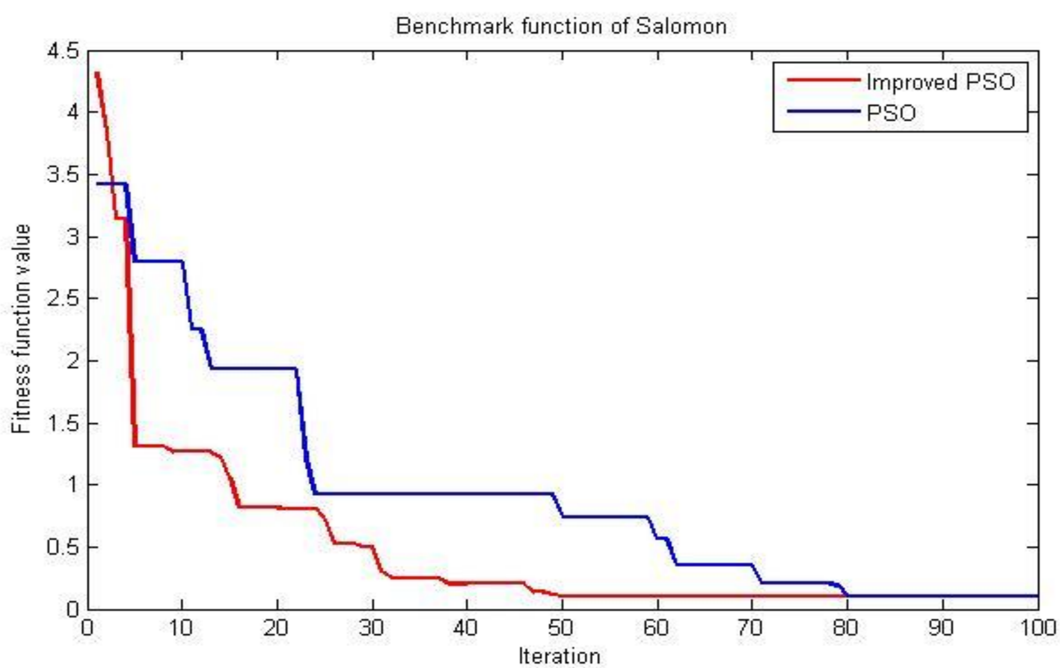

Figure 3.4: Convergence characteristic of PSO and hybrid PSO-BA for Salomon function

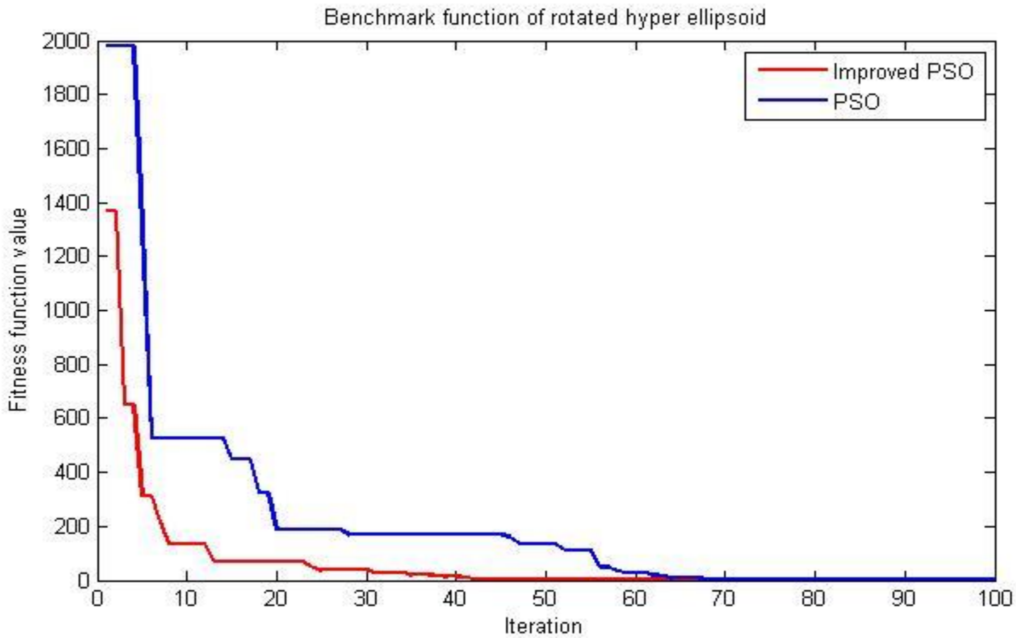

Figure 3.5: Convergence characteristic of PSO and hybrid PSO-BA for Rotated hyper ellipsoid function

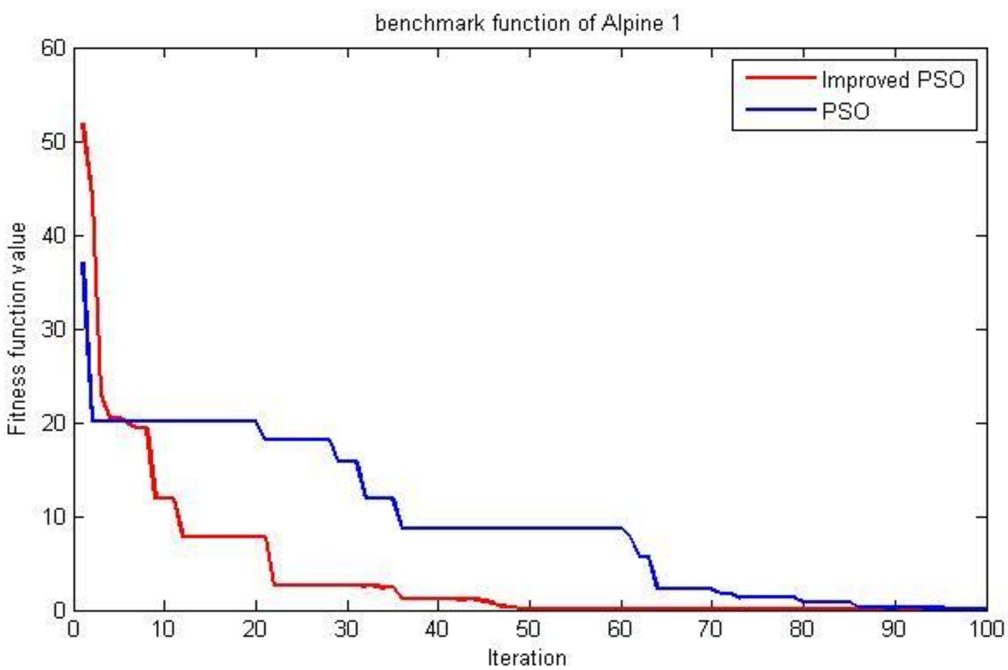

Figure 3.6: Convergence characteristic of PSO and hybrid PSO-BA for Alpine 1 function

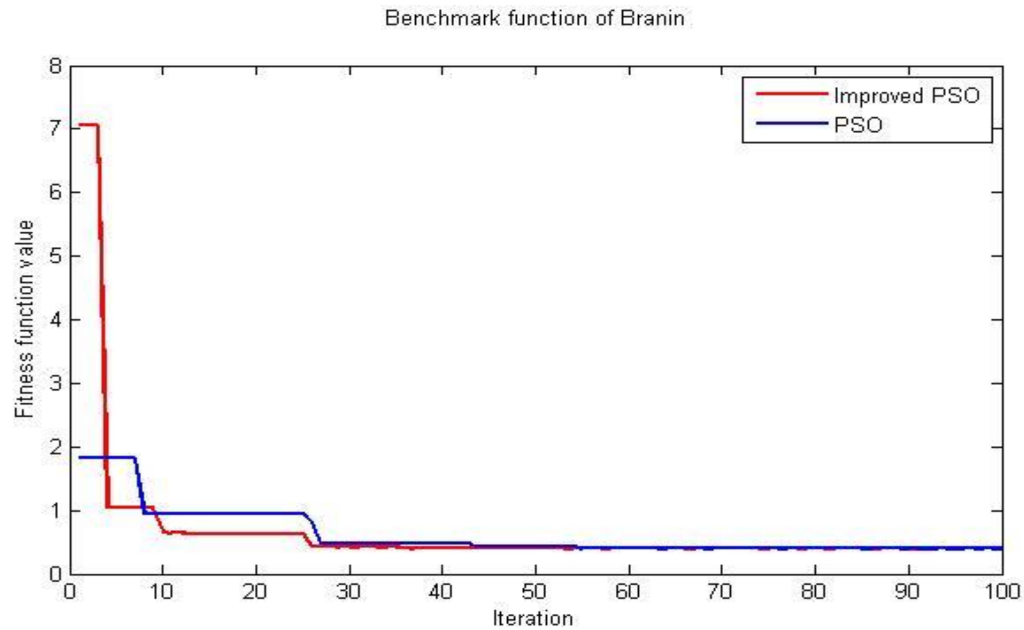

Figure 3.7: Convergence characteristic of PSO and hybrid PSO-BA for Branin function

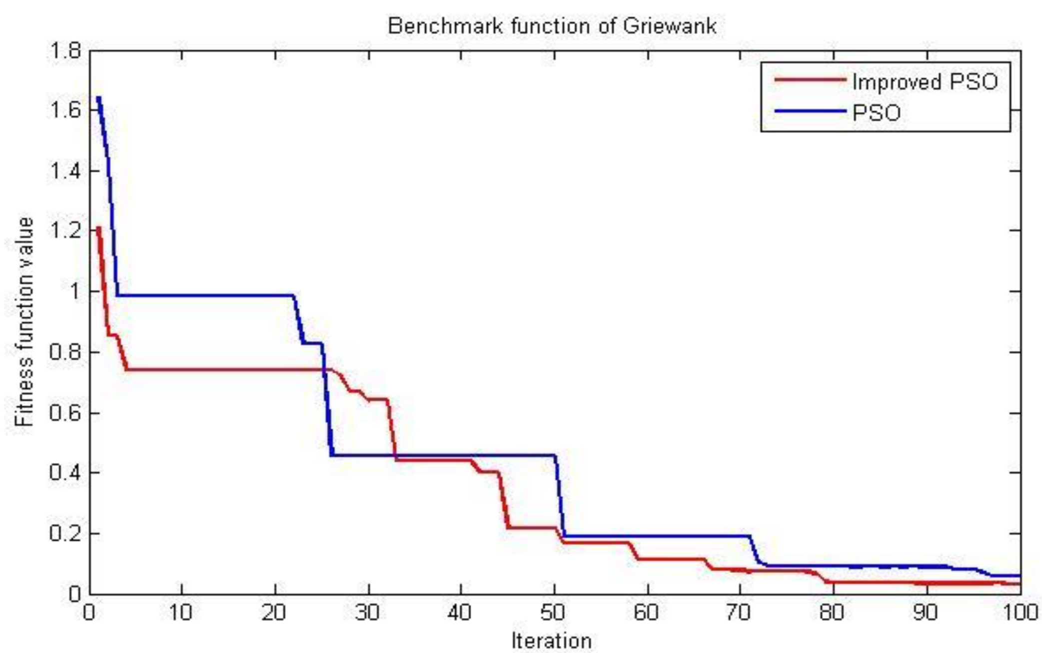

Figure 3.8: Convergence characteristic of PSO and hybrid PSO-BA for Griewank function

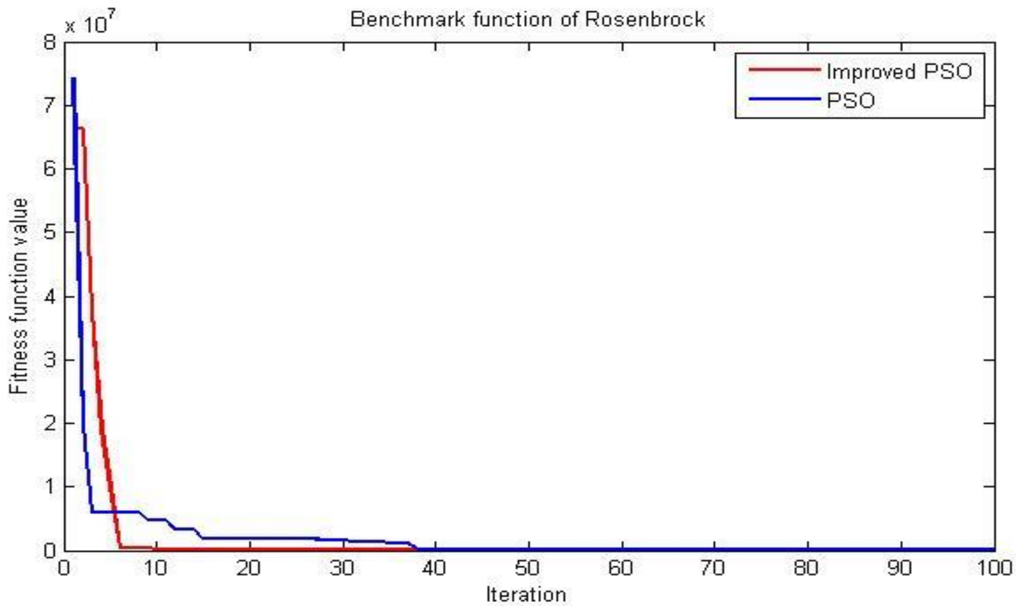

Figure 3.9: Convergence characteristic of PSO and hybrid PSO-BA for Rosenbrock function

Table 3.1 gives the results for the PSO, BA and improved PSO for 9 benchmark functions using the same criteria: best, mean, median, worst and standard deviation. The best results highlighted show that the improved PSO which is a hybrid PSO-BA performs better than the PSO and the bat algorithm together for the first 7 functions. And continue giving better value than PSO for the two last functions. For the function like Sphere, and rotated hyper ellipsoid where the optimal value is zero, the hybrid PSO-BA give zero for all the five criteria.

Table 3.1: Comparative results of PSO, BA and Hybrid PSO-BA in $/hr

function	**criteria**	**PSO**	**BA**	**Improved PSO**
f_1(x)	Best	0.0004	$\rightarrow 0$	$\rightarrow$ **0**
	Mean	0.0011	0.0016	$\rightarrow$ **0**
Sphere	Median	0.0006	0.0000	$\rightarrow$ **0**
	Worst	0.0043	0.1560	$\rightarrow$ **0**
	S.D	0.0012	0.0119	$\rightarrow$ **0**

function	criteria	PSO	BA	Improved PSO
f_2(x)	Best	1.8966	1.9899	**0.0621**
	Mean	8.3713	2.1550	**1.7259**
	Median	8.5126	1.9899	**1.0489**
Rastrigin	Worst	15.6164	17.2017	**4.0959**
	S.D	4.0935	**1.2409**	1.3429
f_3(x)	Best	0.5493	1.6462	**0.0002**
	Mean	2.2109	1.6587	**0.5756**
Ackley 1	Median	1.5076	1.6462	**0.0141**
	Worst	9.1946	2.6828	**2.4731**
	S.D	2.5810	**0.0893**	0.9417
f_4(x)	Best	0.0999	0.1999	**0.0999**
	Mean	0.1318	0.1998	**0.0998**
Salomon	Median	0.1048	0.1998	**0.0998**
	Worst	0.2014	0.1998	**0.0998**
	S.D	0.0473	$\to 0$	$\to$ **0**
f_5(x)	Best	0.0120	$\to 0$	$\to$ **0**
Rotated	Mean	0.1830	0.0052	$\to$ **0**
hyper-	Median	0.1583	0.0000	$\to$ **0**
ellipsoid	Worst	0.5044	0.4444	$\to$ **0**
	S.D	0.1644	0.0327	$\to$ **0**
f_6(x)	Best	0.0174	0.0000	$\to$ **0**
	Mean	2.2442	**0.0006**	0.0010
Alpine 1	Median	1.8033	0.0004	**0.0002**
	Worst	5.8968	0.0913	**0.0046**
	S.D	2.1076	0.0060	**0.0016**
f_7(x)	Best	0.3979	0.3979	**0.3979**
	Mean	0.6992	0.3982	**0.3978**
	Median	0.3979	0.3978	**0.3978**
Branin	Worst	1.4021	0.5138	**0.3978**

	S.D	0.4850	0.0051	$\rightarrow$ **0**
function	**criteria**	**PSO**	**BA**	**Improved PSO**
f_8(x) Griewank	Best	0.0513	**0.0000**	**0.0099**
	Mean	0.1555	**0.0015**	**0.0741**
	Median	0.1269	**0.0000**	**0.0655**
	Worst	0.3405	**0.2128**	**0.1998**
	S.D	0.0955	**0.0111**	**0.0561**
f_9(x) Rosenbrock	Best	5.3205	**0.0076**	**0.4656**
	Mean	200111.3	**1.7765**	**22.9587**
	Median	95.0779	**0.0660**	**3.5682**
	Worst	1000002	**61.3425**	**87.6316**
	S.D	521579.6	**5.4108**	**35.3942**

3.4.2. Hybrid PSO-BA application in Economic dispatch

To solve the ED problem and compare the performance of the improved PSO with its recently modified version, 5 different cases are proposed. 25 independent runs with 150 000 iterations are made. All the cases include the valve point effect of the generators. Appendix A depicts Matlab code for the hybrid PSO-BA for 13 generating units.

Table 4.2 shows that the improved PSO performs better than its initial algorithm for all the five cases. From the same table, the difference between values obtained helps to see the improvement of the results that is the reduction of cost is proportional to the number of generating units. Thus few generating units (6, 13, and 15) show little reduction in the cost while high number of generating units (40 and 140) shows higher impact on the reduction of the cost therefore higher improvement indices of the hybrid PSO-BA.

Table 3.2: Result of the 5 cases of generating units for the ED in $/hr

Generating units	criteria	PSO	Improved PSO
6 units	Best	15490.1	**15464.9**
	Mean	15587.3	**15527.0**
	Median	15580.9	**15522.8**
	Worst	15716.2	**15623.5**
	S.D	62.1	**39.0**
13 units	Best	18771.3	**18533.4**
	Mean	19292.3	**19189.1**
	Median	19272.6	**19219.2**
	Worst	19705.5	**19656.8**
	S.D	**225.6**	267.8
15 units	Best	38684.4	**38636.9**
	Mean	38904.1	**38845.0**
	Median	38848.1	**38805.1**
	Worst	**39177.9**	39313.3
	S.D	138.6	**136.0**
40 units	Best	131441.2	**130426.8**
	Mean	144429.8	**141211.0**
	Median	144537.8	**139795.4**
	Worst	**157086.3**	158797.7
	S.D	8115.5	**6522.5**

Generating units	**criteria**	**PSO**	**Improved PSO**
140 units	Best	**5728727.5**	5769300.5
	Mean	7832220.6	**7626462.1**
	Median	9298355.6	**7643879.4**
	Worst	9390437.0	**9344361.2**
	S.D	1747032.8	**1428972.2**

3.5. Conclusion

This chapter has presented an improved particle swarm optimization to deal with the economic dispatch problem. The ameliorated PSO is based on an integration of bat algorithm frequency and loudness characteristics in order to boost the PSO performance and also helps to tackle premature convergence which is another drawback of PSO. The new algorithm is an assimilation of some characteristics of BA by PSO. The proposed algorithm was tested on nine benchmark functions and applied to five different cases of economic dispatch problems. All the results show that it performs better than the original PSO. And it also has smoother and faster convergence characteristics. Seven benchmarks functions show that the hybrid PSO-BA gives better solution than PSO and BA together.

More studies can be done by considering other recently developed optimization algorithms to associate with PSO for a better performance to solve economic dispatch problems. The next chapter continues to look for ideas to improve the PSO algorithm. It will investigate the effect of Cuckoo Search algorithm based on Particle swarm Optimization.

CHAPTER 4: IMPROVED PARTICLE SWARM OPTIMIZATION BASED ON CUCKOO SEARCH OPERATIONS

This chapter investigates the Cuckoo Search Algorithm (CSA) in order to enhance the particle swarm optimization algorithm. It studies the different operations that constitute the CSA then analyses and experiments how to apply them to ameliorate the performance of the PSO algorithm. The next section reviews the literature around CSA. Then section two analyses the effect of levy flight and random search operation in Cuckoo and their application in PSO. The experimental results of the simulation are depicted in section three and section four compare the two ideas developed. Lastly the conclusion summarizes the whole chapter.

4.1. The Cuckoo search algorithm

Cuckoo search algorithm was developed by Xin-She Yang and Suash Deb in 2009. It is inspired by a lifestyle of a bird family called Cuckoo. The algorithm is based on eggs laying and breeding of cuckoos. Mature female cuckoo lays eggs in some other birds' nest; try to mimic them like other eggs in the nest. Then take one of the eggs of that nest away. If these eggs are not recognized and not killed by host birds, they grow and become mature cuckoo [36]. Cuckoos look for the most suitable nest to lay eggs in order to optimize eggs survival rate. The algorithm basically consists of 3 rules [42]:

- Each cuckoo lays one egg at a time and put them into host nest
- Best nests with high class of eggs (solutions) will carry over to the nest generations
- The number of available host nest is fixed, and a host can discover an egg laid by cuckoo with probability Pa. Pa $\in [0, 1]$.

The powerful performance of CSA is mostly due to the combination of two operational techniques. These include direct search through levy flight and random search based on the probability for a host bird to discover an alien egg in its nest [58]. In fact there is a probability Pa for the cuckoo eggs to be discovered by the host bird. This can results with either the destruction of the eggs or the abandon of the nest by the host bird. Pa is usually considered 0.25.

The Levy flight process is a random walk that forms a series of instantaneous jumps chosen from a heavy-tailed probability density function [55, 56]. A new solution $x_i(t+1)$ for cuckoo i is generated using a Lévy flight according to the following equation:

$$x_i^{(t+1)}=x_i^{(t)}+ \alpha \oplus Levy\ (\lambda) \tag{5.1}$$

where α (α>0) represents a step scaling size and ^ represents entry-wise multiplication. Those parameters must be related to the scales of problem the algorithm is trying to solve. In most cases, α can be set to the value of 1 or some other constants. The cuckoo search algorithm uses levy flight in the search process to improve the performance of the cuckoo. The algorithm is as follow [56]:

1) Generate initial population of n nests.
2) Move a cuckoo i randomly by flights.
3) Evaluate the fitnessF_i.
4) Randomly choose a nest j among n available nest.
5) If F_i>F_j then replace j by the new solution.
6) Abandon a fraction Pa of the worse nests and create the same fraction of new nests at new locations via levy flights.
7) Keep the best solutions (nests with quality solutions).
8) Sort the solution and the best current solutions.
9) If stopping criterion is not satisfied, increase generation number and go to step 2.
10) Post process results and find the best solution among all.

Cuckoo search algorithm is a strong tool to solve optimization problem. It has also been applied to solve the economic dispatch problem.

4.2. Effect of levy flight and random search operation in CSA to improved PSO performance

Levy flight and random search are the two main operations that cuckoo use to optimize function value. The improved algorithm uses firstly the levy flight update formula and the PSO update formula. The idea behind is to get from the two formula the best position of the particle. There is a comparative analysis to find the best update equation from the two formulas. That position is keep as the real update position of the algorithm for the iteration.

Secondly, considering that the cuckoo lying of eggs might be done with some velocities, the eggs discovering probability is applied to the velocity update equation in order to generate more solutions. The new algorithm obtained is as follow:

1) For each particle, initialize randomly the position and velocity vectors.
2) Measure the fitness of each particle (pbest) and store the particle with the best fitness (gbest) value.
3) PSO velocity and position update equation considering eggs discovering probability.

$$v_i^{k+1} = wv_i^k + c_1r_1\left(p\_b_i^k - x_i^k\right) + c_2r_2\left(g\_b^k - x_i^k\right) \quad (5.2)$$

$$x_i^{k+1} = x_i^k + rand \times v_i^{k+1} \qquad \text{For rand > pa} \quad (5.3)$$

4) Update through levy flight

σ = (γ (1+ β)× sin (π × β /2)/ (γ ((1+ β)/2)× β ×2^ ((β -1)/2))) ^ (1/ β);

Step=rand/abs (rand) ^ (1/ β);

$$Dy_i^{k+1} = step \times \sigma \times (p\_b_i^k - g\_b^k) \quad (5.4)$$

$$y_i^{k+1} = x_i^k + rand \times Dy_i^{k+1} \quad (5.5)$$

5) Compare the value obtained in step (3) and step (4) and keep the best update position.
6) Evaluate fitness, update pbest.
7) Update Gbest of population.

8) Repeat steps 2-7 until a termination criterion is satisfied.

4.3. Experimental results

The improved PSO using Cuckoo assets and the original PSO program have been tested first on ten benchmark functions and then on the five different cases of economic dispatch problem which comprise 6, 13, 15, 40 and 140 generating units. Matlab software version 7.10.0.499 (R2010a), installed on a personal computer with a 2.6GHz processor and 8GB RAM, running on windows 7, is used to program and test the algorithm

4.3.1. Simulations and analysis

Best, mean, median, worst, and standard deviation remain the five criteria considered to evaluate and compare the three algorithms. The simulation is made of 10 independent runs of 100 iterations each. Figures 5.1-10 depict the convergence characteristic of the PSO and the improved one for ten test functions. Then Table 5.1 presents the results of simulation for the PSO and the improved PSO for the ten benchmark functions.

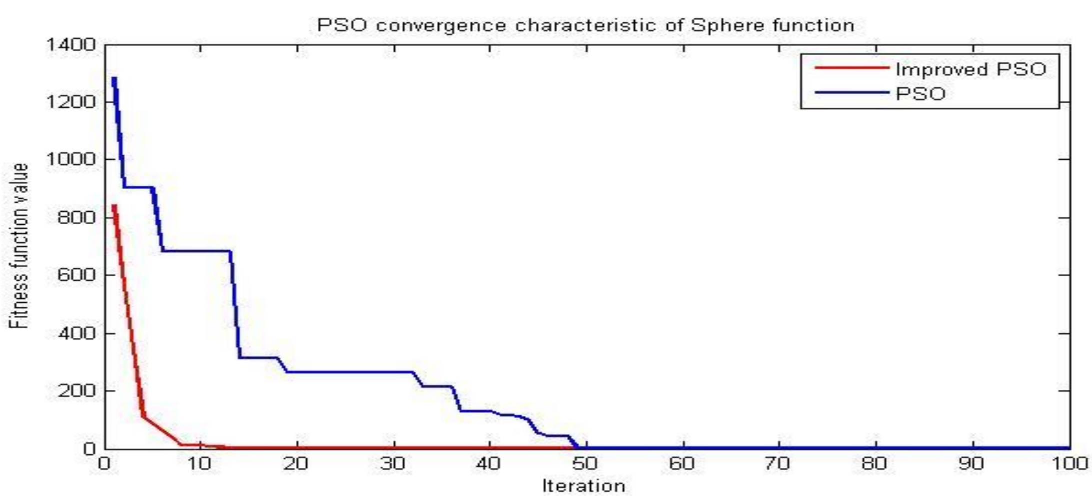

Figure 4.1: Convergence characteristic of PSO and the improved PSO for Sphere function

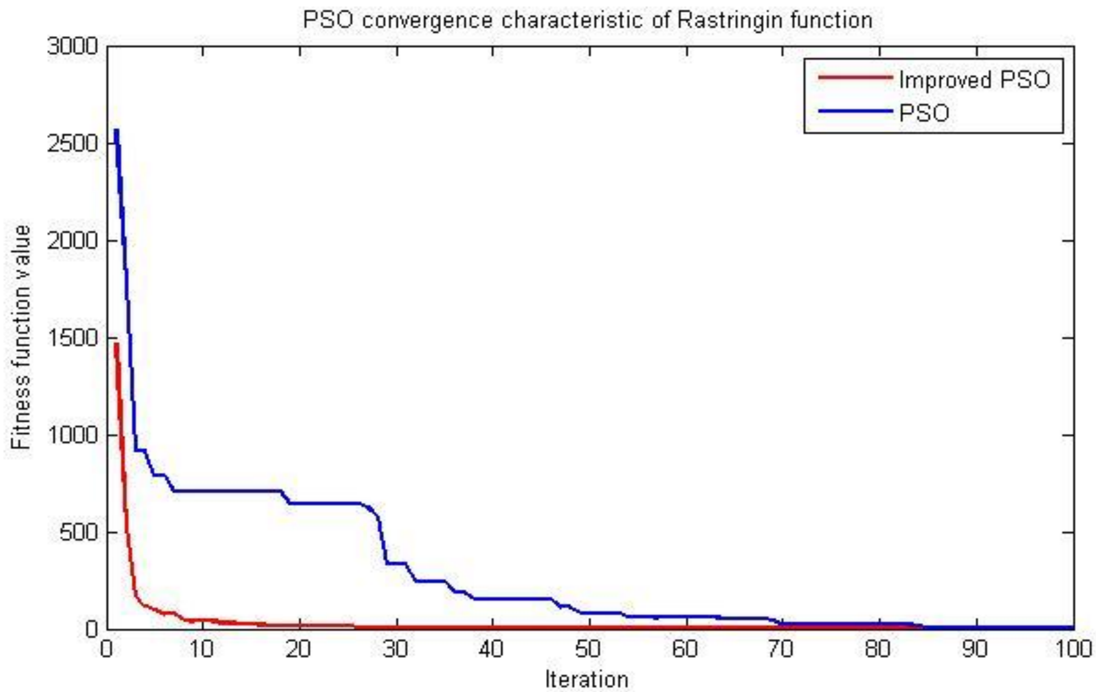

Figure 4.2: Convergence characteristic of PSO and improved PSO for Rastringin function

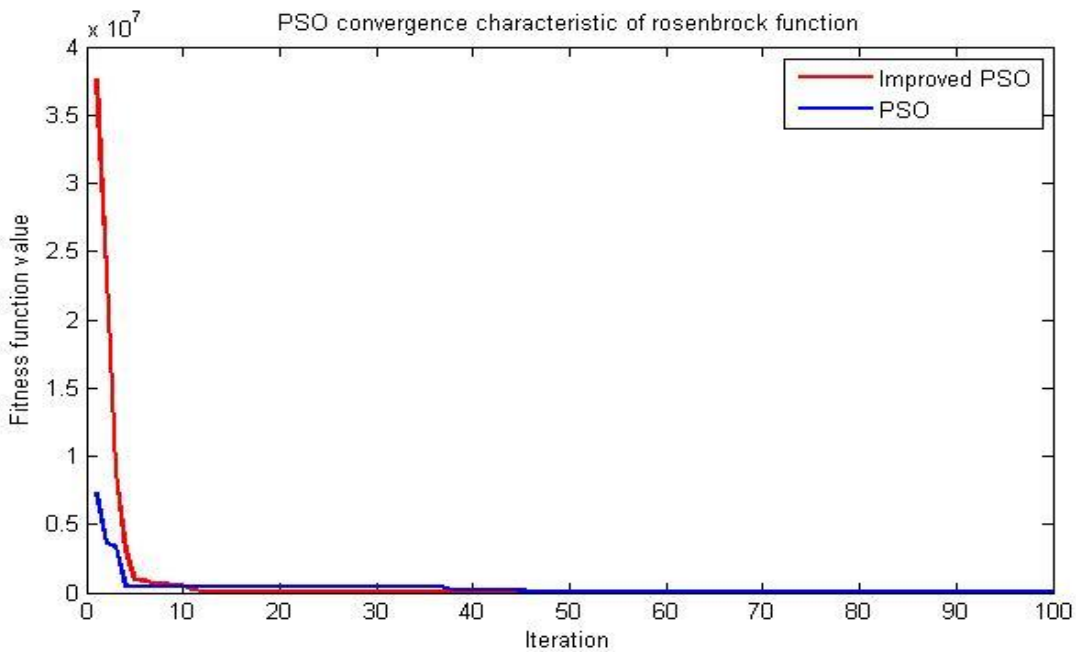

Figure 4.3: Convergence characteristic of PSO and improved PSO for Rosenbrock

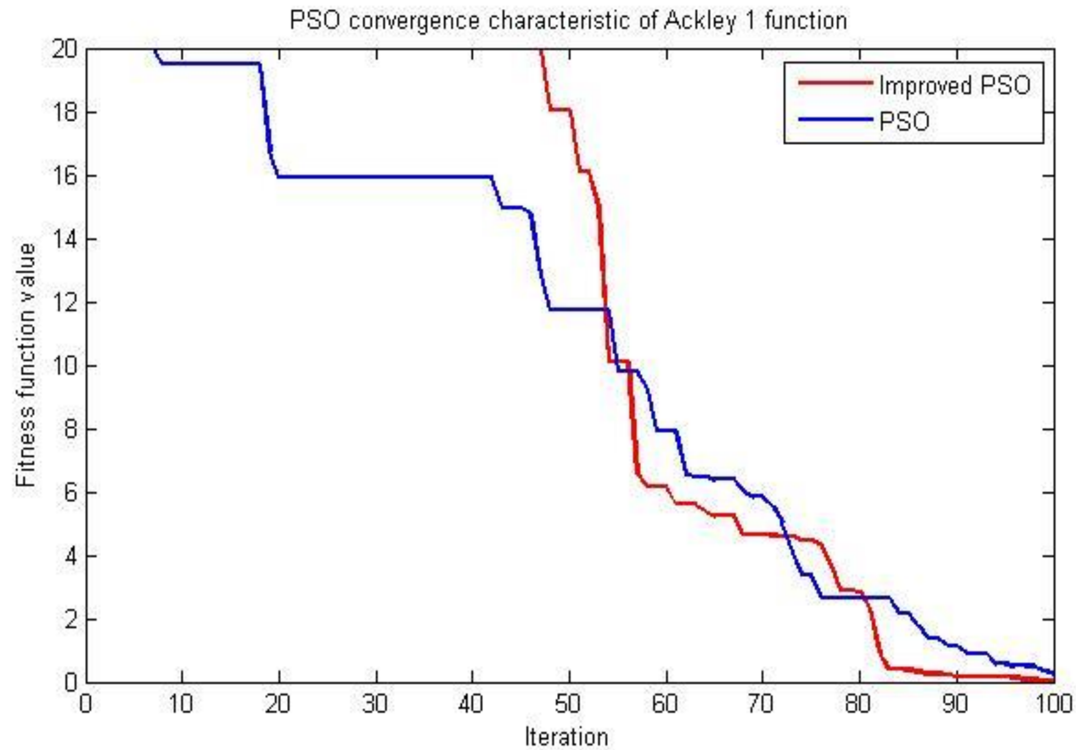

Figure 4.4: Convergence characteristic of PSO and improved PSO for Ackley 1 function

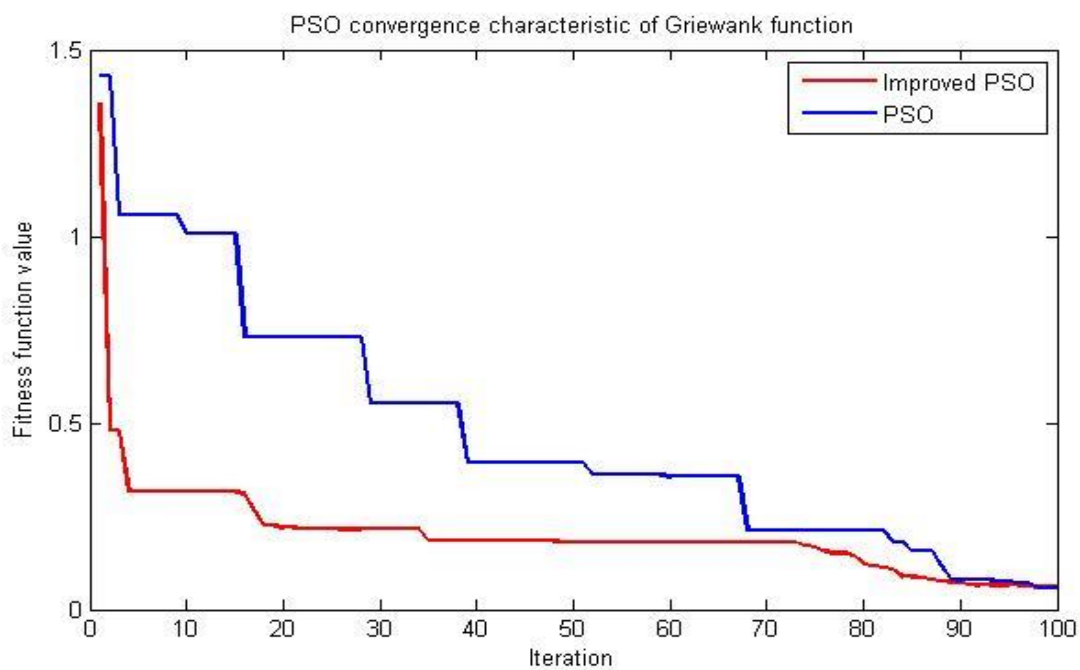

Figure 4.5: Convergence characteristic of PSO and improved PSO for Griewank function

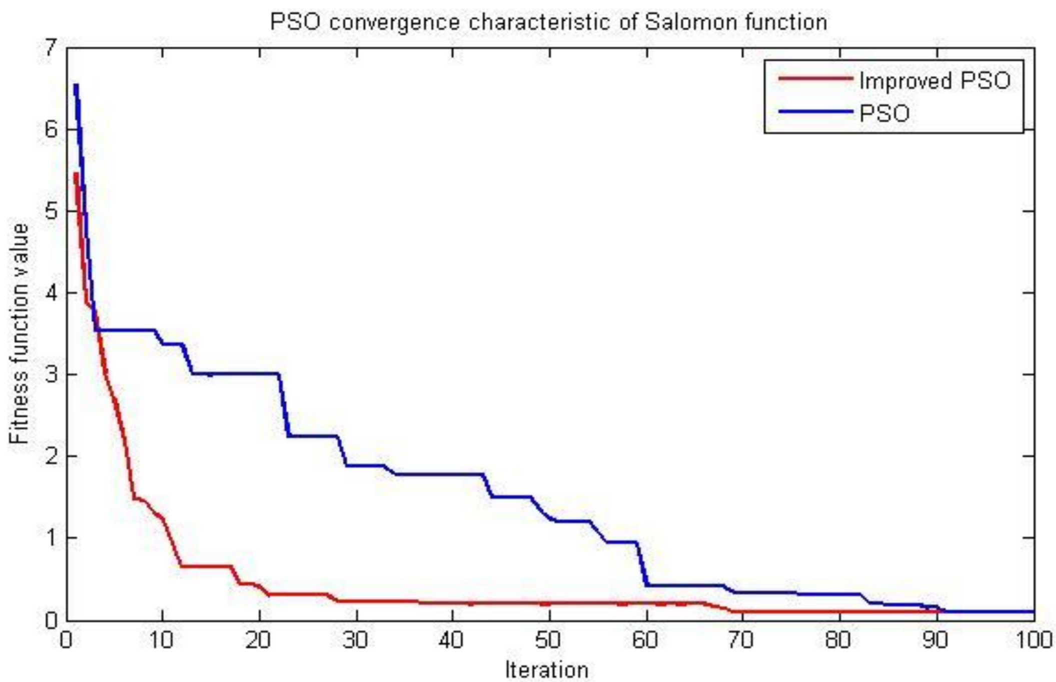

Figure 4.6: Convergence characteristic of PSO and improved PSO for Salomon function

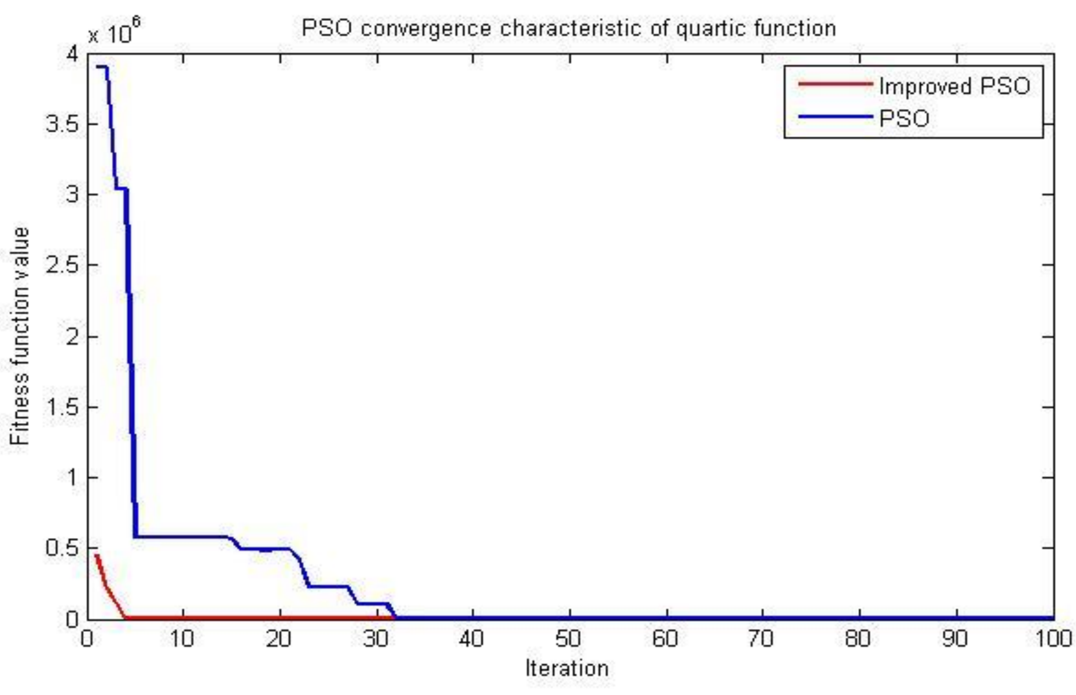

Figure 4.7: Convergence characteristic of PSO and improved PSO for quartic function

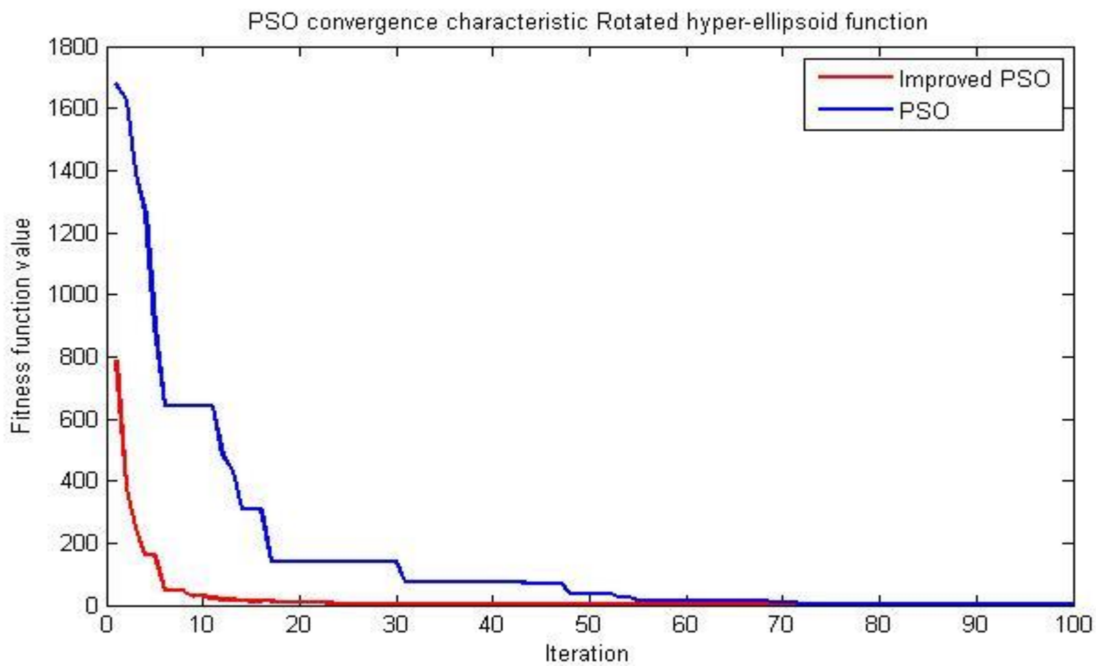

Figure 4.8: Convergence characteristic of PSO and improved PSO for Rotated hyper-ellipsoid function

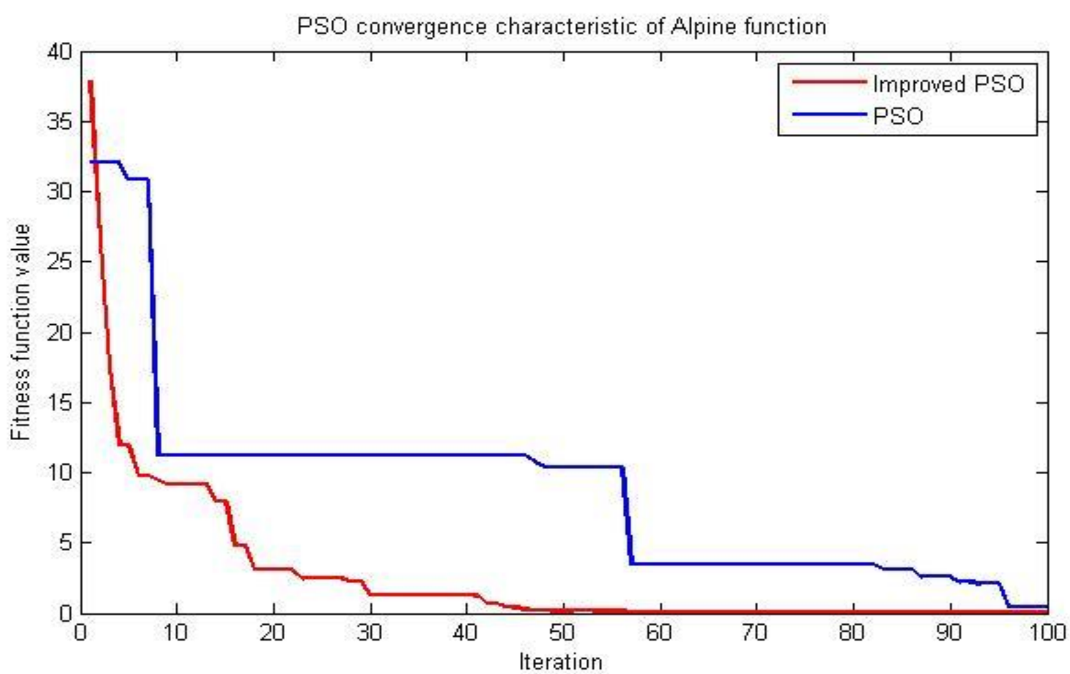

Figure 4.9: Convergence characteristic of PSO and improved PSO for Alpine function

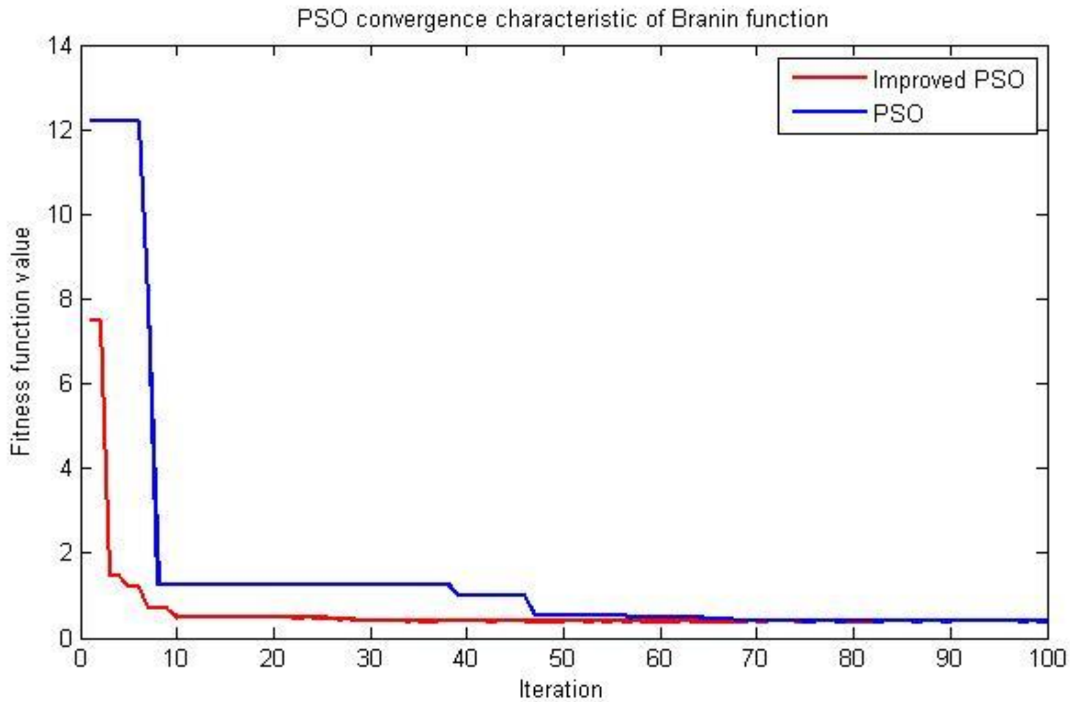

Figure 4.10: Convergence characteristic of PSO and improved PSO for Branin function.

In every plot depicted from figures 5.1 to 5.10, the red colour represents the improved PSO convergence characteristic and the blue one represents the original PSO before the improvement through Cuckoo search operation. The overall graphs show that the ameliorated algorithm has a faster convergence characteristic than the normal PSO. A deep observation helps to notice that for a smaller number of iterations, the gap between the two algorithms could have been bigger. This because five figures out of the 10 reaches their optimal value about 50% of the total iteration count. Furthermore, it is good to observe that the curves of the improved PSO are smoother because the curves have fewer changes in direction during their convergence. Thus, the improved PSO has a smoother convergence to the optimal solution for all the ten benchmark functions.

Table 5.1 demonstrates that the improved PSO using Cuckoo's operational technique that is levy flights and random search; performs better than PSO for all the ten benchmark functions. That is it gives lowest value for all the 10 benchmarks functions. Moreover, it gives the optimal value for all the five criteria for the function sphere and Rotated hyper-ellipsoid.

Table 4.1: Improved PSO algorithm compare with PSO in $/hr.

function	criteria	PSO	Improved PSO
f_1(x) Sphere	Best	0.0001	→ **0**
	Mean	0.0006	→ **0**
	Median	0.0006	→ **0**
	Worst	0.0016	→ **0**
	S.D	0.0005	→ **0**
f_2(x) Rastrigin	Best	5.5506	**1.2597**
	Mean	9.1947	**4.0126**
	Median	8.0566	**3.3870**
	Worst	13.2185	**6.6601**
	S.D	2.8995	**1.8036**
f_3(x) Rosenbrock	Best	2.5302	**0.0096**
	Mean	248.8170	**159.4381**
	Median	52.8431	**16.0943**
	Worst	1085.7700	**567.4930**
	S.D	441.0580	**238.9095**
f_4(x) Ackley 1	Best	0.5851	**0.0000**
	Mean	3.0096	**0.0406**
	Median	2.0174	**0.0014**
	Worst	7.6726	**0.0150**

	S.D	2.3069	**0.0585**
Function	**criteria**	**PSO**	**Improved PSO**
f_5(x) Griewank	Best	0.0256	**0.0156**
	Mean	0.1834	**0.0728**
	Median	0.1603	**0.0753**
	Worst	0.4706	**0.1624**
	S.D	0.1253	**0.04165**
f_6(x) Salomon	Best	0.0999	**0.0999**
	Mean	0.1220	**0.0998**
	Median	0.1007	**0.0998**
	Worst	0.2009	**0.0998**
	S.D	0.0414	$\rightarrow$ **0**
f_7(x) Quartic function	Best	0.0527	**0.0531**
	Mean	0.0509	**0.3700**
	Median	0.4647	**0.3426**
	Worst	0.9436	**0.8261**
	S.D	0.3351	**0.2656**
f_8(x) Rotated hyper-ellipsoid	Best	0.0117	$\rightarrow$ **0**
	Mean	0.1267	$\rightarrow$ **0**
	Median	0.0848	$\rightarrow$ **0**

	Worst	0.6276	**0.0001**
	S.D	0.1825	**$\rightarrow 0$**
Function	**criteria**	**PSO**	**Improved PSO**
f_9(x)	Best	0.0069	**0.0002**
Alpine	Mean	0.9801	**0.0220**
	Median	0.6489	**0.0021**
	Worst	2.4238	**0.1949**
	S.D	0.9479	**0.0608**
f_{10}(x)	Best	0.3979	**0.3979**
Branin	Mean	0.4988	**0.4696**
	Median	0.3979	**0.3979**
	Worst	1.4021	**1.0024**
	S.D	0.3172	**0.1905**

4.3.2. The improved PSO application in Economic dispatch

The improved PSO which is an association with techniques from cuckoo search algorithm is applied to solve the ED problem and compare the performance with its recent version. 5 different cases are proposed. 25 independent runs with 150 000 iterations are made. The results of the best, mean, median, worst fuel cost and the standard deviation are used to evaluate the algorithms. Appendix B shows Matlab code to solve the ED with 40 generating units with improved PSO. Table 5.2 shows that the improved PSO gives better results when applied to the economic dispatch problem for all the five different cases. It also shows that the improvement is proportional to the number of generating units. That is units 6, 13 and 15 show little improvement compared to 40 and 140 units.

Table 4.2: Result of the 5 cases of generating units for the ED in $/hr

Generating units	**criteria**	**PSO**	**Improved PSO**
6 units	Best	15490.1	**15458.5**
	Mean	15587.3	**15564.9**
	Median	15580.9	**15549.3**
	Worst	15716.2	**15645.6**
	S.D	62.1	**37.3**
13 units	Best	**18771.3**	18804.7
	Mean	19292.3	**19205.7**
	Median	19272.6	**19231.1**
	Worst	19705.5	**19442.7**
	S.D	225.6	**165.1**
15 units	Best	38684.4	**38658.5**
	Mean	38904.1	**38893.9**
	Median	**38848.1**	38890.4
	Worst	39177.9	**39172.1**
	S.D	138.6	**138.2**
40 units	Best	131441.2	**130567.9**
	Mean	144429.8	**141469.4**
	Median	144537.8	**141425.9**
	Worst	**157086.3**	161742.0
	S.D	8115.5	**8016.7**
140 units	Best	**5728727.5**	5740224.3
	Mean	7832220.6	**7799231.9**
	Median	9298355.6	**8990707.2**
	Worst	9390437.0	**9382885.0**
	S.D	1747032.8	**1699574.5**

4.4. Comparison of the two ideas developed to improve the PSO performance when applied to the ELD

Although PSO using CSA has been successful when tested with 10 benchmark functions while hybrid PSO-BA was only with 9, table 5.3 shows that the latter has better improvement when applied to solve economy load dispatch problems. According to the same table, hybrid PSO-BA gives the lowest cost for at least four criteria out of the five considered for the analysis. And it is good to note that the improvement is observed when at least three criteria are satisfied. Table 5.3 shows the results of the two ideas employed to ameliorate the PSO algorithm plus the initial PSO algorithm

Moreover, figures 5.11-15 depict the convergence characteristics of the improved PSO-BA and PSO-CSA for the five different cases. A deep observation of the five figures helps to analyse the difference between the two algorithms. In fact, the first four pictures show a slight variation of the value during the convergence and seem to reach the same final value. While the fifth picture that is 140 generating units depicts a clear difference between the Improved PSO-BA and PSO-CSA. The overall graphs show that each algorithm could have reached the same final solutions with less iteration count. Since on every plot, the final value is reached about 50% of its final iteration count. All these show that hybrid PSO-BA has better, faster and smoother convergence characteristics than the improved PSO-CSA does.

Table 4.3: PSO with the other ideas developed for the ED in $/hr

Generating Units	**criteria**	**PSO**	**Hybrid PSO-BA**	**PSO using Cuckoo Operations**
6 units	Best	15490.1	**15464.9**	15458.5
	Mean	15587.3	**15527.0**	15564.9
	Median	15580.9	**15522.8**	15549.3
	Worst	15716.2	**15623.5**	15645.6
	S.D	62.1	39.0	**37.3**

Generating units	**criteria**	**PSO**	**Hybrid PSO-BA**	**PSO using Cuckoo Operations**
13 units	Best	18771.3	**18533.4**	18804.7
	Mean	19292.3	**19189.1**	19205.7
	Median	19272.6	**19219.2**	19231.1
	Worst	19705.5	**19656.8**	19442.7
	S.D	225.6	267.8	**165.1**
15 units	Best	38684.4	**38636.9**	38658.5
	Mean	38904.1	**38845.0**	38893.9
	Median	38848.1	**38805.1**	38890.4
	Worst	39177.9	39313.3	**39172.1**
	S.D	138.6	**136.0**	138.2
40 units	Best	131441.2	**130426.8**	130567.9
	Mean	144429.8	**141211.0**	141469.4
	Median	144537.8	**139795.4**	141425.9
	Worst	**157086.3**	158797.7	161742.0
	S.D	8115.5	**6522.5**	8016.7
140 units	Best	**5728727.5**	5769300.5	5740224.3
	Mean	7832220.6	**7626462.1**	7799231.9
	Median	9298355.6	**7643879.4**	8990707.2
	Worst	9390437.0	**9344361.2**	9382885.0
	S.D	1747032.8	**1428972.2**	1699574.5

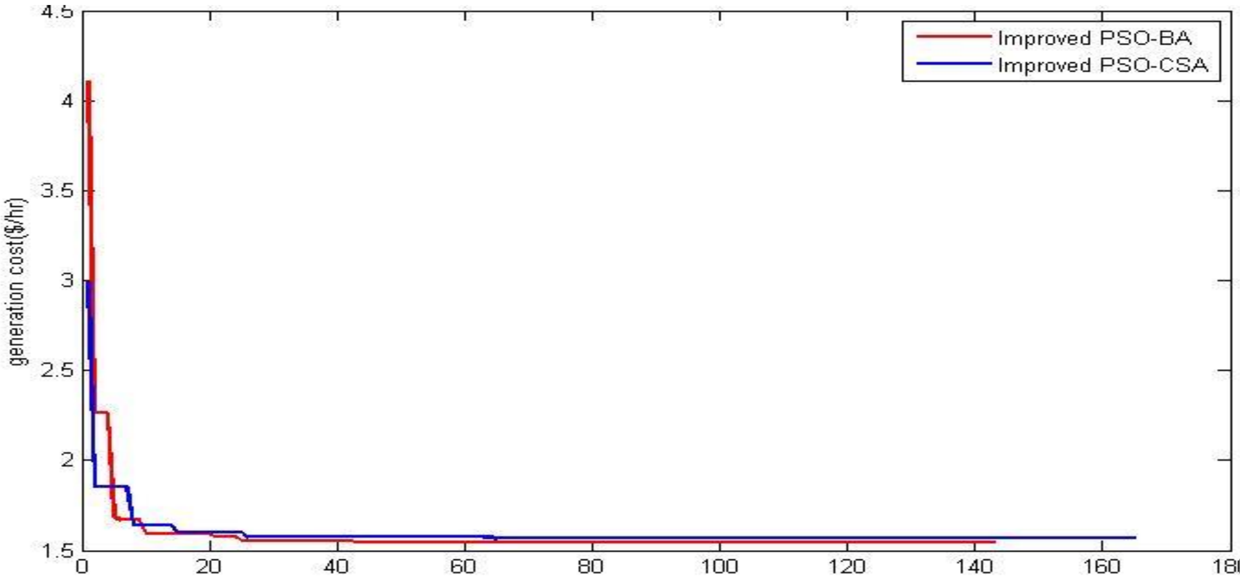

Figure 4.11: Convergence characteristic of improved PSO-BA and PSO-CSA for 6 generating units

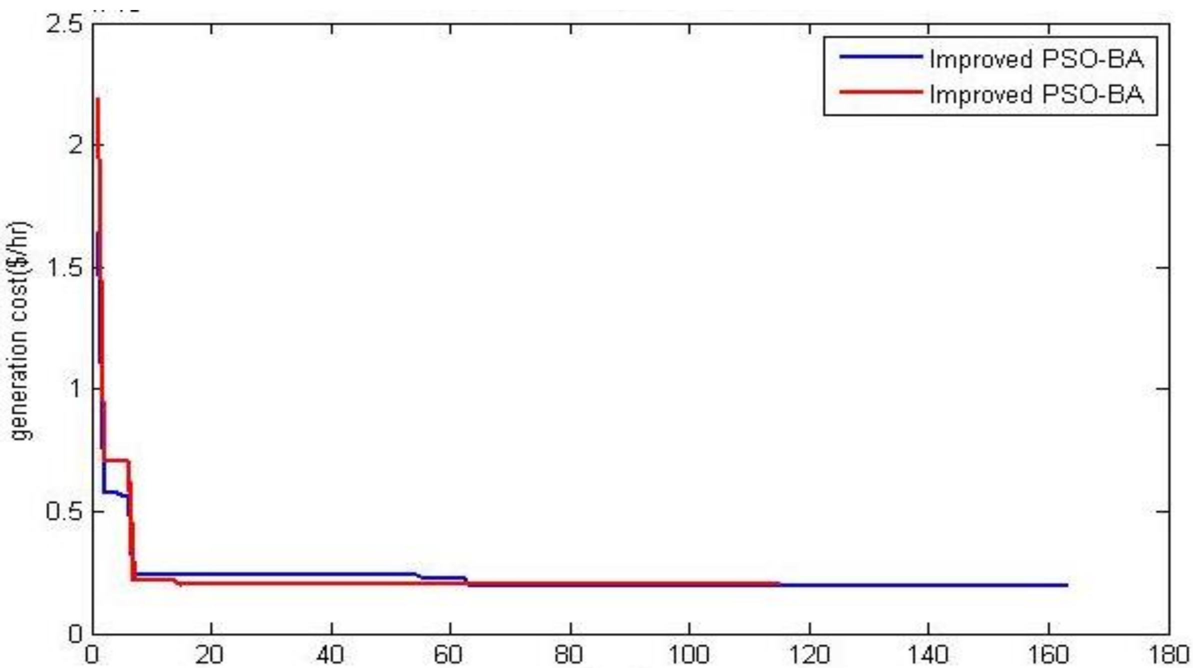

Figure 4.12: Convergence characteristic of improved PSO-BA and PSO-CSA for 13 generating units

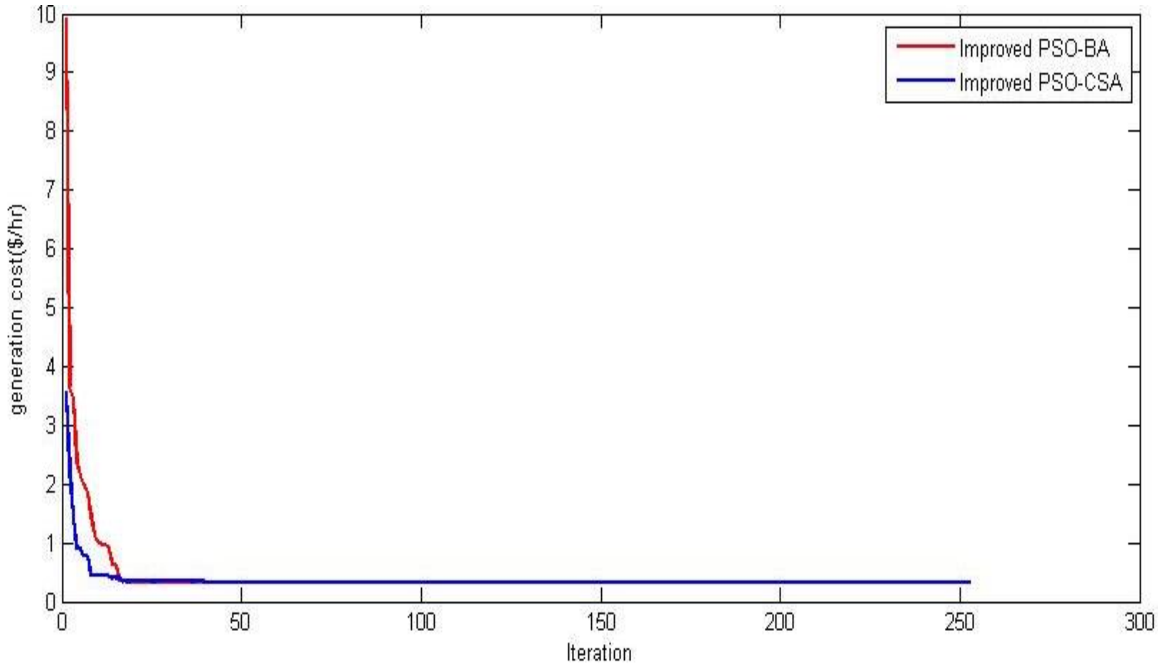

Figure 4.13: Convergence characteristic of improved PSO-BA and PSO-CSA for 15generating units

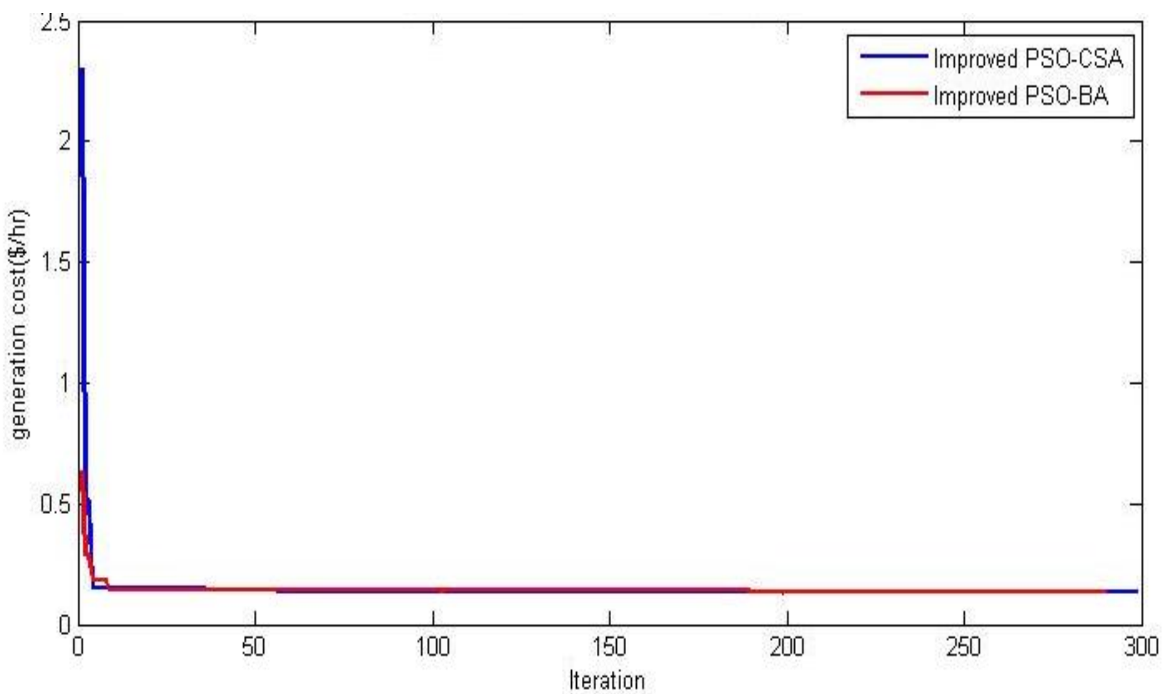

Figure 4.14: Convergence characteristic of improved PSO-BA and PSO-CSA for 40 generating units

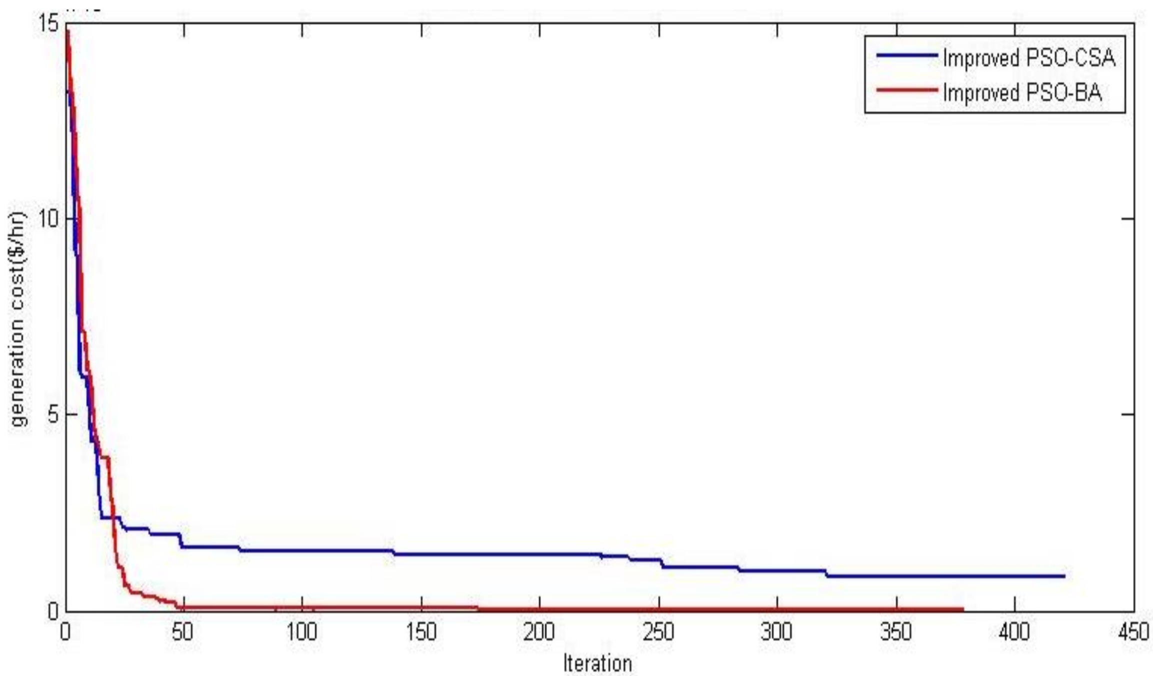

Figure 4.15: Convergence characteristic of improved PSO-BA and PSO-CSA for 140 generating units

4.5. Conclusion

This chapter has investigated the effect of levy flight and random search operation in Cuckoo search algorithm in order to ameliorate the performance of the particle swarm optimization algorithm. The levy flight is used as a comparative operation with the PSO update position. Random search in the other hand is applied through the probability for the cuckoo's egg to be discovered by the host bird and the fact that cuckoo uses some velocity to detect a nest, lays and mimic its egg so that it can look like those of the host and move to the other nest. The new algorithm obtained through experimentation of those two techniques has shown a big improvement when applying with 10 well known benchmark functions. Moreover, when applying it to solve the economic dispatch problem, it gives satisfactory improvement and better results.

CHAPTER 5: CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In conclusion, the dissertation has presented a practical economic load dispatch with all the details of all the constraints that are usually encountered. The research was focus on improving the particle swarm optimization algorithm in order to have better results of specific cases of economic dispatch problems. It has first applied the most recent version of the PSO algorithm which is also called its modified version. Then experimental studies through the simulation via Matlab software of different ideas on improving the algorithm were made. Many hypotheses have been considered in order to do so: improve PSO as a stand-alone algorithm; combine PSO with other algorithms that have been successful to solve the ED problem in order to make hybrid techniques using their topological structure.

The first successful idea is an improved PSO based on an integration of bat algorithm frequency and loudness characteristics in order to boost the PSO performance and form a hybrid PSO-BA. The new algorithm is an assimilation of some characteristics of BA by PSO. The proposed algorithm is tested on 9 benchmark functions and applied to five different cases of the economic dispatch problems. All the results show that it performs better than the original PSO. And it also has smoother and faster convergence characteristics. 7 benchmarks test show that the hybrid PSO-BA gives better solution than PSO and BA together.

Another experimental studies have also helped to found a new algorithm not hybrid to Cuckoo search algorithm but more inspired by the Cuckoo search operations (levy flight and random search operation). The new algorithm depicts a big improvement when applying with 10 well known benchmark functions and when applied to the economic dispatch problem it gives a more optimal solution that is a lower cost related to the generation and transport of power.

The research has helped to tackle some drawbacks of particle swarm optimization algorithm that are: premature convergence, fast falling into local optimal and the lack of competitiveness compared to other algorithm. Through the results from the simulation, the improved algorithms developed show considerable reduction in the cost of production and transport of electricity, which is the core purpose of economic dispatch of power systems, compare to the latest PSO algorithm. These can have a significant impact on people daily live with the high cost of electricity in the world in general and in South Africa in particular. However, in the family of optimization algorithms, PSO still need more improvement to become the best evolutionary algorithm

5.2 Future works

The development of more powerful computers and the gathering of data that are becoming easier, announce future good days in the field of artificial intelligence in their research for answers to today problems. Evolutionary algorithms and specifically particle swarm optimization which has received lots of attentions these recent years have many ways of improving and a broad area to be applied. Further studies can be done in improving the PSO algorithm through association with other nature inspired algorithms or as a standalone via multi swarm optimization. Moreover, Research can also be done in order to apply the two new ideas found in other area of science or technologies to optimize the performance, minimize the cost or reduce the time related to the accomplishment of something. This includes also the possibilities to apply the two improved PSO to Dynamic Economic Dispatch (DED) problems.

REFERENCES

[1] M.R. AlRashidi, and M.E. El-Hawary, "Survey of Particle Swarm Optimization Applications in Electric Power Systems*," IEEE transaction on evolutionary computation*, vol. 13, no. 4, pp.913- 917, august 2009.

[2] H. Mostefa, "Dynamic economic dispatch by the use of global optimization methods," *MTech dissertation the university Mohamed Khider- Biskra, Algeria*, 14 January 2013.

[3] J. D. Glover, M. S. Sarma, "Power System analysis and design", 3rd edition Brooks/Cole ISBN: 0- 534- 95367- 0, 2001, pp.504- 547.

[4] A.J. Wood, and B.F. Wollenberg, "Power generation, operation and control", 3rd Ed. New York: John Wiley & Sons Press, 2013.

[5] J.C. Dodu, P. Martin, A. Merlin, and J. Pouget, "An optimal formulation and solution of short-range operating problems for a power system with flow constraints," *in Proceeding of the IEEE*, pp.54-63, 1972.

[6] T. Ding, Rui Bo, F. Li, and H. Sun, "Bi-level Branch and bound method for economic Dispatch with disjoint prohibited zones considering network losses*," IEEE Transaction on power systems*, vol. 30, No 6, pp. 2841-2855, November 2015.

[7] B.R. Adarsh, T. Raghunathan, T. Jayabarathi, X.S. Yang, "Economic dispatch using chaotic bat algorithm" Energy, pp. 666- 675, 2016.

[8] G. Kumar, R. Singh, "Economic dispatch of power system optimization with power generation schedule using evolutionary technique," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, Issue 7, pp.10715-10722, July 2014.

[9] D.C. Walters, and G.B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading*," IEEE Transaction on power systems*, pp.1325-1332, 1993

[10] A. I. El-Gallad, M. El-Hawary, A. A. Sallam, and A. Kalas, “Swarm intelligence for hybrid cost dispatch problem,” *in Proceeding of the Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 753–757, 2001.

[11] X.S. Yang, S.S.S. Hosseini, and A.H. Gandomi, “Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect,” *Applied Soft Computing 2012*, pp.1180- 1186.

[12] M.H. Sulaiman, Z.N. Zakaria, M.I. Mohd-Rashid, and S.R.A. Rahim, “A new swarm intelligence technique for solving economic dispatch problem,” *in Power Engineering and Optimization IEEE Conference*, pp. 199-202, 2013.

[13] P. Lu, J. Zhou, H. Zhang, R. Zhang, and C. Wang, “Chaotic differential bee colony Optimization algorithm for dynamic economic dispatch problem with valve point Effects,” *International Journal in Electrical Power & Energy Systems*, pp.130-143, 2014.

[14] J. Kennedy and R. Eberhart, “Particle swarm optimization*,” in Proceeding of the IEEE International Conference on Neural Network*, vol. 4, pp. 1942–1948, 1995.

[15] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *in Proceeding of the 6th International Symposium on Micro Machine Human Science*, pp. 39–43, 1995.

[16] H. Xiaohui, S. Yuhui, and R. Eberhart, “Recent advances in particle swarm,” *in Proceeding of the Congress on Evolutionary Computation*, vol. 1, pp. 90–97, 2004.

[17] Z. L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints*,” IEEE Transaction on power systems*, vol. 18, no. 3, pp. 1187–1195, Nov. 2003.

[18] J. Sun, V. Palade, X.J. Wu, W. Fang, and Z. Wang, “Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization,” *IEEE Transaction on industrial Informatics*, pp.222-232, 2014.

[19] T.A.A. Victoire, and A.E. Jeyakumar, “Hybrid PSO-SQP for economic dispatch with Valve- point effect,” *Electric Power Systems Research*, pp.51-59, 2004.

[20] D. Swagatam, and P.N. Suganthan, “Problem definition and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems,” *Technical report*, Dec 2010.

[21] JC Welman and Sj Kruger, “Research Methodology”, 2nd Edition, Oxford university press, Cape Town, South Africa.

[22] V. Miranda and N. Fonseca, “EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems*,” in Proceeding of the IEEE/PES Transmission Distribution Conference Exhibition, Asia-Pacific*, vol.2, pp.745–750, 2002.

[23] T. Wael, Elsayed, F. Ehab, and El- Saadany, “A fully decentralized approach for solving the Economic Dispatch problem*,” IEEE Transactions on power systems,* Vol. 30, No. 4, pp. 2179- 2189, July 2015.

[24] A. I. S. Kumar, K. Dhanushkodi, J. J. Kumar, and C. K. C. Paul, “Particle swarm optimization solution to emission and economic dispatch problem,” *in Proceeding of the Conference on Convergent Technology, Asia-Pacific Region,* vol. 1, pp. 435–439, 2003.

[25] Hardiansyah, “a modified particle Swarm Optimization Technique for economic load dispatch with valve-point effect”, *IJ. Intelligent Systems and Applications,* pp.32- 41, July 2013.

[26] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee, “A particle swarm optimization for economic dispatch with non-smooth cost functions*,” IEEE Transaction on Power Systems,* vol. 20, no. 1, pp. 34–42, February. 2005.

[27] M. N. Alama, “Particle Swarm Optimization: Algorithm and its Codes in MATLAB” *Research Gate Indian Institute of Technology, Roorkee-247667*, India March 2016.

[28] J. Jain and R. Singh, “Biogeography- based optimization algorithm for load dispatch in power system*”, International journal of emerging technology and advanced engineering,* vol. 3, Issue 7, pp.549- 553, July 2013.

[29] L.D.S. Santos, Coelho and V.C. Mariani, “Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve- point effect*,”* *IEEE transactions on power systems*, VOL. 21, no.2, May 2006.

[30] A. El-Gallad, M. El-Hawary, A. Sallam, and A. Kalas, “Particle swarm optimizer for constrained economic dispatch with prohibited operating zones,” *in Proceeding of the Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 78–81, 2002.

[31] Z.L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” *IEEE Transaction on power systems*, pp. 1187- 1195, 2003.

[32] X.S. Han, H. B. Gooi, and D.S. Kirschen, “Dynamic Economic dispatch: Feasible and Optimal Solutions*,” IEEE transactions on power systems,* Vol. 16, NO. 1, February 2001.

[33] D. W. Ross and S. Kim, “Dynamic economic dispatch of generation,” *IEEE Transaction on power Apparatus and Systems,* vol.1, no. 6, pp.22- 28, 1980.

[34] K.T. Chaturvedi, M. Pandit, and L. Srivastava, “Self-organizing hierarchical particle Swarm optimization for non-convex economic dispatch,” *IEEE Transaction on power systems*, pp.1079- 1087, 2008.

[35] T. A. A. Victoire and A. E. Jeyakumar, “Reserve constrained dynamic dispatch of units with valve-point effects,” *IEEE Transaction on Power Systems,* vol. 20, no. 3, pp. 1273–1282, Aug. 2005.

[36] R. Rajabioum, “Cuckoo optimization algorithm, “*Applied Soft Computing*, pp.5508-5518, November 2011.

[37] K.T. Chaturvedi, M. Pandit, and L. Srivastava, “Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch”, *International Journal of Electrical and Power Energy System*. pp.249–257, June 2009.

[38] K. Meng, H.G. Wang, Z. Dong, and K.P. Wong, “Quantum-inspired particle swarm optimization for valve-point economic load dispatch”, *IEEE Transaction on Power Systems*. pp. 215–222, January 2010.

[39] A. Selvakumar, and K. Thanushkodi, "Anti-predatory particle swarm optimization: solution to non-convex economic dispatch problems", *Electrical Power Systems Research*, pp.2–10, January 2008.

[40] E. Ulker and S. Ulker, "Microstrip coupler design using bat algorithm", *International Journal of Artificial Intelligence and Application*, Vol 5, No1, pp.127- 133, January 2014.

[41] A. Ochoa, L. Margin, J. Arreola, A. De Luna, G. Garcia, E. Soto, S. Gonzalez, K. Haltaufoerhyde and V. Scarandangotti, "Improved solution based on bat algorithm to vehicle routine problem in a caravan range community", *2013 Thirteenth International Conference on Hybrid Intelligent Systems*, ,pp18- 22, December 2013.

[42] R. Kaur and G.K. Gill, "Solution to economic load dispatch problem using cuckoo search algorithm*", International Journal of Electrical and Electronics Research,* Vol 3, pp.362-369, 2015.

[43] A.B.S. Serapiao, "Cuckoo search for solving economic load dispatch problem", *Intelligent control and automation*, pp.385- 390, 2013.

[44] S. M. Elsayed, R. A. Sarker and D. L. Essam, "GA with a New Multi-Parent Crossover for solving IEEE-CEC 2011 Competition Problems", *conference paper*, pp.1034- 1040, June 2011.

[45] S. M. Elsayed, R. A. Sarker and D. L. Essam, "Differential Evolution with Multiple Strategies for Solving CEC 2011 Real-world Numerical Optimization Problems", *conference paper*, pp.1041-1048, , June 2011

[46] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimization problems", *International Journal of Mathematical Modelling and Numerical Optimization*, Vol. 4, No. 2, pp. 150–194, 2013.

[47] B.R. Adarsh, T. Raghunathan, T. Jayabarathi, X.S. Yang, "Economic dispatch using chaotic bat algorithm" Energy, 2016 pp. 666- 675.

[48] Chawla, Mridul, and M. Duhan. "Bat algorithm: a Survey of the State-of-the-Art ", *Applied Artificial Intelligence,* 2015

[49] R. K. Swain, N.C. Sahu, and P.K. Hota, “Gravitational search algorithm for optimal economic dispatch”, *Procedia technology*, vol. 6, pp. 411-419, January 2012.

[50] T.-S. Pan, T.-K. Dao and S.-C. Chu, “Hybrid particle swarm optimization with Bat algorithm”, *Advances in intelligent systems and computing*, pp.37-47, 2015.

[51] R. Rajabioum, “Cuckoo optimization algorithm, “*Applied Soft Computing*, pp.5508-5518, November 2011.

[52] R. Kaur and G.K. Gill, “Solution to economic load dispatch problem using cuckoo search algorithm”, *International Journal of Electrical and Electronics Research,* Vol 3, pp.362-369, 2015.

[53] M. Tuba, M. Subotic, N. Stanarevic “Modified cuckoo search algorithm for unconstrained optimization problems”, *Proceedings of the European Computing Conference (ECC ’11),* pp. 263-268, Paris, France, April 2011.

[54] A.B.S. Serapiao, “cuckoo search for solving economic load dispatch problem*”, Intelligent control and automation*, pp.385- 390, 2013.

[55] N. Dieu, P. Schegner, W. Ongsakul, “Cuckoo search algorithm for non-convex economic dispatch” *The Institution of Engineering and Technology*, Vol. 7, pp. 645–654, 2013.

[56] A.B. Bohler and S. Branke, “Multi swarm hybrid for multi-modal optimization”, *world congress on computational intelligence, Australia*, pp.1759-1766, 2012.

APPENDICES

Appendix A: Hybrid PSO-BA code in MATLAB: case of 13 generating units

```
tic
clc
clear all
close all

LB = [0 0 0 60 60 60 60 60 60 40 40 55 55]; %lower bounds of
variables
UB = [680 360 360 180 180 180 180 180 180 120 120 120 120];
%upper bounds of variables
% PSO parameters values
M = 13;             % number of variables
n = 100;           % population size
wmax = 0.9;        % inertia weight
wmin = 0.4;        % inertia weight
c1 = 2;            % acceleration factor
c2 = 2;            % acceleration factor
A = 0.9;
r = 0.1;
Qmin = 0;          % Frequency minimum
Qmax = 2;          % Frequency maximum


% Start PSO main program
maxite = 150000;    % set maximum number of iteration

maxrun = 25;       % set maximum number of runs to 25
for run = 1: maxrun

    % Start PSO initialization
```

```
for i=1:n
    for j=1:m
        x0(i,j)=round(LB(j)+rand()*(UB(j)-LB(j)));
    end
end
x = x0;       % initial population
v=0.1*x0;    % initial velocity
for i=1:n
    f0(i,1)= fn_ELD_13(x0(i,:));
end
[fmin0,index0]= min (f0);
Pbest = x0;              % initial pbest
Gbest = x0(index0,:);     % initial gbest
% End PSO initialization

% Start PSO algorithm
Ite = 1;
tolerance = 1;
while ite <= maxite && tolerance>10^-12

   w = wmax-(wmax-wmin)*ite/maxite; % update inertial weight

    % PSO velocity updates
    for i=1:n
        for j=1:m

 Q(i)=Qmin+(Qmin-Qmax)*rand;
 v1(i,j)=w*v(i,j)+(x(i,j)-pbest(i,j))*Q(i);
 v2(i,j)=w*v(i,j)+c1*rand()*(pbest(i,j)-x(i,j))...
          +c2*rand()*(gbest(1,j)-x(i,j));
 v(i,j)=(v1(i,j)+v2(i,j))/2;
```

```
    end
end

% PSO position update
for i=1:n
    for j=1:m
        x(i,j)=x(i,j)+ v(i,j);
    end
end

% handling boundary violations
for i=1:n
    for j=1:m
        if x(i,j)<LB(j)
            x(i,j)=LB(j);
        elseif x(i,j)>UB(j)
            x(i,j)=UB(j);
        end
    end
end
% evaluating fitness
for i=1:n
    f(i,1)=fn_ELD_13(x(i,:));
end
% updating pbest and fitness
for i=1:n
    if f(i,1)<f0(i,1)
        pbest(i,:)=x(i,:);
        f0(i,1)=f(i,1);
    end
end
[fmin,index]=min(f0);    % finding out the best particle
```

```
        ffmin(ite,run)= fmin;     % storing best fitness
        ffite(run) = ite;          % storing iteration count
        % updating gbest and best fitness
        if fmin < fmin0
            gbest = pbest(index,:);
            fmin0 = fmin;
        end
        % calculating tolerance
        if ite>100
            tolerance = abs(ffmin(ite-100,run)-fmin0);
        end
        % displaying iterative results
        if ite == 1
   disp(sprintf('Iteration    Best particle    Objective fun'));
        end
        disp(sprintf('%8g  %8g
%8.4f',ite,index,fmin0));
        ite = ite+1;
    end
    % End PSO algorithm
    gbest;
      fvalue =(0.00028*gbest(1)^2 + 8.1*gbest(1) + 550)+
abs(300*sin(0.035*(0 - gbest(1))))+ (0.000560*gbest(2)^2 +
8.1*gbest(2) + 309)+ abs(200*sin(0.042*(0 - gbest(2))))...
              + (0.00056*gbest(3)^2 + 8.1*gbest(3) + 307)+
abs(200*sin(0.042*(0 - gbest(3))))+(0.00324*gbest(4)^2 +
7.74*gbest(4) + 240)+ abs(150*sin(0.063*(60 - gbest(4))))...
              +(0.00324*gbest(5)^2 + 7.74*gbest(5) + 240)+
abs(150*sin(0.063*(60 - gbest(5))))+(0.00324*gbest(6)^2 +
7.74*gbest(6) + 240)+ abs(150*sin(0.063*(60 - gbest(6))))...
```

```
              +(0.00324*gbest(7)^2 + 7.74*gbest(7) + 240)+
abs(150*sin(0.063*(60 - gbest(7))))+(0.00324*gbest(8)^2 +
7.74*gbest(8) + 240)+ abs(150*sin(0.063*(60 - gbest(8))))...
              +(0.00324*gbest(9)^2 + 7.74*gbest(9) + 240)+
abs(150*sin(0.063*(60 - gbest(9))))+(0.00284*gbest(10)^2 +
8.6*gbest(10) + 126)+ abs(100*sin(0.084*(40 - gbest(10))))...
              +(0.00284*gbest(11)^2 + 8.6*gbest(11) + 126)+
abs(100*sin(0.084*(40 - gbest(11))))+(0.00284*gbest(12)^2 +
8.6*gbest(12) + 126)+ abs(100*sin(0.084*(55 - gbest(12))))...
              +(0.00284*gbest(13)^2 + 8.6*gbest(13) + 126)+
abs(100*sin(0.084*(55 - gbest(13))));
     fff(run)=fvalue;
    rgbest(run,:)=gbest;
    disp(sprintf('--------------------------------------'));
end
% End PSO main program
disp(sprintf('\n'));
disp(sprintf('******************************************'));
disp(sprintf('Final Results-----------------------------'));
[total_cost,bestrun] = min(fff) ;
mean_cost = mean(fff);
median_cost = median(fff);
worst_cost = max(fff);
standardeviation_cost = std(fff);
best_variable = rgbest(bestrun,:);
disp(sprintf('best objective function value obtained(best total
cost of generators in $/hr)  ='));
disp(sprintf('  %8.4f', total_cost));
disp(sprintf('\n'));
disp(sprintf('bestrun ='));
disp(sprintf('  %8g', bestrun));
disp(sprintf('\n'));
```

```
disp(sprintf('mean total cost of generators in $/hr = '));
disp(sprintf('%8.5f',mean_cost));
disp(sprintf('\n'));
disp(sprintf('median total cost of generators in $/hr = '));
disp(sprintf('%8.5f',median_cost));
disp(sprintf('\n'));
disp(sprintf('worst total cost of generators in $/hr = '));
disp(sprintf('%8.5f',worst_cost));
disp(sprintf('\n'));
disp(sprintf('standard deviation cost of generators in $/hr =
'));
disp(sprintf('%8.5f',standarddeviation_cost));
disp(sprintf('\n'));
disp(sprintf(' the best output power for every unit in MW  ='));
disp(sprintf('  %8.4f', best_variable));
disp(sprintf('\n'));
disp(sprintf('**********************************************'));
 disp('-------------------------------------------------------');
toc
% PSO convergence characteristic
plot(ffmin(1:ffite(bestrun),bestrun),'-k');
xlabel('Iteration');
ylabel('generation cost($/hr)');
title('PSO convergence characteristic(13 units)');
```

Appendix B: Improved PSO using Cuckoo operations: case of 40 generating units

```
tic
clc
clear all
close all

LB =[36 36 60 80 47 68 110 135 135 130 94 94 125 125 125 125 220
220 242 242 254 254 254 254 254 254 10 10 10 47 60 60 60 90 90
90 25 25 25 242];          %lower bounds of variables
UB =[114 114 120 190 97 140 300 300 300 300 375 375 500 500 500
500 500 500 550 550 550 550 550 550 550 550 150 150 150 97 190
190 190 200 200 200 110 110 110 550];      %upper bounds of
variables

% PSO parameters values
m = 40;              % number of variables
n = 100;            % population size
wmax = 0.9;         % inertia weight
wmin = 0.4;         % inertia weight
c1 = 2;             % acceleration factor
c2 = 2;             % acceleration factor
pa= 0.25;

% Start PSO main program
maxite = 150000;    % set maximum number of iteration

maxrun = 25;        % set maximum number of runs to 25
for run = 1: maxrun
    % Start PSO initialization
    for i = 1:n
        for j = 1:m
```

```
            x0(i,j)= round(LB(j)+rand()*(UB(j)-LB(j)));
        end
    end
    x = x0;        % initial population
    v = 0.1*x0;    % initial velocity
    for i = 1:n
        f0(i,1 = fn_ELD_40(x0(i,:));
    end
    [fmin0,index0]= min(f0);
    Pbest = x0;                % initial pbest
    Gbest = x0(index0,:);      % initial gbest

    % End PSO initialization



    % Start PSO algorithm
    ite=1;
    tolerance = 1;
    while ite<= maxite && tolerance>10^-12

        w= wmax-(wmax-wmin)*ite/maxite; % update inertial weight

           %levy flight
        for i=1:n
          for j=1:m
   beta = 3/2;
sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^…(
(beta-1)/2)))^(1/beta);
  step = and()/abs(rand())^(1/beta);

     DY(i,j)= step*sigma*(pbest(i,j)- gbest(1,j));
       y(i,j)= x(i,j)- rand()*DY(i,j);
```

```
    end
end

 % PSO velocity updates
 for i=1:n
     for j=1:m
         v(i,j)=w*v(i,j)+c1*rand()*(pbest(i,j)-x(i,j))...
                 +c2*rand()*(gbest(1,j)-x(i,j));
     end
 end

 % PSO position update considering discovery probability
 for i=1:n
     for j=1:m
         if rand()>pa
             x(i,j)=x(i,j)+rand()*v(i,j);
        else
             x(i,j)=x(i,j)+v(i,j);
         end
     end
 end

 % handling boundary violations
 for i=1:n
     for j=1:m
         if x(i,j)< LB(j)
             x(i,j)= LB(j);
         elseif x(i,j)>UB(j)
             x(i,j)= UB(j);
         end
     end
```

```
   end

    % handling boundary violations
   for i=1:n
       for j=1:m
           if y(i,j)< LB(j)
               y(i,j)= LB(j);
           elseif x(i,j)> UB(j)
               y(i,j)= UB(j);
           end
       end
   end
% evaluating fitness
 % evaluating fitness
   for i=1:n
       f(i,1)= fn_ELD_40(x(i,:));
       g(i,1)= fn_ELD_40(x(i,:));
   end



% communication between PSO and cuckoo
   for i=1:n
       if g(i,1)< f(i,1)
           x(i,:)= y(i,:);
           f(i,1)= g(i,1);
       end
   end

   % updating pbest and fitness
   for i=1:n
       if f(i,1)< f0(i,1)
           pbest(i,:)= x(i,:);
```

```
                f0(i,1)= f(i,1);
            end
        end

        [fmin,index]= min(f0);    % finding out the best particle
        ffmin(ite,run)= fmin;     % storing best fitness fd
        ffite(run) = ite;          % storing iteration count

        % updating gbest and best fitness
        if fmin < fmin0
            gbest = pbest(index,:);
            fmin0 = fmin;
        end
        % calculating tolerance
        if ite>100
            tolerance = abs(ffmin(ite-100,run)-fmin0);
        end

        % displaying iterative results
        if ite == 1
            disp(sprintf('Iteration    Best particle
Objective fun'));
        end
        disp(sprintf('%8g  %8g
%8.4f',ite,index,fmin0));
        ite = ite+1;
    end
    % End PSO algorithm
    gbest;
     fvalue =(0.0069*gbest(1)^2 + 6.73*gbest(1) + 94.705)+
abs(100*sin(0.084*(36 - gbest(1))))+ (0.0069*gbest(2)^2 +
6.73*gbest(2) + 94.705)+ abs(100*sin(0.084*(36 - gbest(2))))...
```

```
              +(0.02028*gbest(3)^2 + 7.07*gbest(3) + 309.54)+
abs(100*sin(0.084*(60 - gbest(3))))+(0.00942*gbest(4)^2 +
8.18*gbest(4) + 369.03)+ abs(150*sin(0.063*(80 - gbest(4))))...
              +(0.0114*gbest(5)^2 + 5.35*gbest(5) + 148.89)+
abs(120*sin(0.077*(47 - gbest(5))))+(0.01142*gbest(6)^2 +
8.05*gbest(6) + 222.33)+ abs(100*sin(0.084*(68 - gbest(6))))...
              +(0.00357*gbest(7)^2 + 8.03*gbest(7) + 287.71)+
abs(200*sin(0.042*(110 - gbest(7))))+(0.00492*gbest(8)^2 +
6.99*gbest(8) + 391.98)+ abs(200*sin(0.042*(135 - gbest(8))))...
              +(0.00573*gbest(9)^2 + 6.6*gbest(9) + 455.76)+
abs(200*sin(0.042*(135 - gbest(9))))+(0.00605*gbest(10)^2 +
12.9*gbest(10) + 722.82)+ abs(200*sin(0.042*(130 -
gbest(10))))...
              +(0.00515*gbest(11)^2 + 12.9*gbest(11) + 635.2)+
abs(200*sin(0.042*(94 - gbest(11))))+(0.00569*gbest(12)^2 +
12.8*gbest(12) + 654.69)+ abs(200*sin(0.042*(94 -
gbest(12))))...
              +(0.00421*gbest(13)^2 + 12.5*gbest(13) + 913.4)+
abs(300*sin(0.035*(125 - gbest(13))))+(0.00752*gbest(14)^2 +
8.84*gbest(14) + 1760.4)+ abs(300*sin(0.035*(125 -
gbest(14))))...
              +(0.00708*gbest(15)^2 + 9.15*gbest(15) + 1728.3)+
abs(300*sin(0.035*(125 - gbest(15))))+(0.00708*gbest(16)^2 +
9.15*gbest(16) + 1728.3)+ abs(300*sin(0.035*(125 -
gbest(16))))...
              +(0.00313*gbest(17)^2 + 7.97*gbest(17) + 647.85)+
abs(300*sin(0.035*(220 - gbest(17))))+(0.00313*gbest(18)^2 +
7.95*gbest(18) + 649.69)+ abs(300*sin(0.035*(220 -
gbest(18))))...
              +(0.00313*gbest(19)^2 + 7.97*gbest(19) + 647.83)+
abs(300*sin(0.035*(242 - gbest(19))))+(0.00313*gbest(20)^2 +
```

```
7.97*gbest(20) + 647.81)+ abs(300*sin(0.035*(242 -
gbest(20))))...
              +(0.00298*gbest(21)^2 + 6.63*gbest(21) + 785.96)+
abs(300*sin(0.035*(254 - gbest(21))))+(0.00298*gbest(22)^2 +
6.63*gbest(22) + 785.96)+ abs(300*sin(0.035*(254 -
gbest(22))))...
              +(0.00284*gbest(23)^2 + 6.66*gbest(23) + 794.53)+
abs(300*sin(0.035*(254 - gbest(23))))+(0.00284*gbest(24)^2 +
6.66*gbest(24) + 794.53)+ abs(300*sin(0.035*(254 -
gbest(24))))...
              +(0.00277*gbest(25)^2 + 7.1*gbest(25) + 801.32)+
abs(300*sin(0.035*(254 - gbest(25))))+(0.00277*gbest(26)^2 +
7.1*gbest(26) + 801.32)+ abs(300*sin(0.035*(254 -
gbest(26))))...
              +(0.52124*gbest(27)^2 + 3.33*gbest(27) + 1055.1)+
abs(120*sin(0.077*(10 - gbest(27)))) +(0.52124*gbest(28)^2 +
3.33*gbest(28) + 1055.1)+ abs(120*sin(0.077*(10 -
gbest(28))))...
              +(0.52124*gbest(29)^2 + 3.33*gbest(29) + 1055.1)+
abs(120*sin(0.077*(10 - gbest(29))))+(0.0114*gbest(30)^2 +
5.35*gbest(30) + 148.89)+ abs(120*sin(0.077*(47 -
gbest(30))))...
              +(0.0016*gbest(31)^2 + 6.43*gbest(31) + 222.92)+
abs(150*sin(0.063*(60 - gbest(31))))+(0.0016*gbest(32)^2 +
6.43*gbest(32) + 222.92)+ abs(150*sin(0.063*(60 -
gbest(32))))...
              +(0.0016*gbest(33)^2 + 6.43*gbest(33) + 222.92)+
abs(150*sin(0.063*(60 - gbest(33))))+(0.0001*gbest(34)^2 +
8.95*gbest(34) + 107.87)+ abs(200*sin(0.042*(90 -
gbest(34))))...
              +(0.0001*gbest(35)^2 + 8.62*gbest(35) + 116.58)+
abs(200*sin(0.042*(90 - gbest(35))))+(0.0001*gbest(36)^2 +
```

```
8.62*gbest(36) + 116.58)+ abs(200*sin(0.042*(90 -
gbest(36))))...
              +(0.0161*gbest(37)^2 + 5.88*gbest(37) + 307.45)+
abs(80*sin(0.098*(25 - gbest(37))))+(0.0161*gbest(38)^2 +
5.88*gbest(38) + 307.45)+ abs(80*sin(0.098*(25 - gbest(38))))...
              +(0.0161*gbest(39)^2 + 5.88*gbest(39) + 307.45)+
abs(80*sin(0.098*(25 - gbest(39))))+(0.00313*gbest(40)^2 +
7.97*gbest(40) + 647.83) + abs(300*sin(0.035*(242 -
gbest(40))));


     fff(run)= fvalue;
    rgbest(run,:)= gbest;
    disp(sprintf('--------------------------------------'));
end
% End PSO main program
disp(sprintf('\n'));
disp(sprintf('******************************************'));
disp(sprintf('Final Results-----------------------------'));
[total_cost,bestrun] = min(fff) ;
mean_cost = mean(fff);
median_cost = median(fff);
worst_cost = max(fff);
standardeviation_cost = std(fff);
best_variable = rgbest(bestrun,:);
disp(sprintf('best objective function value obtained(best total
cost of generators in $/hr)  ='));
disp(sprintf('  %8.4f', total_cost));
disp(sprintf('\n'));
disp(sprintf('bestrun ='));
disp(sprintf('  %8g', bestrun));
disp(sprintf('\n'));
```

```
disp(sprintf('mean total cost of generators in $/hr = '));
disp(sprintf('%8.5f',mean_cost));
disp(sprintf('\n'));
disp(sprintf('median total cost of generators in $/hr = '));
disp(sprintf('%8.5f',median_cost));
disp(sprintf('\n'));
disp(sprintf('worst total cost of generators in $/hr = '));
disp(sprintf('%8.5f',worst_cost));
disp(sprintf('\n'));
disp(sprintf('standard deviation cost of generators in $/hr=
'));
disp(sprintf('%8.5f',standarddeviation_cost));
disp(sprintf('\n'));
disp(sprintf(' the best output power for every unit in MW  ='));
disp(sprintf('  %8.4f', best_variable));
disp(sprintf('\n'));
disp(sprintf('**********************************************'));
 disp('------------------------------------------------------');
toc
% PSO convergence characteristic
plot(ffmin(1:ffite(bestrun),bestrun),'-k');
xlabel('Iteration');
ylabel('generation cost($/hr)');
title ('PSO convergence characteristic(6 units)');
```

Appendix C: Ethical clearance certificate

UNISA college of science, engineering and technology

Dear Mr Ghislain Yanick GNINKEU TCHAPDA (58540385)

Date: 2016-11-07

Application number:
001/GYGT/2016NOV/CSET_SOE

REQUEST FOR ETHICAL CLEARANCE: (Application of improved particle swarm optimization in economic dispatch of power system)

The College of Science, Engineering and Technology's (CSET) Research and Ethics Committee has considered the relevant parts of the studies relating to the abovementioned research project and research methodology and is pleased to inform you that ethical clearance is granted for your research study as set out in your proposal and application for ethical clearance.

Therefore, involved parties may also consider ethics approval as granted. However, the permission granted must not be misconstrued as constituting an instruction from the CSET Executive or the CSET CRIC that sampled interviewees (if applicable) are compelled to take part in the research project. All interviewees retain their individual right to decide whether to participate or not.

We trust that the research will be undertaken in a manner that is respectful of the rights and integrity of those who volunteer to participate, as stipulated in the UNISA Research Ethics policy. The policy can be found at the following URL:
http://cm.unisa.ac.za/contents/departments/res_policies/docs/ResearchEthicsPolicy_apprvCounc_21Sept07.pdf

Please note that the ethical clearance is granted for the duration of this project and if you subsequently do a follow-up study that requires the use of a different research instrument, you will have to submit an addendum to this application, explaining the purpose of the follow-up study and attach the new instrument along with a comprehensive information document and consent form.

Yours sincerely

Dr. Ilunga

Prof M Ilunga
Chair: Ethics Sub-Committee (SOE) CSET

LL Jewell for LW Snyman
Name of Director
Director: School/Institute

Prof I Alderton
Executive Dean (Acting): College of Science, Engineering and Technology

University of South Africa
College of Science, Engineering and Technology
The Science Campus
C/o Christiaan de Wet Road and Pioneer Avenue, Florida Park, Roodepoort
Private Bag X6, Florida, 1710
www.unisa.ac.za/cset

UNISA college of science, engineering and technology