ELD, LOAD FLOW ANALYSIS & OPTIMAL LOAD FLOW STUDIES USING INTELLIGENT OPTIMIZATION TECHNIQUE

A thesis submitted
In fulfillment of the requirements of the degree of
Doctor of Philosophy

Submitted by

DEVINDER KUMAR
(2K13/PhD/EE/03)

Under the Supervision of

Prof. N.K JAIN
Department of Electrical Engineering
Delhi Technological University

And

Prof. UMA NANGIA
Professor
Department of Electrical Engineering
Delhi Technological University

Department of Electrical Engineering
Delhi Technological University
Delhi-110042, India

JULY, 2022

CERTIFICATE

This is to certify that the thesis entitled "**ELD, Load flow analysis & optimal load flow studies using intelligent optimization technique**" which is being submitted by **Mr. Devinder Kumar** for the award of degree of **Doctor of Philosophy in Electrical Engineering**, Delhi Technological University, Delhi, is a record of student's own work carried out by him under our supervision and guidance. The matter embodied in this thesis has not been submitted in part or full to any other university or institute for award of any degree or diploma.

Prof. N.K Jain	**Prof. Uma Nangia**
Department of Electrical Engineering	Department of Electrical Engineering
Delhi Technological University	Delhi Technological University
Delhi-110042, India	Delhi-110042, India

ACKNOWLEDGEMENT

It has been a pleasure for me to work on this thesis. First and foremost, I would like to express my sense of gratitude and indebtedness to my supervisors Dr. N.K Jain and Prof. and Head of Department Uma Nangia, Electrical Engineering Department, Delhi Technological University for their inspiring guidance, valuable suggestions, encouragement, untiring effort throughout the course of this work and help in providing necessary facilities and resources during the entire period of my research work. Their wide knowledge and logical way of thinking have helped me to complete my research work. They always made themselves available for me despite of their busy schedules and I consider it as a great opportunity to carry out my research work under their guidance and to learn from their research expertise.

I am indebted to my wife Mrs. Pavitra Khandelwal, son Daksh , my father Sh. Om Prakash and my mother late. Dhanwati Devi for their unconditional support, patience throughout my study period. They encouraged me all the time, no matter what difficulties I encountered. I would like to express my deepest regards and gratitude to my parents for their unmitigated love. I would like to express my greatest admiration to all my family members and relatives for their positive encouragement that they showered on me throughout this research work. I extend my personal thanks to my friends and colleagues for their valuable support and reminding me to complete my work at the earliest. Sincere thanks to Delhi Technological University, Govind Ballabh Pant Institute of Technology faculties and authorities for providing me necessary facilities for the smooth completion of my work.

Above all, I would like to thank The Almighty God, who has blessed me strength and support throughout my life and always provides me with a path to acquire knowledge and learning.

DEVINDER KUMAR
(2K13/PhD/EE/03)

ABSTRACT

As protection of the environment gains more and more attention, the economic emission dispatch problem has emerged as an intriguing and crucial task in the power system. In essence, the EED problem is a multi-objective optimization problem, which concurrently reduces fuel costs and pollutant emissions while also satisfying certain system constraints like power balance and generating restrictions. The thesis developed a method based on the meta-heuristic Particle swarm optimization for single objective, bi objective and multi objective power system optimization problems based on cost of fuel function, emission criterion function and operational constraints of the generating system.

In this thesis comprehensive, a systematic and chronological effort has been made for literature review from 1983 to 2019 with historical development, addition of new parameters, tuning or refinement of parameters and its variants for different optimization problems with constraints, multi-objectives. In addition it also covers the parallel PSO, its hybrids, communication topology and for multi-objective problems strategy used for parallel computing are covered in detail.

The new variant Perfectly Convergent Particle Swarm Optimization (PCPSO) developed is an intelligent algorithm which does not get trapped in local optima by using personal best value along with new parameters and new velocity update equation for better exploration in the search space. The velocity clamping effectively helped to control the maximum velocity of the particles from explosion state and align them towards the true global optimal with increased computational efficiency in less time. It has been implemented on uni-modal, multi-modal with local optima and noisy function.

PCPSO technique was used to solve combined economic and multiple emissions dispatch scenarios with max-max price penalty factor using quadratic functions, while considering the

implications of emissions. Implementing this method on three different standard IEEE test systems , such as the IEEE six-unit system, IEEE ten-unit system, and IEEE forty-unit system, and comparing the results with other meta - heuristic algorithms, allowed for the evaluation of this algorithm's effectiveness.

Moreover same strategy is used for solving combined economic and multiple emissions dispatch problems while taking into account the impacts of various pollutants with seven price penalty factors using cubic functions. Cubic cost functions are more accurate and show the actual response of all thermal units. This algorithm has better search capabilities with strong convergence characteristics that minimizes the cubic cost and cubic multiple emissions functions at various load demands with minimum transmission losses for an IEEE 30 bus,6 generators test system.

PCPSO was able to provide balanced exploration and exploitation in the search space. The suggested algorithm's effectiveness was tested on different separate test systems, both small and large, with differing degrees of complexity. In the realm of Multi Area Load Dispatch (MALD), this technique aids in the refinement of the global solution as well as local search. Energy transfer between locations and fossil fuel emissions from generating units are key concerns. As a result, the goal of MAPD is to minimize the overall generation cost of the areas while also lowering pollutant emissions. To appreciate the value of resolving the entire region into tiny regions, a comparison with the Single Area load Dispatch (SALD) method is made. The Price Penalty Factor (PPF) method is used to reduce a multi-objective optimization problem into a single-objective optimization problem while satisfying its different equality and inequality constraints. System security is ensured by keeping the tie line transfer limits between areas and the constraints in the multi area load dispatch. Various benchmark IEEE models were applied to this algorithm to test the developed PCPSO effectiveness and reliability.

TABLE OF CONTENTS

CHAPTER – I: INTRODUCTION

CHAPTER – II: LITERATURE REVIEW

CHAPTER – III: DEVELOPMENT OF NEW VERSION OF PARTICLE SWARM OPTIMIZATION

CHAPTER – IV: MACHINE LEARNING THROUGH BACK PROPAGATION NETWORKS USING PCPSO IN HIGHER DIMENSIONS

CHAPTER – V: COMBINED ECONOMIC EMISSION DISPATCH WITH QUADRATIC FUNCTION WITH POINT VALVE LOADING USINGPERFECTLY CONVERGENT PARTICLE SWARM OPTIMIZATION

CHAPTER – VI: COMBINED ECONOMIC EMISSION DISPATCH USING PERFECTLY CONVERGENT PARTICLE SWARM OPTIMIZATION

CHAPTER – VII: MULTI AREA ECONOMIC EMISSION LOAD DISPATCH USING PERFECTLY CONVERGENT PARTICLE SWARM OPTIMIZATION

LIST OF ABBREVIATIONS

ED:	Economic dispatch
ELD:	Economic load dispatch
CEED:	Combined economic emission load dispatch
GA:	Genetic algorithm
EP:	Evolutionary programming
PSO:	Particle swarm optimization
BBO:	Biogeography Based Optimization
HSABC:	Harvest season artificial bee colony
DE:	Differential evolution
BSA:	Backtracking search algorithm
GSA:	Gravitational search algorithm
ev-MOGA:	epsilon-multi-objective genetic algorithm variable
FPA:	Flower pollination algorithm
QOTLBO:	Quasi oppositional teaching learning based optimization
MABC/D/Cat: MABC/D/Log):	Modified artificial bee colony algorithm
KSO:	Kernel search Optimization
GCPSO:	Guaranteed Convergence Particle Swarm Optimization
MPSO:	Multi-start Particle Swarm Optimization
PCPSO:	Perfectly Convergent Particle Swarm Optimization
OPSO:	Opposition based Particle Swarm Optimization
ANN:	Artificial Neural Network
MLP:	Multi- layer perceptron
LMBP:	Lavenberg-Marquardt Back propagation
BRBP:	Bayesian Regularization Back Propagation
PPF:	Price Penalty Factors
FA:	Firefly algorithm
SA:	Simulated annealing
QPSO:	Quantum Particle Swarm Optimization
GOA:	Grasshopper optimization algorithm
PS:	Pattern search
NSGA-II:	Non- dominated sorting genetic algorithm
AEO:	Artificial ecosystem based optimization
CAE04:	Multi objective 4th chaotic function Artificial ecosystem based optimization
SCA:	Sine-cosine algorithm
MAED:	Multi-area economic dispatch
SAECPD:	Single area economic emission power dispatch
MAECPD :	Multi area economic emission power dispatch
VVL:	Valve point loading
RRL:	Ramp rate limit
POZ:	Prohibited operating zones
DEC2:	DE algorithm with chaotic sequences based on logistic map
HSLSO:	Hybridizing sum-local search optimizer
SQP:	Sequential quadratic programming
SSA-WSA:	Squirrel search algorithm based weighted sum approach
MOSSA:	Multi-objective squirrel search algorithm

LIST OF SYMBOLS

The main list of symbols used in this thesis is listed below. Other symbols not listed in this list are defined locally.

v_j :	Velocity of the particle j
x_j:	Position of the particle j
ω :	Inertia weight factor
c_1,c_2	Acceleration factors
p_j :	Personal best of particle j,
g:	Global best of the entire swarm
r_1,r_2	Pseudo random numbers between 0 and 1.
χ:	Constriction factor
ϕ1, ϕ2:	Phi
$v_{max,j}$:	Fraction of the search space
u_j, l_j :	Upper and Lower limits of the search space in j dimension and
λ:	Velocity Clamping percentage, is usually lie between 0 and 1
$-x_j{}'(k)$:	Resets the particle position to global best position g(k)
ρ (k) (1-2r)	Generates a random search in the neighborhood area of global best particle
ρ(k) :	Diameter of random search space
ϵ:	Stagnation threshold
J:	Jacobin matrix
x_k :	Current Weight and Biases,
e:	Network errors,
μ:	Scalar value
I:	Identity matrix.
F_i:	Input function value
F_i' :	Target or Output values of neural network

N: Size of training data set

e_i: Error due to difference between input and output of the neural network.

F_{CT}: Fuel cost of all generators in \$/h,

P_i: Real output power in MW of i_{th} generator

P_D , P_L: Total demand and Transmission losses in MW

$P_{i\,min}, P_{i\,max}$: Minimum and maximum power limits of i_{th} generator,

N: Number of committed generating units

a_i, b_i, c_i, d_i: Fuel cost curve co-efficient of the i_{th} generators

.B_{ij}: Matrix of transmission loss coefficient of generating units.

E_T: Total Emission with valve loading effect in ton/h,

$d_i, e_{i,}, f_i$: Coefficients of emission in ton/MW^2h, ton/MWh and ton/h

γ_i, δ_i: Valve point loading effect emission coefficient of i^{th}generating unit.

h_1 : Min-Max price penalty factor

$E_{i(SO2)}(P_i), E_{i(NOx)}(P_i), E_{i(CO2)}(P_i)$: Total SO2, NOx, and CO2 emissions in Kg/h.

E_t: Total Emission with valve loading effect in ton/h

$F_{T\,so2}, F_{T\,NoX}, F_{T\,Co2}$: Total fuel cost with SO2 Emission, Total fuel cost with NOx Emission, Total fuel cost with CO2 Emission,

$F_{T\,CEED}$: Total fuel cost with all Emissions

P_{Tm}: Power transfer on 'l' tie lines between the areas in MW

M: Number of interconnected areas

a_{mn}, b_{mn}, c_{mn}: Cost coefficients of n^{th}generator in m^{th} area

P_{mn}: Real output power by the n^{th} generator in m^{th} area in MW

d_{mn}, e_{mn}, f_{mn}: Coefficients of emission of n^{th}generating unit in m^{th} area ton/MW^2h, ton/MWh and ton/h

$P_{TLmi,}q_{mi}$: Tie line power and transmission coefficient for cost of transmission of power from m^{th}area to i^{th} area

$B_{mnj}, B_{m0n}, B_{m00}$: Transmission loss coefficients

ρ_{im}: Fractional loss rate,

P_{TLim}: Power flow from m^{th} area to i^{th} area and power flow from i^{th} area to m^{th}area

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 General

In the civilized era, electrical power is crucial to addressing a diversity of requirements. In order to meet the demand for power, it is crucial that the electrical power generated is transmitted and distributed effectively. There are numerous techniques to produce electrical electricity. The much more important challenge in managing and running a power generation system involves effectively managing each generator to fulfill the necessary demands. For a specific load requirement, the strategy for the most cost-effective operation of the committed units is determined. The emissions from thermal power plants must be reduced together with the operating costs due to the negative effects on the environment. As more power is produced, the unit's emissions and operating expenses increase. These two goals change nonlinearly in relation to the production of unit power. The first set of constraints is that each dedicated unit must produce power within its own minimum and maximum possible bounds. Additionally, the total power produced by each unit must only be sufficient to meet the entire load demand including system transmission losses. Another constraint is provided by the power balance. Reduced emissions are produced as fuel quality is better, albeit at a higher cost. As a result, it is impossible to achieve both cheap cost and low emission concurrently. As a result, the combined economic emission dispatch problem has nonlinear constraints and multi-objective in character.

In today's deregulated electricity market, any network operator's top priority is to find a good strategy for economic dispatch. To address and optimize economic load dispatch (ELD) difficulties, numerous proprietary and conventional methods are utilized. Even while these approaches assume that the cost curve for generator units is increasing linearly, it is actually rather nonlinear.

When two or even more generators are utilized to generate electricity and their combined capacity is greater than the amount of energy required, ELD is used in the power system. ELD responds with the amount of power required from each generator unit to meet all requirements while using the least amount of fuel possible.

How well these approaches correspond to the actual circumstance is another crucial issue. An illustration would be the fact that, in the real world, not all power plans are situated close to the load centers. Fuel prices may vary as a result of this. Numerous scheduling techniques are used to avoid this situation. Modern power systems are also interconnected, making them part of a network of connected power systems. In order to create the best permutation, it is crucial to take the requirements of the entire grid and limits brought about by the grid into account. How all these techniques respond to adjustments in the characteristics and limitations, which are crucial in the grid situation, is also crucial.

Combined economic emission load dispatch is a highly nonlinear, non-convex, discontinuous, and non-differentiable optimization problem in which the objective function may have more than one local minimum. Numerous strategies, including traditional and stochastic ones, have been developed to address the optimizing of ELD. The main issue with economic dispatch in the power system is the choice of heuristic optimization technique.

Analysis and optimization of carbon emissions are crucial since carbon footprints were created to encourage commercial and industrial organization to ensure extremely low emissions. This makes it crucial for any issues with economic dispatch to take the carbon emissions into account. When we evaluate the optimum fuel cost and the costs associated with emissions, the system is frequently forced to adopt a non-optimum cost option. The economic dispatch problem must therefore be changed to the economic and emission dispatch problem. Power companies aim for the most economical operation possible due to the increasing cost of fuels globally, the rising demand for electricity, and the growing concern for the environment. We are also taking combined economic dispatch and emission dispatch into account in our study using new developed variant of Particle swarm optimization.

1.2 Economic Dispatch

Another issue in a power system that requires top priority is the Economic Dispatch (ED) optimization problem. This involves a thorough plan for the output from each associated generator in a power system so that the entire cost is kept as low as possible, all load demand needs are satisfied, and the desired quality is guaranteed. Conventionally, electrical power systems are managed to keep system restrictions while lowering operating costs. Fuel, labor, supplies, and maintenance costs are all included in the total cost of operating a generator. Since the costs of lab our, supplies, and maintenance typically represent set percentages of the cost of fuel, we will simply consider the cost of fuel to be the only variable cost.

As regards to the power generated vs generator capacity, the cost of fuel is typically assumed to be a smooth quadratic function. Such approach, though, has a number of quite suppositions, some of which have been stated under.

Following is a list of a few of those.

i. It presumes a smooth cost function—in a real-time setting, this expression would require many factors and be more complex.

ii. It makes the assumption that the issue is static; however, the actual instance of such a power system involves several dynamic changes.

iii. This curve does not take into account emissions variables that are a consequence of the thermal plant's operation and pay the costs of pollutants in accordance with the carbon credit regulations.

iv. Complete disregard is given to the start-up and shutdown expenses of generators, that account for a sizeable portion.

The ED issue can be resolved utilizing native techniques such lambda iteration, gradient search, base point, participation factor approach, dynamic programming, linear programming, etc. if the fuel-cost curve is assumed to be a smooth curve. Such presumptions were extremely unrealistic due to the factors listed above and the valve point loading effect that is described beneath.

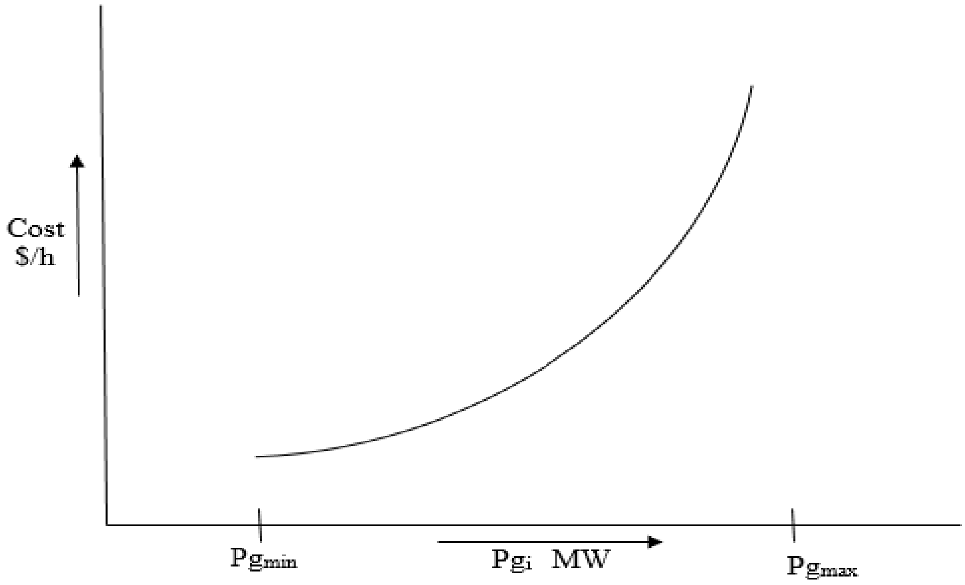


Fig 1.2 Fuel cost curve

1.3 Non Smooth characteristics

If valve point effects were taken into account in such a real-world scenario, the objective function of an ED problem contains non-differentiable regions. As a result, the effect appears to be produced in numerous steps and the goal function is created as a composition of a number of non-smooth cost functions. Primarily for two key scenarios, non-smooth cost functions are therefore considered. The valve-point loading (VPL) scenario is the first instance where the goal function is treated as the superposition of quadratic and sinusoidal functions. The objective function in the second situation, which involves considering the identical problem as a multiple-fuel problem, can be represented as either bitwise or piecewise quadratic cost functions. The responses in both of the aforementioned situations include several minimum.

1.4 **Non Smooth characteristics with valve point effects**

The input-output curve of a generator with multiple valve steam turbines would differ from the smooth cost function due to the effects of the valves. This main effect being brought on by waves produced as steam valves are opened. The prior quadratic cost functions are being supplemented with a number of sinusoidal functions to account for the effects of valve-points on generators.

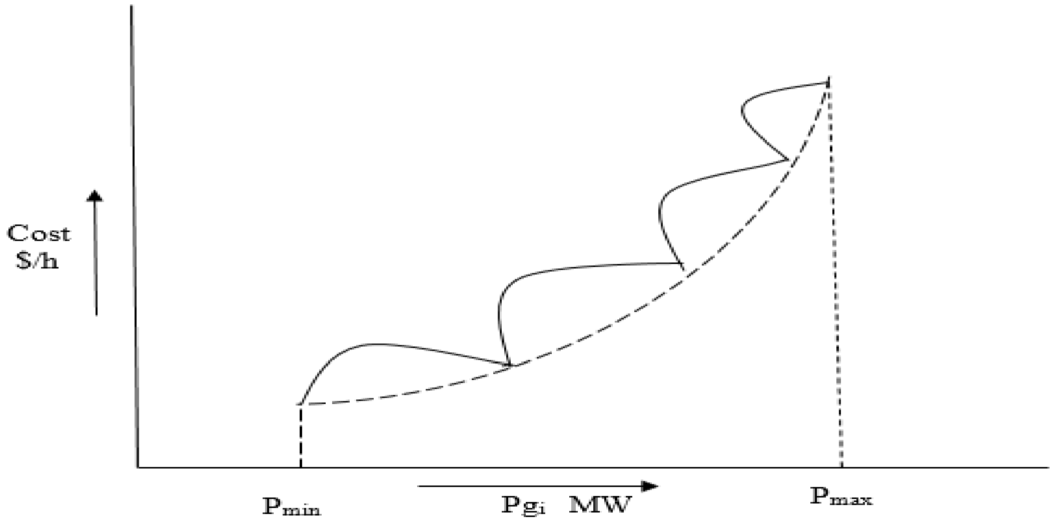


Fig 1.4 Discontinuities in Fuel cost curve due to Valve point loading effects.

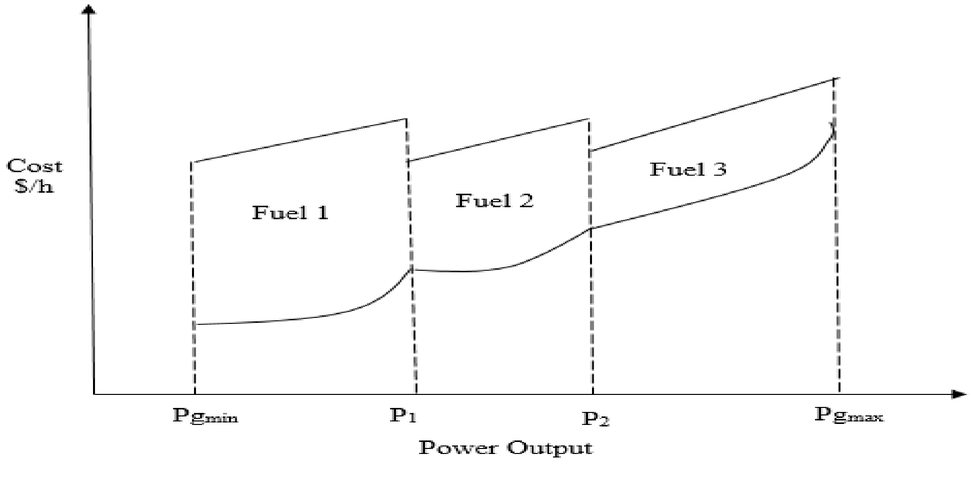


Fig 1.5 Quadratic piecewise cost function

In this instance, the generator uses a variety of fuels. The optimization problem could then be written as bitwise or piecewise quadratic cost functions.

1.5 Emission Dispatch

Fossil fuels are now used as sources of energy to produce mechanical power that is applied to generating units' rotor shafts. This results in significant emissions of nitrogen oxides, sulphur oxides, and carbon dioxide, which pollute the atmosphere. Environmental regulation enforcement and emission control have received a great deal of attention due to the environmental damage caused by fossil-fueled generating units. Additionally, the utilities have been compelled to change their design or operational tactics to reduce carbon emissions and atmospheric emissions from the thermal power plants as a result of rising public awareness of environmental degradation and the passing of the Clean Air Act Amendments in 1990. Additionally, there are many stringent regulations governing carbon credits.

1.6 Combined economic emission load dispatch (CEED)

The optimization problem for the economic dispatch needs to take into account fuel and emission optimization in order to get global optima, as mentioned above. As a result, the entire issue is now a financial and emissions-related dispatch issue. The previous approaches are no more capable of offering an acceptable solution to economic dispatch with integration of all these aspects, such as valve-point-effect and pollution optimization. This is because the economic and emission dispatch issue evolves into a multi-objective, highly multidimensional, non-convex challenge. Consequently, modern meta-heuristic techniques have been applied to address the economic and emission dispatch problems. Few of the algorithms that are extensively used are the genetic algorithm (GA), evolutionary programming(EP), particle swarm optimization(PSO), Biogeography Based Optimization(BBO) , harvest season artificial bee colony, differential evolution(DE), Backtracking search algorithm(BSA), Gravitational search algorithm(GSA), epsilon-multi-objective genetic algorithm variable(ev-MOGA), Flower pollination algorithm(FPA), quasi oppositional teaching learning based optimization(QOTLBO), modified artificial bee colony algorithm (MABC/D/Cat , MABC/D/Log) , Kernel search Optimization (KSO)

The most of the economic and emission dispatch issues are solved utilizing such methods or their combination. However, there are indeed issues in this domain. These are some of the other difficulties.

I Large processing time: The majority of techniques have a high computational cost, which is troublesome when taking into account dynamical system changes.

ii. Converging to a local optimal solution: In several cases, the problem's solution converges into a local minimum, which is then treated as the problem's optimum. The global optimal solution, however, might not be the same.

Iii Solutions that are not practical: This possibility exists and is more frequent whenever pollutants are being restricted.

iv. Algorithm failures for large- and moderate systems: The technique could breakdown and produce incorrect outputs, which is much more serious than any other issue if all the situations or use situations are still not handled appropriately and accurately.

Because of this, it is frequently required to use combinational techniques or new methods. Therefore, current research is focused on boosting the performance of solutions and resolving all the issues mentioned earlier. It's worth noting that research has shown that hybridising tactics increases their effectiveness.

1.7 Outline of Thesis

This thesis consists of seven chapters including introduction, literature review, development of new technique of particle swarm optimization, Machine Learning through Back Propagation networks using PCPSO in higher dimensions, combined economic emission dispatch with quadratic function with point valve loading using perfectly convergent particle swarm optimization, Multi-area Economic Emission load dispatch using perfectly convergent Particle swarm optimization and conclusion and future scope followed by references.

Chapter1: This chapter provides the insight into the economic dispatch, Non Smooth characteristics, Non Smooth characteristics with valve point effects, emission dispatch, Combined economic emission load dispatch with the meta-heurastic optimization algorithms.

Chapter2: Literature review is presented in this chapter from the development of the technique. The entire journey of Particle swarm optimization is presented in a systematic and chronological effort has been made for literature review from 1983 to 2019 which fits the need of researcher from starting with all the developments like historical development, addition of new parameters, tuning or refinement of parameters and its variants for different optimization problems with constraints, multi-objectives. In addition, this also covers the literature survey of parallel PSO, its hybrids, communication topology and for multi-objective problems strategy used for parallel computing is covered in detail. This review will give me a correct direction for development of new version.

Chapter3: Development of new version of Particle swarm optimization is represented in this chapter and it is tested on the various benchmark functions including uni-modal, multi modal and multi modal with noisy environment.

Chapter4: This chapter focus on the machine learning on the data sets are taken from the experiments carried out on benchmark functions using opposition based Particle swarm optimization with Cauchy mutation (OPSO) and Perfectly convergent Particle swarm optimization (PCPSO) as a set of input vectors and there corresponding output values. Artificial Neural Networks gives us various methods to learn the mapping of any data sets

Chapter5: This study presents perfectly convergent particle swarm optimization (PCPSO) for solving combined economic and multiple emissions dispatch scenarios with max-max price penalty factor using quadratic functions, while considering the implications of emissions. Implementing this method on three different standard IEEE test systems , such as the IEEE six-unit system, IEEE ten-unit system, and IEEE forty-unit system, and comparing the results with other meta - heuristic algorithms, allowed for the evaluation of this algorithm's effectiveness.

Chapter6: This chapter presents perfectly convergent Particle swarm optimization (PCPSO) for solving combined economic and multiple emissions dispatch problems while taking into account the impacts of various pollutants with seven price penalty factors using cubic cost and emission functions. Also results are compared with latest meta-heuristic techniques.

Chapter7: In this chapter Multi Area Economic Emission Dispatch (MAEED), this technique aids in the refinement of the global solution as well as local search. Energy transfer between locations and fossil fuel emissions from generating units are key concerns. As a result, the goal of MAEED is to minimize the actual generation cost of the areas while also lowering pollutant emissions.

Chapter8: This chapter presents the conclusion of research work in this thesis and future scope of the research work is discussed in brief.

CHAPTER2.

LITERATURE REVIEW OF PARTICLE SWARM OPTIMIZATION

2.1 Introduction

Nature computing paradigms are the correct way to solve real world problems due to dynamic in nature, noisy or multi dimension problems. Many species solve their complex tasks in nature by (PSO) emerged from biological research and simulation on swarming animals. Reeves [1] in 1983 firstly attempted in computer animation to show natural phenomena by generating moving particles to predefined locations with initial velocity and having characteristics like texture, colour, lifetime and angular movements. It was widely used for special effects and natural looking. The first computer simulation and movies related work was given by Craig Reynolds [2] about simulating bird swarms in 1986. He also added orientation and communication in them. The result was some simulated swarm of who's the individual he called as Boids [3] directed by three simple rules. Heppner F and Grenander [4] in 1990 reviewed Reynolds work for doing more detailed bird flock animation and studies. They introduced the concept of need of roosting or swarming, which results in too realistic nature like. Hoffmeye J[5] a biologist in 1994 studied the semiotics and defined the swarm in concept of algorithm. This algorithm being in Ontogeny group are co-operative in nature.

2.2 **Particle Swarm Optimization and its Refinement:**

Kennedy and Eberhart [6] in 1995 sought to extend the work of Reynolds to reflect social behaviour in swarms and make a realistic goal oriented. It uses simply mimic swarm behaviour in birds flocking, fish schooling or swarms of bees in solving optimization problems. Another version" Lbest" of PSO developed by Eberhart and Kennedy [7] in order to examine how changes in the paradigm effects the number of iterations to meet the error criterion. Eberhart et al. [8] in 1996 proposed a velocity clamping strategy to control the velocity of the particles. Kennedy [9] in 1997 performed analysis of PSO algorithm for social interaction with new four types of models (full model, cognition only, social only and selfless model) and suggested cognition only and social only performed well but did not want to replace the core algorithm as it will result in premature convergence. Kennedy and Eberhart [10] suggested in 1997 a first discrete binary version of PSO, by changing the velocity in each vector with probability of each bit by zero or one value. In starting PSO was without inertia weight but in 1998, Shi and Eberhart [11] initially introduced constant inertia weight to balance between exploration and exploitation and this algorithm was called standard PSO.PSO is initialized by initial solutions of the particles moving in the search space, each particle is represented by a position and velocity and keeps updating as follows

$$x_j(k+1) = x_j(k) + v_j(k+1) \tag{2.1}$$

$$v_j(k+1) = \omega v_j(k) + c_1 r_1\big(p_j(k) - x_j(k)\big) + c_2 r_2\big(g(k) - x_j(k)\big) \tag{2.2}$$

Where, j =1, 2, 3…i

k+1 denotes next iteration, k is the current iteration number, $v_j$is velocity of the particle j, $x_j$is position of the particle j, $\omega$is Inertia weight factor, c_1,c_2 are acceleration factors, $p_j$is personal best of particle j, g is the global best of the entire swarm, r_1,r_2 are pseudo random numbers between 0 and 1.

Ozcan and Mohan [12, 13] in 1998, 1999 observed that the particles surf on sine wave instead of flying and made many simplifications and removed the stochastic element of the algorithm. The optimal strategy of using inertia weight w by Eberhart and Shi [14] in 1999 from 0.9 to 0.4 in a linearly decreasing way improved exploration and optimal global minimum. Maurice Clerc [15] in 1999, modified the PSO by derived a constraint coefficient which operates without Vmax. Kennedy [16] in1999 worked on four topologies: circles, wheels, stars and random edges and concluded that topology affects the swarm's performance but also dependent on objective function. Suganthan [17] in 1999 introduced linear and non-linear methods that determine the inertia weight depending on iteration number. Eberhart and Shi [18] in 2000 suggested two approaches using constriction factor χ and inertia weight ω are used and found to be mathematically equivalent and the initial value of ω to be set at 0.9 and reducing it linearly to 0.4 for better exploration and exploitation. Carlisle and Dozier [19] in 2000, proposed this technique to forget the former experience by periodically resetting particles memory and replacing their best fitness value and position with the current position and fitness value. Kennedy [20] in 2000, performed cluster analysis of best performer particles but with high computational cost and real time. Van Den Berg and Engelbrecht [21] in 2001 proposed optimal size of swarm depends on the problem and larger swarm size when the search space is more complex. Carlisle and Dozier [22, 24] in 2001, 2002 proposed to deploy special particles called 'sentry' to monitoring the environment change and inform all other particles to reset memory. It slows down the particles and difficult to track moving optimum. In 2001 Carlisle and Dozier [23] suggested to start with constricted model with a global neighbourhood of 30 particles which are updated asynchronously but without improve the optimizer from initial configuration. Later in 2002 Clerc and Kennedy [25] suggested the use of constriction factor χ, which alleviates the requirement of velocity clamping shows better results with new velocity equation as:

$$v_j(k+1) = \chi[v_j(k) + \phi_1 \left(p_j(k) - x_j(k)\right) + \phi_2 \left(g(k) - x_j(k)\right)] \qquad (2.3)$$

$\mathrm{X} = \frac{2}{\left|2-\phi-\sqrt{\phi^2-4\phi}\right|}$ where $\phi = \phi1 + \phi2$, $\phi > 4$(4)

Ratnaweera et al.[26] in 2002, proposed to decrease c1 linearly with time while c2 to be increased linearly. Kennedy and Mendes [27] in 2002 suggested local version of PSO with inertia weight and the other with constriction factor and suggested a small neighbourhood is suitable for complex problems and large for simple problems. Van Den Berg and Engelbrecht

[28] in 2002 proposed Guaranteed Convergence PSO (GCPSO) for avoiding swarm stagnation problem with different velocity update equation, within a radius with random search around global position. Fan [29] in 2002 introduced an adaptive scaling term as an efficient speedup strategy for better control and convergence. Trelea [30] in 2003analysed the convergence boundaries, convergence point and parameter setting procedure of PSO showed ω =0.6 and $\phi 1$ = $\phi 2$=1.7 results in good performance of PSO. Zheng et al.[31] in 2003 observed with increasing inertia weights was used yielding good results. Ratnaweeraet al.[32] in 2004proposed the value of $\phi 1$ is decreased and $\phi 2$ is increased for better exploration and exploitation, further the author modified by using adaptation rule to re-initialize the velocity of particles. Chatterjee and Siarry [33] in 2006 introduced non-linear variation of inertia weight with the particle old velocity for improvement in the speed of convergence and fine tuning in search space. Bratton and Kennedy [34] in 2007 defined a standard PSO as the baseline for the researchers a common grounding to work from. Arumugam et al.[35] in 2008 proposed smaller acceleration coefficients and larger inertia weights if the personal best are not comparable to global best of the particles. Chen and Zhao [36] in 2009 suggested an adaptive variable swarm size and periodic partial increasing or declining of particles in the form of ladder function. Nakagawa et al. [37] in 2009 proposed by adding a random number to the velocity of particle depending upon the distance from the global best position for the velocity control. Van Den Berg and Engelbrecht [38] in 2010 proposed a mutation operator may be used to avoid the stagnation for the local convergence problem in standard PSO. Nickabadi et al. [39] in 2011 proposed adaptive change in inertia weight during the search process and possesses a linear relationship between inertia weight in current iteration and improvement in number of particles in previous iteration. Bansal et al. [40] in 2011 proposed 15 different strategy of inertia weight in PSO and suggested chaotic inertia weight strategy is best for accuracy and random inertia weight is best for efficiency.

Engelbrecht [41] in 2012 advocated through study is to initialize particles to zero or close to zero without imposing a personal best bound. Schmitt and Wanga [42] in 2013 suggested to reinitialize the velocity in each dimension whenever there is a stagnation. Cleghorn and Engelbrecht [43] in 2014 suggested that topology does not effect the convergence boundaries but effects the speed of convergence and divergence through the first order stability analysis. Van Zyl and Engelbrecht [44] in 2015 proposed initialization strategy called seed set which is a set of randomly generated n- dimensional orthogonal unit vectors obtained from modified Gram Schmit method and is suited to high dimension problems. This strategy forces the swarm within the subspace of search space for better exploration. Bonyadi and Michalewicz [45] in 2016 investigated the behaviour of particles and suggested the oscillation patterns into four groups based on the maximum frequency of oscillations and the boundaries does not depend on the number of dimensions. Liu et al. [46] in 2016 studied the effect of 198 regular topology with 09 different number of particles and devised formulae to help to choose optimal topology parameters. Bonyadi and Michalewicz [47] in 2017 examined the relationship between the base frequency' F' and the correlation between the particle position, particles with smaller' F' will exhibit smooth trajectories and larger' F' values are prone to more oscillations with large steps between positions. Oldewage [48] in 2018 proposed the initialization strategy which forces the swarm on sub space of search spacefor exploration rather than entire search space with optimal number of seed set size depending upon dimensionality. Shi et al. [49] in 2018 proposed new strategy called Oscillatory PSO which uses a particle to drive into oscillatory trajectories for complete search space. The cognitive and social factors are made to sum to unit, thus ensuring that particles converge toward the weighted sum between current global best and particle wise best solutions. Inertia weight is selected to ensure a complex roots are obtained from PSO update equation. Oldewage et al. [50] in 2019 investigated different particle swarm movement patterns behaviour are highly influenced by inertia weight and acceleration coefficients in higher dimensions space. Parameter configuration with inertia weight ω=0.9694 and c1=c2=0.099381 gives smooth particle trajectories and restrict unwanted roaming behaviour due to initial velocity explosion in range even in higher dimensions. Sun et al.[51] in 2019 proposed the random sampling of control parameters with immediate particle updating strategy is used along with stochastic correction approach on each dimension is also used to take information from other particles for better convergence rate and accuracy.

2.3 **Hybrids of Particle swarm optimization**

Hybrid models of PSO has been developed whose objective is to combine the good properties of different algorithms to mitigate their individual weakness. Following are the different PSO based hybrid approaches which is used to refine the properties of PSO to achieve global values.

Discrete PSO-For binary problems first discrete version [52] was developed by Kennedy and Eberhart in 1997 by changing the velocity to the probability of each bit being in one state or other.

GA Selection based PSO-Angeline [53] in 1998, applied GA tournament based selection criterion which replaced velocity and position of worst performing particle with good performers particles velocity and position. This improved the local search capabilities of PSO.

Fuzzy PSO-Shi and Eberhart [54] in 2001 suggested this algorithm where PSO is with an explicit selection procedure. Here replication, mutation, reproduction, evaluation and selection operation are employed.

PSO-GA-Lovbjerg et al.[55] Presented in 2001 introduced breeding between particles in different sub populations which result in faster convergence and a better optimal solutions.

Dissipative PSO-In 2002 Xie et al.[56] suggested this algorithm to overcome the problem of entrapped in local minima, a dissipative system was implemented which uses negative entropy and produces craziness between particles resulting to come out of stagnant stage.

Multi-phase Discrete PSO- ln 2002 Al-Kazemi and Mohan[57] suggested this hybrid version to fewer objective function with small swarm by using three coefficients whose values were set either 1 or-1 depending upon the phase of optimization and personal best was replaced with the previous position.

Evolutionary PSO-Miranda and Fonseca[58] proposed this algorithm in 2002 having hybrid characteristics of EA and of PSO. In EPSO there is replication, mutation, reproduction, evaluation and selection of particles which generates new solutions in search space.

Attractive-Repulsive PSO-This algorithm[59] was suggested by Riget et al. in 2002 which has two phases as attractive and repulsive with two operators as addition and subtraction which are used to update PSO equations to avoid premature convergence.

Niche PSO-In 2002 Britis et al. [60] devised a technique in which GCPSO is run and only those particles are separated out as multiple sub swarms whose fitness do not show any change during running of algorithm. These multiple sub swarms explore and exploit the search space simultaneous with the PSO.

Spatial Extension PSO-Krink et al. [61] in 2002 proposed three strategies: Random direction changer, Realistic bounce and Random velocity changer to avoid collision system. The last two strategies were helpful in multi-modal functions.

Stretching PSO(SPSO) –Parsopoulos and Vrahatis [62] in 2002 suggested to use deflection, stretching and repulsion technique were used in original PSO. First two techniques apply the concept transforming the objective function by incorporating the already found minimum points and repulsion technique avoids the particle to go to already found minimum points. So more global minimum points can be found out.

NBest PSO-Brits et al. [63] in 2002 proposed to use local neighbourhoods based on spatial proximity and got a parallel niching effect in a swarm.

Barebone PSO (BBPSO)-Kennedy [64] in 2003 proposed that the velocity and position update rules are substituted by a procedure that samples a parametric probability density function.

Gaussian PSO(GPSO)-A new hybrid algorithm was developed by Secrest et.al [65]in 2003which is based on probability distribution of the moving swarm with a Gaussian distance from the global and local best.

Fitness to Distance Ratio PSO -Peram etal. [66] in 2003 suggested a new concept in the algorithm each particle tracks that particle in the neighbourhood having better fitness value instead of attracting towards global best particle. It uses the relative fitness and the distance of other particles for direction purpose.

Dynamic Double PSO (DDPSO) -In 2004 Cui [67] et al. proposed a way for guarantied convergence to global minima using convergence analysis and position of particles are set dynamically with constraints.

Fully Informed PSO (FIPS) -Mendes et al. [68] developed a strategy in 2004 about a topology that all the particles are equally informed.

Hybrid Gradient Descent PSO-Noel et al. proposed a method [69] in 2004 for using the gradient information for faster convergence to global minima by employing random size and avoiding calculations of local neighbourhood.

Quantum Delta PSO-Sun et al.[70] developed an algorithm in 2004 where the concept is taken from quantum physics which obeys uncertainty principle (position and velocity cannot be measured simultaneously) but the motion of particles is quantum in nature and it has only one parameter to control.

Unified PSO(UPSO)-Parsopoulos and Vrahatis [71] in 2004 proposed a technique which only uses the features of Gbest and Lbest and its velocity updating is done in two parts depending on the information.

Co-operative PSO(CPSO)-Van den berg and Engelbrecht [72] in 2004 proposed to use co-operative behaviour to improve original PSO and uses multiple swarms to optimize the different component of the solution vector.

Species based PSO (SPSP) - Li [73] in 2004 proposed on the bases in there similarity of species of sub-populations of swarm where each species is grouped around a dominating particle called species seed. At every iteration species seed are identified and taken as neighbourhood best for the species group .Finally after many successive iteration this algorithm provides many local minima from which global minima can be identified.

Kalman PSO (KPSO) –Monson and Seppi [74] in 2004 suggested to use Kalman filter to update the particle position and thus enhances exploration without effecting the ability to converge rapidly to good optimal solutions.

Parallel PSO (PPSO)- Chang et al.[75] suggested in 2005 a technique in which fitness was calculated of each particle independently and efficiency of strategy communication was calculated. In correct strategy was given sub optimal and correlation was unknown, a hybrid strategy is considered to be good.

Angle Modulated PSO (AMPSO) -Again in 2005, Pam para et al. [76] developed an algorithm in which a bit string is generated by using trigonometric functions. It changes high dimension problem to four dimension problem and makes it highly efficient in operation and also saves memory.

Exploring Extended PSO (EEPSO) -In 2005 Poli et al. [77] proposed that Genetic Programming can be used routinely evolve specialist position update for use by PSO in special domain problems.

Hierarchical PSO-Stefan and Martin [78] in 2005 introduced a new method in which particles are arranged in a hierarchical way depending on the fitness value. Good particles are on the top of hierarchy and have more, influence on the swarm.

New PSO(NPSO)-Yang and Simon [79] in 2005 suggested that each particle adjust position from its own previous worse position and its group previous worst to obtain optimal fitness solution. No changes are made in velocity and position equations, only the term uses is worst position rather best position. This technique tries to get away from the worst instead of coming closer to best position.

Interactive PSO-Madar et al. proposed in 2005 that procedure of IPSO [80] is same that of PSO with a difference that best particle selection is done by the user in every iteration.Interactive PSO is different from Interactive EC from information sharing point of view.

Neural PSO-Duo et al. suggested a technique [81] in 2005 to combine feed forward neural network with PSO to acquire good learning in movements by the best particles previously in the search space.

Perturbation PSO-Yaun et al. in 2005 developed an algorithm [82] it keeps on changing velocity and position equations keeping the existing equations of PSO for other particles.

Principal component PSO (PCPSO)-Voss et al .in 2005 propose a strategy [83]in which particles are flown simultaneously in two different dimension search spaces to reduce the time in higher dimensions.

Opposition based PSO (OPSO)-Tizhoosh in 2005 proposed a novel concept [84] by considering counter estimates, opposite numbers, anti-chromosomes, counter actions, and opposite's weights in machine learning algorithms has proven to be effective method by making revolutionary jumps in starting as time saving.

Fuzzy Adaptive Turbulence PSO (FATPSO) -Hongbo et al. [85] Proposed in 2005 that premature convergence can be effectively avoided by using minimum threshold velocity to control the velocity parameter which is tuned adaptively by fuzzy logic controller in TPSO algorithm.

Adaptive PSO Guided by Acceleration Information-In 2006 Zeng et al. [86] proposed that acceleration term is also added to the position and velocity updating equations, thus making PSO fast and efficient.

Comprehensive Learning PSO-In this technique Liang et al. [87] in 2006 the particle velocity is updated by analysing the other particle velocity history information and the diversity of swarm is maintained.

PSO with Escape Velocity-A novel technique[88] is proposed by Zang et al. in 2006 equips the particles with the escape velocity to avoid to get trapped in local minima and increase the diversity of population which outperforms PSO for high dimension and multi modal problems.

Genetic PSO-Yin [89] in 2006 proposed this novel technique which incorporates crossover and mutation features of GA in PSO

Genetic Binary PSO-Sandri et al. in 2006 proposed [90]keeping the dynamic conditions of the swarm, in binary state each particle is treated as chromosome and the chain with the size of dimension.

Gregarious PSO-IN 2006 Pasupelti and Battiti [91] suggested a new technique in which particles uses only social knowledge and stochastic velocity vector is used in the search space. Self-setting of parameters is done by the integration to obtain the parameters.

Hybrid Discrete PSO-In 2006 Chandrasekaran et al. proposed [92] that each particle shows the job sequence as an optimal solution in job scheduling problem.

Hybrid Taguchi PSO-Roy and Ghosal in 2006 proposed [93] to select the intelligent particles only in Taguchi selection method with PSO.

Improved PSO-Zhao [94] in 2006 proposed to usePSO with Passive Congregation with harmony search and utilises a mechanism called fly-back for constraints.

Augmented Langrangian PSO-Sedlaczek and Eberhart [95] in 2006 suggested this method for equal and unequal constraints by combining augmented langrangian method with PSO for optimization problems.

Optimised PSO-Meissner et al. in 2006 proposed a technique [96] in which swarms are within a swarm to optimize the parameters of PSO and showed better optimal results.

Parallel Asynchronous PSO-Koh et al. in 2006 suggested a novel technique [97]which generates a dynamic view of load balancing along with a chain duty central approach to reduce the unbalance in load and velocity update is continuously done by the latest information.

PSO with craziness and hill climbing-In 2006 ozcan and Yilmaz [98] proposed to enhance a balance between discovery and extract by using craziness and hill climbing for optimizing multi modal functions.

Restricted velocity PSO (RVPSO)-Liu and Chen in 2006 developed [99] a restriction in velocity due to limited search space for unconstrained problems.

Self-organisation PSO-Jie [100] et al. proposed in 2006 that an extra feedback agent is required to improve swarm performance in next iteration and hence stagnation can be avoided.

Two Swarm PSO-Lie et al. [101] in 2006 suggested that two swarms are flown in different paths from each other by setting different parameters. One swarm will explore global and the other will enhance the local search by using Roulette wheel selection.

Unconstrained PSO-Moore and Venayagamoorthy [102] in 2006 proposed not of using the constraints for position and velocity equations unlike the classic form of PSO.

Velocity Limited PSO-Xu and Chen [103] in 2006 suggested in this approach that only those particles who satisfy the constraints for velocity and position are considered otherwise discarded.

Adapted Dissipative PSO-Shen et al.[104] proposed an approach in 2007 by introducing adaptive mutation and adaptive inertia weight strategy into dissipative PSO which improve the diversity of the swarm and avoids premature convergence problem.

Area extension PSO-Atyabi and Phon-Amnuaisuk [105] in 2007 introduced this technique for solving multi-robots task problem with large area by adding new elements in the equation which results in new velocity equation for correct direction,premature convergence is solved by adding hot area/zone,new credit assignment and boundary methods avoided the particles to struck in the areas and communication limitation helped to solve real world problems.

Behaviour of Distance PSO-Wang and Qian [106] in 2007 proposed that the particle changes fly behaviour guided by optimum of each particle and the optimum of the swarm and the individual can adapt themselves to search for best position more effectively.

Best Rotation PSO-Alviar et al. [107] in 2007 proposed this approach that a swarm population is divided into sub swarms and stagnation problem is avoided by forcing swarm from one local minima to another local minima and a periodic rotation is performed from particles of sub swarms which makes better exploration in search space.

Rotation Invariant PSO -Wilke et al. [108] in 2007 proposed was rotation invariant and uses random matrices rather than random diagonal matrices to perturb the direction of movement in every iteration using exponential map method.

Combi national PSO-Jarbouria et al. [109] in 2007 suggested a new clustering technique based on this algorithm with each particle is represented as a string of length' n' where i th element of the string denotes the group number assigned to the object' i 'and an integer vector corresponds to candidate solution to the clustering problem, The performance of this algorithm depends upon the choice of parameters and initial population.

Co-operative Multiple PSO (CMPSO)-Felix et al. [110] proposed in 2007 that this technique works well in multi dimensions problems and is more efficient than conventional PSO.

Dual Layer PSO (DLPSO)-Subrarnanyam et al. in 2007 [111] suggested a strategy that optimizes the neural network in an architectural layer and uses joint weights in neural network.

Dynamic and Adjustable PSO (DAPSO)-Liao et al. [112] proposed novel concept of keeping the distance of each particle to the best position is calculated to adjust the velocity of next step for the diversity of particles.

Estimation of Distribution PSO-Kulkarani and Venayagamoorthy [113] in 2007 proposed (hybrid of EDA and PSO) to introduce Estimation of Distribution algorithm which uses stochastic models to locate the optimal solution areasduring the optimization process. This increases the diversity and efficiency of PSO.

Evolutionary Iteration PSO (EIPSO) -Lee [114] in 2007 pointed a new combination of PSO and Evolutionary programming to avoid the trapping of particles in local minima and provides a strength to PSO efficiency.

Evolutionary Programming PSO (EPPSO)-Wei et al. [115] proposed in 2007 by combining the two algorithm (EP and PSO) gives more diversity among the particles to explore local and global minima and faster convergence.

Greedy PSO-Lamet al. proposed in 2007 a new concept of hybrid Evolutionary algorithm [116] of combining binary PSO with Greedy transform. The greedy transform method was successfully tested on knapsack problem.

Heuristic PSO-Lam et al.[117] proposed in 2007 that rate of convergence to local optima is faster than conventional PSO.To avoid stagnation or premature convergence near the global minima the positions of particles is randomly re-initialised and this combination of heuristic updating and position re initialization makes this algorithm powerful and efficient.

Map Reduce PSO-McNabb [118] et al. proposed in 2007 a novel technique to solve big data problems which is time consuming with conventional PSO but in fact this technique runs a parallel PSO for computationally compressed functions.

Modified Binary PSO- Yuan and Zhao [119] in 2007 suggested this strategy that the particles are produced randomly as binary vectors and the lowest value of position is used to map to permutation space.

Novel Hybrid PSO- Li and Li [120] in 2007 proposed the hybrid of PSO and Harmony search for better exploration in high dimension problems resulting in increasing exploitation of PSO.

Predator Prey PSO- Jang [121] et al. in 2007 suggested a new concept [ ] that predator follow prey and prey escapes from predator means avoiding to get trapped in local minima and move towards optimal global minima.

Quadratic Interpolation PSO (QIPSO)-Pant et al. [122] in 2007 proposed that the hybrid use of EA and PSO in which swarm leader is selected in each iteration and other partners are selected for cross over and an offspring is produced as Quadratic cross over. This new particle is accepted only if it is better than the current best particle of the swarm.

Shuffled Sub swarm PSO -Wang and Qian et al. [123] proposed in 2007 for a better diversity and performance of the swarm.

Trained PSO (TPSO)-Gheitanchi et al. [124] in 2007 proposed this technique to the ad-hoc communication networks to reduce completion complexity and time by training the particles.

2-D OTSU PSO- Wei et al. [125] in 2007 suggested to use optimal threshold selecting search with PSO for better performance. The threshold selecting method is used for image segmentation based on PSO is combined with two-dimension Otsu method.

Vertical PSO- Yang in 2007 [126] proposed that the particles can fly both to global and vertical directions to avoid stagnation or entrapped near to global points.

Opposition based PSO Cauchy Mutation- Wang et al.[127] in 2007 suggested this technique opposition based learning for every particle and dynamic Cauchy mutation is applied to the global best particle in every generation resulting in fast search speed in complex optimization problems.

Clonal PSO (CPSO)- Tan and Xia no [128] in 2007 suggested a method which clones and mutates the best particles of specific generations and then select the best one to continue evolving.

Hybrid combination of PSO and GA (HEA)-Yang [129]et al. in 2007 proposed two stage evolution strategy where evolution process is performed by PSO and diversity is maintained by GA and is used to solve three unconstrained and three constrained problems.

Active Target PSO-Zhang [130] et al. suggested in 2008 a new term as active target along with best position and previous best position for the diversity of PSO and avoiding trapped in local minima.

Adaptive Mutation PSO- Pant [131] et al. in2008proposed a new concept of using beta distribution in adaptive mutation in two forms. One form uses best individual position in a swarm and the other for the best global position.

Co-operatively Coevolving Particle Swarms- Yao [132] in 2008 suggested that the bigger problems can be braked up into smaller ones with least values such that there inter dependence cooperation is generated.

Geometric PSO (GPSO)- Moragila et al. [133] propose in 2008 about the geometric between the PSO and Evolutionary algorithm and can be used for problems of continuous and combinational spaces.

Immune PSO (IPSO) - Lin et al. [134] in 2008 suggested to improve the mutation mechanism of immune algorithm with PSO. Moreover information is conveyed as immune operator in the PSO in some cases.

Modified Genetic PSO (MGPSO) -Zhiming et al. [135] in 2008 proposed a combinational algorithm of Genetic PSO and Differential Evolution. Position updating is done by both algorithm for each particle and the better results become the reference for next position.

Orthogonal PSO (OPSO) -Ho et al. [136] in 2008 suggested to intelligent move methodfor velocity update in which divide and conquer approach is used for finding next particle position and gives better results with large problems than conventional PSO.

PursuitEscape PSO (PEPSO) -Higashtaini et al. [137] in 2008 proposed to divide the swarm in two groups as escape group and pursuit group on behaviour basis. First group results in intensification and the other group results in diversification thus making a perfect balance in algorithm.

Self-adaptive Velocity PSO (SAVPSO) -Lu and Chen [138] in 2008 investigated the impact of constraints on PSO due to its lack of knowledge of feasible solution.

Velocity Limited PSO {VLPSO) -Xu and Chen [139] in 2008 suggested to keep those particles which satisfy the constraints for velocity and position otherwise they are eliminated for further participating.

Frankenstein's PSO(FPSO)-Montes de Oca et al. [140] in 2009 proposed to combine together PSO variants to eliminate the various deficiencies .Velocity update from FIPS, inertia weight from decreasing inertia PSO (DIPSO), acceleration co-efficient and maximum velocity from self-organising hierarchical PSO and time varying acceleration co-efficient PSO(HPSO-TVAC)and its topology.

Adaptive Particle swarm optimization (APSO)-Zhan et al. [141] in 2009 suggested adaptive change in inertia weight as per location of particles, adaptive change in acceleration co-efficient as per evolution state and new update rules in selected evolution state to avoid stagnation.

Regrouping PSO (RegPSO)-Evers and Ghalia [142] in 2009 proposed automatically triggering swarm regrouping when premature convergence is detected.Particles areregrouped in range in each dimension proportionally to the degree of uncertaintyimplied by the maximum deviation of any particle from globally best position.

Discrete PSO with Embedded GA Operators – Premlatha and Natarajan [143]in 2009 suggested that during stagnation of the particles GA operator initiates reproduction and named this method DPSO with mutation-crossover.

Hybrid PSO -Li et al. [144] in 2010 suggested to combine three algorithm non-linear simplex method for fast convergence and integrated Tabu search into PSO for Tabu attribute local regions solutions.

Tabu List PSO (TL-PSO) -Nakano et al.[145]in 2010 proposed to store the history of Pbest particles in Tabu list which will be used only when particle is not performing well for updates and thus avoiding stagnation issues.

Cultural based PSO -Daneshyari and Yen[146] in 2010 proposed to find global minimum by using multiple evolution and multiple progresses simultaneously strategy as compared to conventional PSO.

Genetically improved PSO (GIPSO)- Abdel-Kader [147] in 2010 suggested this algorithm for k-means clustering and used to find initial kernel of solutions containing cluster centroids which are then used by the k-means for local search.

Perturbed PSO – Xinchao[148] in 2010 proposed to mutate global best particle during the run of the algorithm to avoid premature convergence.

Dynamic PSO based on improved Artificial Immune network (IAINPSO)-Tang et al.[149] in 2010 proposed this technique based on the variance of the population fitness a convergence factor is used for faster convergence with high precision and less number of iterations.

Feedback Learning PSO (FLPSO}-Tang et al. [150] in 2011 proposed for use of DIPSO for inertia weight, acceleration co-efficient uses some changes HPSO-TVAC with adaptive approach, for velocity updates adaptive strategy along with a mutation operator.

Self-Adaptive Learning PSO (SALPSO) -Wang et al. [151] in 2011 suggested to use four velocity updates rules from different variants of PSO and at every fixed number of iteration best update rule is selected and applied to every particles depending on situations. It is compared with its previous values then update rule is selected for next iteration.

Hybrid of PSO and Tabu Search (TS) – Zhang et al. [152] in 2011 proposed to solve non-linear integer program using hybrid of PSO and TS in which new heuristic rules were prepared to solve infeasible solutions.

Self-Learning PSO (SLPSO)- Changhe et al. [153] in 2012 suggested a different strategy of probability update and rule as that of SALPSO.

Orthogonal Learning PSO (OLPSO) – Zhan et al. [154] in 2011 proposed an orthogonal learning (OL) strategy which guide the particles to fly in better directions by constructing an efficient exemplar and can be applied to any topological structure for faster global convergence with high quality solutions.

Adaptive Fuzzy PSO- Juang et al. [155] in 2011 suggested to use fuzzy set theory to adjust PSO acceleration coefficient adaptively for better optimal and accurate values.

Enhancing PSO using generalised Opposition -based learning – Wang et al. [156] in 2011 proposed an enhanced PSO algorithm which employed generalised opposition based learning and cauchy mutation which helped the particles to escape local minima and results in faster convergence.

Genetic Simulated Annealing Ant Colony system with PSO- Chen and Chien [157] in 2011 presented this method to solve travelling sales man problem with percentage deviation in average solution is much better than existing techniques.

Multiple-Adaptive Methods for PSO (PSO-MAM) - Hu et al. [158] in 2012 put up another idea of using updating only the global particles by two techniques randomly either by mutation or gradient descent method at every iteration.

Self- learning PSO {SLPSO) -- Li et al. [159] in 2012 proposed a novel algorithm where each particle has a set of four strategy to handle different situations in the search space by an adaptive learning framework at individual level which in turn help the particle to choose optimal strategy according to its own fitness.

Chaotic Particle swarm fuzzy clustering – Liu et al.[160] in 2012 proposed a combination of new chaotic PSO and gradient method. The new chaotic PSO is used to search fuzzy clustering model using searching capabilities of fuzzy C-means for exploitation and gradient operator to accelerate the convergence.

Opposition based chaotic GA/PSO hybrid – Dong et al.[161] in 2012 proposed the method by combining the strengths of GA, PSO and chaotic dynamics. The velocity and position updates were fromPSO,selection, cross over, mutation from GA,and opposition based learning was done in chaotic hybrid algorithm for population initialisation.

Opposition based Natural Discrete PSO (ONDPSO)-Khan et al.[162] in 2012 introduced a new method in which particles were encoded by Natural Encoding scheme and position updating is done by new designed updating rule and opposition based learning is used in this process. The encoding scheme and position update rule used this technique allowed the individual term used corresponding to different attributes within the rule antecedent to be a disjunction of the values of those attributes.

Grey PSO – Leu and Yeh[163] in 2012 proposed two grey based parameter strategies, inertia weight and acceleration co-efficient. Each particle has its own inertia weight and acceleration co-efficient whose values are dependent on corresponding grey rational grade which is varying over iterations, those parameters are also varying. Even in the same iteration those parameters may be different for different particles. This strategy gives information about particle distribution in search space.

Mutation linear PSO (MLPSO} - Bonyadi et al.[164] in 2013 came with idea of multi-start PSO by combining the mutation operator with linear decreasing PSO (LPSO) to solve constraint problems.

Automatic Particle Injection PSO (APIPSO)- Elsayed et al.[165] in 2013 proposed to have good balance between exploration and exploitation by using standard PSO in starting and linear decreasing PSO latter phase. In order to prevent from trapping in local minima a mutation operator and automatic injection of new particles is done.

Essential PSO queen (EPSOq)- Ktari and Chabchoub [166] in 2013 suggested to use essential and strong feature of Tabu search to form improved discrete PSO.

PSO Using Dimension Selection method-Jin et al.[167] in 2013 proposed to use random dimension selection instead of stochastic coefficient is the another way of using randomness. Modified PSO is developed using dimension selection method shows better results and randomness is correct and important.

Recombination based hybridisation of PSO and ABC Algorithm -Kiran and Gndz [168]in 2013 proposed of combiningartificial bee colony (ABC) with PSO with five strategies and counters to update the solutions. Artificial agent perform the search process and counters are used to determine the rule that is selected by the bees.Depending upon the characteristic of problem, the artificial agent learn which in turn update the rule to find better solutions.

Novel Fuzzy PSO – Aminian and Teshnehlab [169] in 2013 proposed a novel method inertia weight, cognitive and social co-efficient are adjusted by fuzzy logic for each particle separately.

PSO –AIN Hybrid-Liu et al. [170] in 2013 suggested that the whole population is divided into two kinds of subpopulations as Elite and several normal sub populations. The best individual of normal sub population will be memorised into the Elite sub population during the evolution process.

Hybrid strategy in Continuous Ant Colony optimization (ACO) and PSO-Haung et al.[171] in2013 suggested to hybrid ACO and PSO for better searching capabilities of global minima without entrapping in local minima and introduced four types of hybridization. Sequence approach with the enlarged pheromone table was better than the other types due to diversified generation of new solutions.

Hybrid PSO-Simulated Annealing(SA) Approach- Jiang and Zou [172] in 2013 proposed an improved parameter method based on PSO by changing the fitness function in the traditional evolution process of support vector machines and then combined with SA global searching method which results in avoidance of local minima and better results.

Hybrid of PSO and Artificial Bee Colony(ABC)- El-Abd[173] in 2013 introduced component based one, and the PSO was augmented with the ABC component and was tested on the CEC13 test bed to improve the personal best of particles.

Blend of local and global variant of PSO in ABC –Sharma et al. [174] in 2013 suggested a technique called local-global variant ABC (LGABC) to balance between exploration and exploitation in the search space of ABC with good optimal results after testing on bench marks.

Combining Differential Evolution(DE) and PSO –Maione and Punzi[175] in 2013 introduced a hybrid two step approach in which DE determine the fractional integral and derivative actions satisfying the required time-domain specification and PSO determine rational approximations of the irrational fractional operators.

Improved Quantum behaved PSO Simplex (IQPSOS) - Davoodi et al. [176] in 2014 proposed to combine QPSO which gives direction to global region and Nelder-Mead simplex method for local search in the global region.

Centripetal Accelerated PSO (CAPSO) - Beheshadi et al.[177] in 2014 proposed to combine improved PSO with Newton laws of Motion which accelerates the rate of convergence and learning.

Improved Quantum behavedPSO – Li and Xaiao [178] in 2014 proposed an encoding approach based on Qubits described on Bloch sphere where each particle contained three groups of Bloch coordinates of qubits and all the three groups of coordinates were regarded as approximate solutions. Particles were updated using the rotation of qubits about an axis on the Bloch sphere.

Linear Constraint Minimum Variance (LCMV) assisted by PSO- Darzi et al. [179]in 2014 introduced by incorporating PSO, dynamic mutated AIS and Gravitational search algorithm into the LCMV method in order to improve the weights of LCMV.

Restarted Simulated Annealing PSO – Zheng et al.[180] in 2014 proposed an approach to decompose structuring elements of an arbitrary shape. And suggested a combination of restarted SA and PSO.

Adaptive hybrid of PSO and Differential Evolution(DE) –Yu et al.[181] in 2014 proposed a novel algorithm by balancing the parameter of PSO and DE and adaptive mutation is carried out on current population when clustering of population is near to local minima and thus diversity is also maintained.

Improved Accelerated PSO with DE-Wang et al. [182] in 2014 suggested a hybrid approach by using DE mutation operator to the accelerated PSO for solving optimization problems.

Co Swarm PSO with DE - Yadav and Deep [183] in 2014 proposed by hybridizing shrinking hyper sphere PSO(SHPSO) with DE approach by dividing the population in two sub swarms. First sub swarm use SHPSO and the other sub swarm uses the DE approach and are able to solve any real constrained optimization problems effectively.

Biogeography based PSO with Fuzzy Elitism – Guo et al. [184] in 2014 introduced to split the entire population into many sub groups where BBO was employed to search within group and PSO for global search.

Enhanced Comprehensive learning PSO – Yu and Zhang [185] in 2014 proposed two enhancements of comprehensive learning PSO ,first a perturbation term is added into each particle velocity update equation to achieve better exploitation .Normative knowledge about dimensional bounds of personnel best position is used to activate the perturbation based exploitation, second the particles learning probabilities are determined adaptively based on not on rankings of personal best fitness values but also the particles exploitation progress to facilitate convergence.

Levy Flight PSO –Husevin and Harun [186] in 2014 proposed to combine PSO with levy flight where limit value is defined for each particle and if the particle could not improve self-solution at the end of current iteration, then limit is increased .If this limit is exceeded by the particle, the particle is redistributed in search space by levy flight method to get rid of local minima.

Teaching and Peer-learning PSO(TPLPSO)–Lim and Isa[187] in 2014 proposed this technique consisting of teaching phase,the peer learning phase and the stagnation prevention strategy to improve PSO performance with high searching accuracy and convergence speed.The particle first enters in teaching phase and updates its velocity on its historical best and global best positions. If the particles fails to improve its fitness then it enters the peer-learning phase where an exemplar is selected as guidance particle and finally the last phase is used to alleviate the premature convergence.

Random Drift PSO (RDPSO) - Sun et al.[188] in 2015 got motivated by the model of free electron in metals placed in magnetic field have a drift velocity with thermal motion leading to a minimum potential energy.

Social Learning PSO (SL-PSO)-Cheng and Jin[189] in 2015 proposed that social learning technique inspired by learning methods, which requires no fine tuning of control parameters and is performed on sorted swarm.Unlike learning from the historical best positions, the particles learn from any better particles called demonstrators in the current swarm.

Heterogeneous Comprehensive learning PSO - Lynn and Suganthan[190] in 2015 proposed this technique where particles in a swarm will be allocated different search behaviours by randomly selecting velocity and position update rules from a behaviour pool thereby efficiently addressing the exploration-exploitation trade off.

PSO with Adaptive Inertia Weight using Bayesian Techniques -Zhang et al. [191] in 2015 proposed to apply Bayesian technique for better search ability in the exploitation of the past particle positions and for exploring Cauchy mutation for faster convergence rate with better solution.

Self- Regulating PSO - Tanweer et al. [192] pointed two learning strategies in 2015, the first one uses a self -regulating inertia weight which is employed by the best particle for better exploration and second uses the self-perception of the global search direction is employed by the rest of the particles for exploitation in the solution space.

Enhanced Leader PSO (ELPSO) -Jorde hi et al. [193] in 2015 proposed a five stage successive mutation novel strategy which is applied to the swarm leader at every iteration for mitigating convergence problem.

PSO based on Two Swarm Evolution -Wang et al. [194]in 2015 suggested a new strategy by adopting linear decreasing inertia weight to one swarm and random inertia weight to the other swarm. A random disturbance is added to the particle position at the stagnation point where it breaks the swarm into new escaped swarm from the local minima.

Feature selection Algorithm based on Bare bones PSO –Zhang [195]in 2015 proposed to find optimal feature subset to solving classification problems. A reinforced memory strategy was made to update the local leaders of particlesfor avoiding the degradation of outstanding genes in the particles and balance combination of exploration and exploitation.

Chaotic Simulating Annealing PSO –Geng et al. [196] in 2015 introduced to search more appropriate parameter combination where robust v-support vector regression is used to forecast port throughput.

ABC to generate diversity in PSO – Vitorino et al.[197] in 2015 proposed a method based on ABC to create diversity when all the particles of PSO converged to a single point by switching between two predefined behaviours by using fuzzy rules.

Novel Self Adaptive PSO -Pornsing et al. [198]in 2016 suggested a novel technique by diving the whole swarm into many sub-swarms thus allowing the particles to disperse the whole search space where the worst performer dies out and the best performer produces the offspring. Survival sub swarm adaptive PSO and Survival sub swarm adaptive PSO with velocity line bouncing approaches outperformed other algorithms.

Genetic Learning PSO (GLPSO)-Gong et al. [199] in 2016 develops a new framework by hybridizing PSO with another optimization technique learning called learning PSO ,which have two layers first for exemplar generation and the other for particles update by PSO algorithm. Genetic operators are used to generate exemplars from the particles learn and from the history search information of particles gives guidance to evolution of exemplars.

PSO with Inters warm Interactive Learning Strategy (IILPSO)-Cheng et al.[200] in 2016 put up a new concept of interactive learning behaviour in which particles are divided into two swarms .When there is no significant change in the fitness value then inter-swarm interactive learning strategy will start and check the best particle fitness values in both swarms. Softmax and Roulette method is used to classify them as learning swarm and learned swarm. Exploration with global search ability is increased by using velocity mutation operator and global best vibration strategy.

Distribution-Guided Bare-bones PSO (DBPSO)-Zeng and Shen [201] in 2016 suggested to overcome the problem of getting trapped in local minima by jumping and its probability is adaptively adjusted according to its current location.

Elite Promotion Quantum-Behaved PSO-Yang et al. [202] in 2016 proposed to use differential evolution operators to elite particles of the swarm for more local search and produce more global results efficiently for complex optimization problems.

Parallel Clustered PSO – Hoassin et al. [203] in 2016 proposed to combine PSO and K-means clustering which runs in parallel using MapReduce in the Hadoop platform and takes less time to compute.

Sophisticated PSO(Sop PSO) - Xia et al.[204] in 2017 suggested that this technique by using multilevel adaptation and purposeful detection .In Sop PSO a particle not only updates its learning model but also chooses its target that the particle learns from neighbours while adaptive strategy is applied in multi- level .Tabu search and local searching strategies are to jump the local minima.

Hierarchical Bare Bones PSO-Guo and Sato [205] in 2017 proposed that particles are separated into different groups and play different roles where group leader exchange information with the global particle and rest particles learn from the leaders.In the next iteration any particle can have better position from their group leaders.

New Social-Based Radius PSO -Munlin and Anantathanavit [206] in 2017 proposed to regroup the particles in a given radius of the circle and finds the agent particle which is best particle of the group for each local minima which helps to achieve global minima.

Primal-Dual Asynchronous PSO (pdPSO) –Gbenga et al.[207] in 2017 proposed a novel algorithmby combining Asynchronous PSO and Primal dual interior point algorithm. This algorithm combines the explorative ability of PSO with the explorative ability of Primal dual method thereby possessing a strong capacity to avoid premature convergence.

Movement PSO (MPSO)-Hudaib and Hwaitat[208] in 2018 proposed this algorithm that enhances the behaviour of PSO by using random movement function to search for more optimal points in the search space. This algorithm has good features like exploration, exploitation and local optima avoidance.

Centroid PSO – Anwar[209] in 2018 proposed a dubbed centroid PSO inspired by centre based sampling theorem which states that centre region of the search space contains points of high probability closer to optimum solution for data classification problems.

Scout particle swarm optimization (ScPSO) –Koyuncu and Ceylan[210] in 2018 suggested an efficient technique to hybrid PSO and Artificial Bee Colony (ABC),by adding a scout bee phase to standard PSO. The scout bee phase in ABC regenerates the useless particles that cannot improve their individual best positions and this process is operated through the parameter limit.

Fuzzy Controlled COBRA-fas (Co-operation of Biology Related Algorithm) – Akhmedova et al.[211] in 2019 developed based on six optimization methods namely PSO, Wolf Pack search(WPS),Firefly Algorithm(FFA), Cuckoo Search Algorithm(CSA), Bat Algorithm(BA)and Fish School Search(FSS) for solving real valued unconstrained optimization problems and is better in both exploration and exploitation than any other bio inspired algorithm.

Diversity-Guided Multi-Mutation PSO (DMPSO)-Tian et al.[212] in 2019 proposed Opposition based learning is used to get the high quality initial particles acceleration along with self-regulating inertia weights with three mutation strategies (Gaussian, Cauchy and Chaotic) to maintain diversity of the whole swarm. An Auxiliary velocity-position update mechanism is applied to the global best particle for convergence.

Triple Archives PSO –Xia etal. [213] in 2019 proposed this model in which particles in three archives are used. First the elite particles are recorded in one archive while other particles which show faster progress called profiteers are in another archive. Second, when breeding each dimension of a potential exemplar for a particle, we select a pair elite and profiteers from corresponding archive as two parents to generate the dimension value by genetic operators. Third, each particle carries out a specific learning model as per the fitness of potential exemplars. Finally the outstanding exemplar are saved in third achieve and reused by worse particles for better exploitation.

Fractional-order quantum PSO –Xu et.al[214] in 2019 proposed by using concepts of quantum mechanics and PSO with fractional calculus to achieve better global search ability. Grunwald-Letnikov is most frequently used fractional differential definition uses its discrete expression for its position updating of quantum behaved PSO.

Dual-Environmental Particle swarm optimizer – Zhang et.al [215] in 2019 proposed PSO variant that uses a weighted search centre based on top k-elite particles to guide the population. It averages there position rather than re-sampling fitness values of particles to achieve noise free environment.

2.4 **Constrained optimization problems (COPs)**

Many practical engineering optimization problems have constraints and require the solutions in that search space. PSO can easily solve such problems using certain strategy like static penalty, dynamic penalty, death penalty, MO approach, co-evolutionary, stochastic ranking, α-constrained,ϵ-constrained, hybrids, Del Valle's approach and Debs approach along with some modifications. Most commonly used are death penalty which is simple and parameter free whereas Debs approach is simple, derivative free and explore in infeasible regions also.

Trial and Error approach to constrained PSO-Hu and Eberhart [216] in 2002 proposed this method with two modifications, one the particles are initialized in feasible position and the other are only those solutions who satisfy the constraints are used for local and global positions.

PSO for Constrained problems –Parsopoulos and Vrahatis [217] in 2002 introduced dynamic penalty functions for the three variants of PSO and compared with other EA and found good results.

Death Penalty PSO – Coath and Halgamuge [218] in 2003 proposed this approach in which initialization is done in feasible solution search space and memories are updated, particles keep only feasible solution in memory .This approach is simple and parameter free.

Constraint handling Mechanism for PSO – Pulido and Coello [219] in 2004 presents a simple criterion based on closeness of a particle to the feasible solution as a leader with a turbulence operator for exploration in search space.

Constraint PSO – Zavala et al.[220] in 2005 proposed a novel PSO which uses a ring topology and a combination feasibility and domination in the selection of local best particle to maintain diversity and exploration within the swarm.

Dynamic Multi-swarm PSO for Constraints -Liang and Suganthan in 2006 [221] proposed that the swarm is divided periodically into sub swarms and particles are selected randomly. For better exploration the sub swarms search optimal solutions in constrained space.

Constrained PSO – Bochenek and Forys [222] in 2006 proposed controlled reflection technique for dealing with inequality constraints and particle trap strategy is used for equality constraints. If the particle is entrapped, then a penalty term is added to the objective function to force the captured particle and the constraints become active at the optimum.

Constrained optimization via PSO(COPSO)-Aguirre et al. [223] in 2007 proposed to use Lbest PSO to investigate the constraints and has external file called Tolerant to do analysis of particles. Lifetime of particles is developed by using the tolerant file with ring topology which maintains diversity.

Handling Constraints of PSO using small population size – Cabrera and Coello [224] in 2007 proposed to use leader selection scheme based on a distance of a solution to a feasible region along with a mutation operator to improve the exploration search using small population size of five.

PSO in constrained space – Flores and Mezura[225] in 2008 suggested a modified version of Debs approach, here computing the sum of constraint violation is done differently for equality and inequality constraints and comparing infeasible solutions.

New Vector PSO – Sun et al. [226] in 2009 proposed a vector PSO algorithm to solve constrained optimization problem in which one dimensional search methods were used to find a feasible position for each escaped particle.

Cooperation Comprehensive Learning PSO (CCLPSO)-Liang et al. [227] in 2010 presented a novel idea for solving constraints problem along with objective function where CLPSO was used either to satisfy constraints or optimise the objective and sequential quadratic programming was used for solution improvement during the run.

Improved vector PSO– Sun et al. [228] in 2011 proposed for search of feasible position in a local region consisting of dimensions of the parent and current position of the escaped particle using multi-dimension search algorithm solution.

Cultural Based constrained PSO – Daneshyari and Yen [229] proposed in 2012 combined the objective function and constrained violation in four sections of the belief space, specifically normative knowledge, spatial knowledge and temporal knowledge. With this information communication is good at personal level, swarm level and global level.

Extension of constrained PSO – Afsha r [230] in 2013 suggested that three constrained version of PSO based on identifying and excluding infeasible region of search space.

PSO based Hyper-Heuristic – Koulinas et al. [231] in 2014 proposed a PSO based on hyper-heuristic which worked as upper level algorithm and controlled many low level heuristic which operated to the solution space. The solutions are represented based on random keys and active schedules were made using the priorities of activities which were modified by low level heuristic.

Constrained PSO – Singh et al. [232] in 2014 proposed to detect a silent object in two phases using this technique. In first phase features like multi-scale contrast, centre-surround histogram and colour spatial distribution was obtained and in next phase constrained PSO determined an optimal weight vector to combine these features to obtain saliency map to distinguish salient object from the image background.

Multi-Target PSO- Cui et al.[233] in 2014 presented a novel approach multi-target (m PSO) to solve the parallel model of independent component analysis constrained by 5 parameter reference curve.

Hybrid PSO – Shou et al.[234] in 2015 suggested to solve the pre-emptive resource constrained project scheduling problem in which a maximum of one interruption per activity was allowed. Particle representation of four types were used and two schedule generation schemes were used to decode the particle representations. Peak cross over operator were used for particle updating for particle representations.

Constrained modified PSO (SASPSO 2011) – Tang et al. [235] in 2016 proposed the adaptive relaxation method which is integrated with the feasibility based rule to handle the constrained optimization problems of modified PSO (named as SASPSO 2011) so as to increase the diversity of solutions along with a parameter selection principle which guarantees the convergence.

Augmented Lagrange constrained PSO – Lu et al. [236] in 2017 proposed to optimize the objective function which combines the constrained PSO (CPSO) with the Augmented Lagrange multiplier (ALM) method. A new particle swarm is generated each time initially in order to avoid falling into a local best value and the best value can be easily found because the best value of the previous generation is saved and delivered to the next generation during the process.

Strongly Constrained space PSO – Ma et al. [237] in 2018 proposes a strongly constrained particle swarm optimization algorithm that brings water balance constraint into the search for feasible regions and this algorithm pays importance of the water constraint and rest of the constraints uses the constant penalty function method to avoid the problem of feasible regions.

2.5 Multi-Objective Particle Swarm Optimization

For solving real world multi-objective or multi criteria problems we optimize a solution and create feasible solutions across a parento optimal front but due to the unconstrained nature of PSO, the technique is modified to achieve a set of elite non dominated solutions.

Multi-Objective PSO {MOPSO) -Coello and Lechuga[238] in 2002 firstly proposed using the adaptive grid method to preserve the external file.

PSO Method in Multi-Objective Problems-Parsopoulos et al. [239] in 2002 suggested that three variants in weighted aggregation methods for multi-objective PSO. In linear aggregation function weights are fixed in objective function but in bang-bang aggregation function weights keep changing more than dynamic aggregation function in all the iterations.

Swarm Metaphor for Multi-objective Design optimization – Ray and Liew [240] in 2002 proposed to choose particles whose performance is better to be leaders and other particles select their leader randomly from the leader group where the leader with low followers have highest probability of being selected.

Vector Evaluated PSO -Omkar et al. [241] in 2002 suggested a multi swarms strategy depending on the number of objectives. Every swarm has its own objective function to optimize and the velocity update is done from the information from other swarms. This strategy gives a set of parento front solutions.

Dynamic Neighbourhood PSO {DNPSO)-Another strategy by Hu et.al [242] in 2003 was proposed to use Nbest instead of current Gbest and is the best particle in the specific neighbourhood. In this way the selection of neighbours for the current particle is one objective and the other selection of their best.

Particle Swarm with Extended Memory-Hu et al. [243] in 2003 suggested for Multi-objective optimization to combine extended memory to DNPSO as the number of Parento front solution are limited and some best solution are lost.

Divided Range PSO-Ji et.al [244] proposed a multi objective PSO in 2004 where the particles are divided in sub-swarms for one objective function then discrete PSO is run for each sub-swarms till stopping criteria is meet otherwise particles are again ordered for next objective function and the categorizing take place once more.

PSO with Passive Congregation (PSOPC) – He et al. [245] suggested interesting concept in 2004 about passive congregation (selfish behaviour in information sharing and forms passive group} where this passive group is added to PSO to increase its efficiency.

Parento Optimality and PSO - Baumgartner et al. [246] in 2004 proposed that parento based approach generates a set of solutions satisfying the main objective without effecting the performance of other objectives.

Handling Multiple Objectives with PSO - Coello et al. [247] in 2004 suggested to use the mutation strategy to solve multi-objectives using PSO.

Improved PSO based Multi-Objective optimization using Crowding,Mutation and E-dominance-Sierra and Coello[248] in 2005 proposed the crowding distance and E-dominance for diversity and divided the population to be divided into small sub populations with different mutation operator from escaping the local minima.

Variable Neighbourhood PSO- Liu et al. [249] proposed in 2006 that in the multi-objective problems trapping in local minima can be escaped by local search repeatedly from starting point to local optimum till better than current value.

Two level of Non-dominated solution approach - Abido [250] in 2007 proposed to find non-dominated solutions at local set and global set levels in multi objectives PSO.

Scalable Co-evolutionary Multi-Objective PSO -Zheng and Liu [251] in 2010 suggested to use decomposed decision variables and cooperative co-evolutionary sub swarms to solve multi-objective PSO.

Multi-Objective PSO based on Decomposition - Mart et al. [252] in 2011 proposed to use decomposition method to solve multi-objection problems.

Binary PSO hybrid with Artificial Immune network (AIN) – Ibrahim et al.[253] in 2011 proposed first the concept of topological monitor reach area and used binary PSO hybridized with AIN to solve multi-objective problem.

Local search based hybrid PSO for Multi-Objective optimization - Mousa et al. [254] in 2012 proposed that by combining GA and PSO the two character features of these algorithm can be used for Multi objective optimization. Firstly evolution of the particle is done to achieve non dominated solutions by initializing a set of random particles then local search done to explore more dominated solutions.

Trust Region (TR) algorithm based local search for Multi-Objective optimization - EI-Sawy et al.[255]in 2012 suggested to solve multi-objective optimization problems by using trust region method based on local search (LS)technique, where a multi-objective optimization problems is converted into single objective optimization problem by using reference point method. For each reference point the TR method is used to obtain a point on a Parento frontier and LS method is used to find more points on parento-front.

Bare Bones Multi-Objective PSO- Zhang et al. [256]in 2012 proposed an algorithm that have three features namely particle updating strategy which do not require tuning of control parameters, mutation operator for search capability and an strategy based on particle diversity to update global particle.

Fuzzy PSO for Multi-objective – Khan and Engelbrecht [257] in 2012 proposed to incorporate fuzzy logic in PSO to solve multi-objective problem where unified And-OR operator were used to aggregate the objective.

Multi-objective PSO (MOPSO) with K- Means – Qiu et al. [258] in 2013 proposed this technique with new Gbest selection strategy and used K-means algorithm. A Gbest particle is selected by using proportional distribution approach and a mutation operator is used to enhance exploration.

Co-evolutionary Multi-Swarm PSO for Multi-objective –Zhan et al.[259] in 2013 proposed this technique based on multiple population with multiple objectives (MPMO) by using an external shared archive for different to exchange search information and by using two designs to enhance the performance.

Multi-objective PSO (MOPSO) and Fuzzy Ant Colony Optimization (FACO)-Elloumi et al.[260] in 2014 introduced combination ofbest particle of fuzzy Ant Colony and integrate it as local best particle of PSO to formulate a new approach as hybrid MOPSO with FACO for solving multi-objective problems.

Multi objective hybrid Quantum PSO(QPSO)- Chen et al. [261]in 2014 suggested to use elitist hybrid QPSO with mutation where elitist mechanism with crowding distance sorting was used to improve the diversity and quantity of optimal solutions.

Multi objective planning using PSO – Ganguly [262] in 2014 proposed a PSO based multi objective planning algorithm with minimizing the three objectives simultaneously to obtain a set of non- dominated solutions.

Multi objective Reliability Redundancy problems using Extended Bare Bones PSO(BBPSO)– Zhang et al.[263] in 2014 proposed a two stage algorithm in which Bare bones PSO multi objective PSO is developed and applied to the first stage and find parento optimal set .This algorithm is the combination of Bare bones PSO and sensitivity based clustering for solving multi objective reliability redundancy allocation problems.

Improved Multi objective PSO with Preference Strategy – Cheng et al. [264] in 2015 proposedthis strategy by using preference factors were used for certain attributes in constraint space. The performance of this technique was strengthen by using dynamic selection of global best, circular non dominated selection of particles and a mutation operator.

Co-operation of Biology related Algorithms for Constrained Multi-Objective Problems (COBRA-m) –Akhmedova and Semenkin [265]in 2015 proposed the cooperation work of five algorithm namely PSO, Wolf Pack search(WPS),Firefly Algorithm(FFA), Cuckoo Search Algorithm(CSA) and Bat Algorithm(BA) with the use of Pareto optimality theory for the multi-objective problems and works effectively.

Multi swarm Comprehensive learning PSO for solving multi objective problems (CLPSO)- Xiang and Zhang [266]in 2017 proposed that each swarm focus on separate objective using CLPSO without learning from other swarm ,mutation is applied to elitists only and modified differential evolution strategy is applied to some extreme least crowded elitists.

External Archive-Guided Multi objective PSO – Zhu et al. [267] in 2017 proposed a novel algorithm where multi objective problems are converted into sub problems using decomposition method and then each particle is assigned accordingly to optimize sub problem. This technique is designed for better exploration and the external archive is used from immune-based evolution strategy for speedup convergence.

Multi-objective PSO using Ring topology - Yue et al.[268] in 2018 proposed this technique to solve multi-modal multi-objective problems using ring topology and special crowding distance, where ring topology helps in finding much more parental-optimal solutions and special crowding distance considers the crowding distance both in decision and objective space to maintain multiple parento solutions.

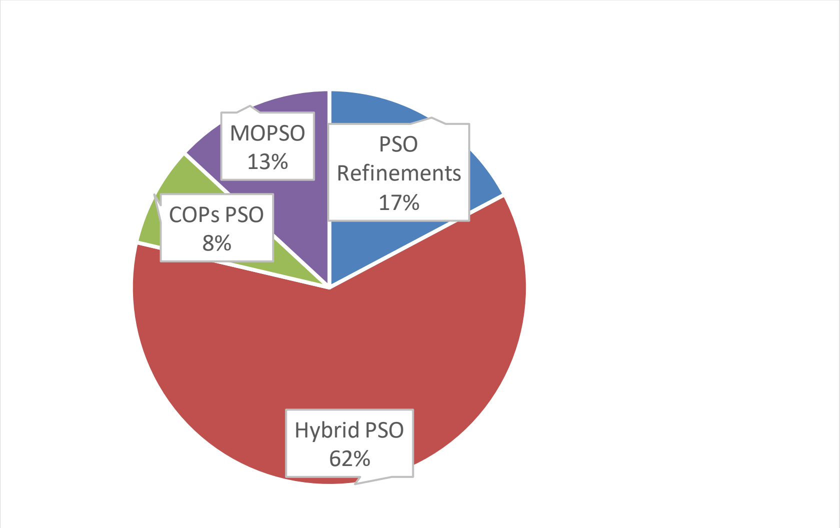
Adaptive Gradient Multi objective PSO(AGMOPSO) – Han et al.[269] in2018 suggested state of art technique in which stocktickerMOG method will update the archive for better convergence, local exploitation and self- adaptive flight parameters mechanism, according to diversity information of the particles will maintain balance convergence and diversity.

Self-organising RBF Neural network usingAdaptive Gradient Multi objective PSO (AGMOPSO) – Han et al.[270] in 2019 proposed to optimize both the structure and parameters of RBF Neural networks by developing AGMOPSO then the AGMOPSO based self- organising RBF Neural network can optimize the parameters (centres, width and weights) as well as network size.

Surrogate assisted PSO with Parento Active learning - Zhiming et al. [271] in 2019 proposedto save computational costof multi-objective optimization problems. PSO is regarded as a sampler to generate candidate solutions and the performance is improved by preselecting results with the improved ε-PAL. A hybrid mutation sampling method based on simulated evolution is used to improve the performance of sampler.

Multitasking Multi-swarm optimization (MTMSO) – Song et al. [272] in 2019 proposed to divide randomly the whole swarm into multiple task swarms for particular task and each swarm is further divided into sub swarms. Each task group works on dynamic multi-swarm optimization algorithm and probabilistic crossover of personnel best of particles from multiple task group is done for cross task knowledge. Task group and each group sub swarm are reformed periodically to maintain search diversity.

Fig2.1 Shows the distribution of all research papers published from 1995 to end of2019 using Refinements in PSO, Hybrid PSO, COPs PSO and Multi-objective PSO



2.6 **Parallel Implementation**

The main problem in PSO implementation is its runtime when dealing with large optimization problems or in higher dimensions and parallel implementation is best suited to solve this problem. In parallel computing computations are carried out simultaneous. The multiple processing units of a single computer do the independent calculations in the inherently parallel structure of PSO by using the parallel sub-swarms to the different processors with the exchange of information between them. On the other hand multiple computers uses Grids, clouds and clusters [273] to perform the same task.

2.6.1Multi core

Optimization problem can be solved speedily by splitting into parts and each part is computed simultaneously by a single or multiple machines by using parallel computing strategy with multi core or multiprocessor. There are various parallelization techniques like Hadoop MapReduce[274],Rparallel package [275] ,MATLAB Parallel Toolbox [276],OpenMP with C++ [277],Parallel computingmodule in Python [278],Julia- Parallel for and MapReduce [279],MPI [280]to access the multiple cores with one or more CPUs.

Parallel Implementation - Gies and Rahmat-Samii in 2003 showed [281] a performance gain of eight times using a system with 10 nodes for a parallel implementation over s serial one.

Parallel Global Optimization with PSO–Schutte et al.[282] in 2004 proposed parallel implementation on two types of problems .Firstly on large scale analytical problems with inexpensive function evaluations and secondly medium scale problems on bio-mechanical system identification with computationally heavy function evaluations. It uses a synchronous scheme based on a master-slave approach.

Parallel PSO accelerated by Asynchronous Evaluations–Venter et al. [283] in 2005 proposed to use this parallel scheme based on Message Passing Interface to provide master-slave implementation. The asynchronous algorithm updates all the design point information as soon as point is available and directly starts the next iteration without waiting for all the points to be assessed.

Parallel PSO with Communication strategies – Chang et al. [284] in 2005 suggested three strategies used according to the independence of the data. The first strategy is designed for solution parameters that are independent or loosely correlated whereas second strategy applied to strongly correlated parameters and third strategy is applied to properties of unknown parameters.

Relative velocity updating in Parallel PSO – Chusanapiputi et al. [285] in 2005 introduced synchronous implementation with relative velocity updating based on parallel relative PSO. In this technique after exploring nearby slave send the best position and velocity updating to the master and master selects best velocity and the next move is decided accordingly.

Parallel Asynchronous PSO (PAPSO) – Koh et al. [286] in 2006 proposed a parallel asynchronous PSO that exhibits good parallel performance for large number of processors as well as good optimization performance. PAPSO gives 3.5 times faster than parallel synchronous PSO results in heterogeneous computing condition

Parallel PSO using MapReduce (MRPSO) – McNabb et al.[287] in 2007 proposed novel technique as MapReduce parallel programming model in Hadoop[288] where in mapping phase particle is mapped and obtains updated velocity ,position, pbest and in reduced phase gbest is calculated by collecting all the information.

Two Phase Parallel PSO(TPPPSO)– Liu et al.[289] in 2007 suggested individual orientation factor function uses exploration and overall orientation factor function uses expanded search area in second phase.

Parallel Multi-Population PSO using OpenMP – Wang et al.[290] in 2008 suggested asynchronous version using OpenMP where the particles were ranked as per performance in fitness function then sub-populations were created and the best position in population and sub population is considered for updating position and velocity.

Multiprocessor modelling of Parallel PSO – Waintraub et al. [291] in 2009 proposed this technique using master-slave approach and developed many PPSO using the enhanced network topologies implemented by communication strategy in multiprocessor architectures.

Parallel PSO (PPSO) – Jeong et al.[292] in 2009 expressed this technique for PC cluster that exchanges the information with sub swarms one by one using coarse grain topology in order to maintain diversity and avoid premature convergence.

Parallelization of PSO using Message Passing Interface (MPIs) – Singhal et al. [293]in 2009 implemented asynchronous PSO using MPI commands on the multiple processes and this algorithm split the particles in such a finest way for every number of processors and the processor with good result becomes the root processor at the end of each cycle.

Synchronous Parallelization of PSO with digital pheromones – Kalivarapu et al. [294] in 2009 suggested to use multiple swarms in n-dimensions search space in parallel computing technique with new PSO variant with digital pheromones increased the efficiency with lower time period in higher dimensions.

Agent based Parallel PSO –Lorion et al.[295] in 2009 introduced a coordination agent between swarms and other coordination swarm agents for distributing and managing a particle swarm on multiple interconnected processors.

Parallel Scalable hardware implementation of Asynchronous discrete PSO – Farmahani-Farmahani et al. [296] in 2010 proposed a hardware pipelined PSO for performing computational operation of algorithm with the notion of system on a programmable chip(master slave multi-processor) for discrete optimization problems. Sub-particle method is used to bring the benefit of full scalability and asynchronous PSO gives better efficiency for large and complex problems.

Communication latency tolerant Parallel PSO – Li and Wada [297] in 2011 suggested globally synchronised parallel PSO with delayed exchange parallelization which improves PSO performance on distributed environment by hiding communication latency. This method delays the best function fitness exchange to one loop later.

Parallel PSO implemented by multiple threads – Tu and Liang[298] in 2011 proposed that communication among the sub groups is implemented by parallel computation models based on broadcast, star, migration and diffusion network topologies .Due to the expense and difficulty of true parallel computation multiple threads are used for simultaneous particle interaction.

Parallel PSO Clustering based on MapReduce – Alijarah and Ludwig[299] in 2012 proposed this algorithm for optimal clustering in three sub-modules. Sub-module first updates the particle swarm centroid in MapReduce and in second sub module fitness evaluations are for new centroid and finally updating in personal best and global best centroids.

Particle Co-operative Micro-PSO – Parsopoulus [300] in 2012 proposed an algorithm based on decomposition of search space into smaller search spaces of smaller dimension using two types of computer systems as academic cluster and desk top multi core system for evaluating this approach.

Twin PSO – Yu [301] in 2014 proposed the incorporation of local search heuristic into PSO algorithm and this new hybrid version is called Twin PSO and was applied to flow shop with multiprocessors scheduling problem.

Parallel Multi-swarm algorithm based on Comprehensive learning PSO – Gulcu and Kodaz [302] in 2015 proposed multi swarm which work co-operatively and the local best get exchanged in every migration process to maintain diversity of solutions.

Parallel PSO using Message passing Interface – Zhang et al.[303] in 2015 proposed to combine Global model PSO, Local model PSO, Bare bones PSO and Compressive Learning PSO these four versions by using the MPI to achieve high quality solutions as compared to serial versions of these four PSO variants.

Parallel PSO-Back Propagation Neural network based on MapReduce – Cao et al. [304]in 2016 proposed a parallel design realization method for PSO optimized BP neural network based on map-reduce on the Hadoop platform and PSO algorithm is used to optimize the inertia weights and thresholds for the back propagation neural network.

Parallel Evolution of quantum behaved PSO-Tian et al.[305] in 2016 introduced the splitting of high dimension problem into sub problems and get optimized individually with the intermittent communication resulting in high quality solutions.

Fine Grain Parallel PSO (FGPPSO) – Nedjahet al.[306]in 2017 proposed this technique for multi core and many core architectures along with serial implementation and the termination criterion was taken as leaning upon the accessibility of solution.

Parallel PSO for Multi core Environment – Abdullah et al.[307] in 2018 proposed Parallel PSO on multi core processing kernel to decrease the determination and transfer information easily among particles of shared area and exchange information by random replacement strategy. This shared PSO technique is more effective than serial PSO can avoid the reduction on test accuracy when applied on single core environment.

Adaptive Parallel PSO – Lai and Zhou [308] in 2018 suggested parallel PSO based on Osmosis and is capable to obtain three parameters as migration interval, migration direction and migration rate which is helpful in determine the number of particles migrated from one sub population to another sub population.

2.6.2 **GPU Computing**

In November 2006,NIVIDA a computer games company introduced the CUDA model which enables the programmers to write their own code using C programming language with NIVIDA extensions[309,310] and capable to compute big data parallel computations .GPU has thousands of cores installed and the power of multiple CPUs in single processor. CUDA, OpenACC [311], Intel Xeon Phi bootable host processors, TPU [312], FPGA [313] are the GPU based parallelization approaches.

Fine- Grained Parallel PSO based on GPU-Acceleration – Li et al.[314] in 2007 first proposed that the particles were mapped into textures on a graphics card and calculated in parallel without Compute unified deice architecture (CUDA) support and then implemented on CUDA.

GPU-based Parallel PSO – Zhou and Tan [315] in 2009 proposed a novel parallel approach to run PSO on GPU based on software platform of CUDA from NVIDIA [309]. The running speed of GPU is more than 11times faster than CPU and running time is also reduced. High dimension problems and large swarm population application are its special advantage in real optimizing problems.

Swarms Flight: Accelerating the particles using C-CUDA - Veronese and Krohling [316] in 2009 suggested the implementation of PSO algorithm in C-CUDA, which showed high computing capabilities with lesser time on well-known bench marks as compared to C and MATLAB.

Parallel PSO based Particle filtering - Rymut and Kwolek [317] in 2010 proposed in their work that CUDA -capable GPU can accelerate object tracking algorithm performs using adaptive appearance models with a speed factor of 40 over CPU. The object tracking is done by PSO algorithm.

Evaluation of Parallel PSO within CUDA – Mussi et al. [318] in 2011 showed the performance evaluation of two variants of parallel algorithms with the sequential implementation of PSO over bench mark functions.

Collaborative multi-swarm PSO for task-making using GPU – Solomon et al. [319]in 2011 proposed collaborative multi-swarm PSO procedure on GPU using multi-swarms rather than one, when applied to real world problems in a heterogeneous distributed computing environment.

Paralleling Euclidian PSO (pEPSO) in CUDA – Zhu et al. [320] in 2011 proposed this algorithm to use fine grain data paralleling to evaluate the fitness function with GPU for fast and better convergence.

Accelerating Parallel PSO via GPU(GPSO) – Hung and Wang [321] in 2012 suggested this algorithm by using thread pool model and implement GPSO on a GPU. The GPU architecture fitted with PSO framework reduces the computational timing and high efficiency with better optimal results.

GPU based Parallel Co-operative PSO using C-CUDA – Kumar et al. [322] in 2013 suggested that computational time is reduced huge computations got benefit from the GPU with Compute unified deice architecture (CUDA).They performed detailed study of Parallel implementation of co-operative PSO and a comparative study on CPSO in implemented in C and C-CUDA.

SIMT GPU Based PSO Approach- Awwad et al. [323] in 2013 suggested to compute CUDA GPU solution to solve the topology problem and achieved a performance speed up factor of 392 over a CPU implementation for the large scale optimization problems.

Co-operative Evolutionary multi-swarm optimizer based on CUDA – Souza et al. [324] in 2013 proposed this technique based on CUDA to solve optimization problems. This method use the concept of master-slave swarm with the mechanism of data sharing for the acceleration of convergence.

Parallel GPU based implementation of High Dimension PSO –Calazan et al. [325] in 2013 proposed that each particle gets implemented as a block of threads and each dimension is mapped onto a distinct thread for faster rate of convergence.

Parallel PSO – Chen et al. [326] in 2014 proposed an efficient PSO based algorithm to find optimal uniform designs with respect to the CCD criterion. Parallel computation technique based on the state of art graph processing unit (GPU) is employed to accelerate the computations.

FJSP based on CUDA Parallel Cellular PSO – Shenghui and Shuli [327] in 2014 proposed this algorithm by putting large number of GPU threads to each particle and employs CA logic where each particle is considered as CA model. Calculation space is provided to each particle on respective thread and number of threads in GPU equals to number of particles.

PSO Efficient implementation on GPU using Low Latency Memory – Silva and Filho [328] in 2015 developed this technique using the shared memory available in the GPU of CUDA platforms. Each dimension of each particle is mapped as thread and are executed in parallel in GPU block which has maximum number of allowed parallel threads and use multiple sub-swarms. Each sub swarms Is executed in a GPU block aiming at maximizing data alignments and avoiding instructions bifurcations with two communication strategies and two topology.

Parallel PSO based on CUDA in the AWS Cloud – Li et al. [329] in 2015 proposed this algorithm to run all the processes in parallel for updating current position, velocity,best fitness and global best fitness .This algorithm is speeds up 80 times as compared to PSO algorithm on CPU.

Parallel PSO approaches on GPU for constraint – Dali and Bouamama[330] in 2015 suggested two approaches to solve constraint satisfaction problems (CSPs), first one by parallel GPU-PSO for max-CSPs and the other by GPU distributed PSO for reducing the calculation time to explore the search space efficiently.

CUDA implementation on Standard PSO – Hussain et al.[331] in 2016 proposed the use of coalescing memory access ,video RAM memory is used which is more efficient in simultaneous memory access by threads in a wrap for a standard PSO on GPU based on CUDA and found 46 times faster than CPU serial implementation.

Parallel PSO on GPU with application to trajectory optimization – Wu et al.[332] in 2016 presented the full implementation of PSO in parallel through GPU on CUDA platform and studied the effect of number of particles, dimensions, size of thread block in the GPU and there interaction on computational time.

GPU-Based Parallel PSO for Graph Drawing – Que et al.[333] in 2017 developed two procedures, one serial and the other parallel for undirected graph drawing. The serial PSO procedures were executed on CPU with lesser time on small graphs whereas parallel PSO is executed on GPU with lesser time on large graphs.

Adaptive PSO with heterogeneous multi-core parallelism and GPU acceleration – Wachowiak et al. [334] in 2017 adapted this method for parallelization on heterogeneous parallel hardware that contain multi-core technologies speeded by GPU and Intel-Xeon Phi co-processors expedited with vectorization. Task-parallel elements are carried out with multi-core parallelism and while data-parallel components get executed via co-processing by GPU.

PSO based Parallel Road Network method on GPU- Wan et al. [335]in 2018 developed this technique based on the features of two stages as computation and matching relationship identification using data partition and task partition strategies are used, to fully use GPU threads. This method can easily handle massive data with good efficiency.

MS2 PSO – Tangherloni et al.[336] in 2018 proposed efficient parallel and distributed implementation of a Parallel Estimation (PA) based on PSO for the estimation of reaction constants of biochemical systems.MS2 PSO is based on Master-Slave distributed computing in which master process offloads the time consuming calculations. Each Slave exploits cup SODA which allows to run in parallel on the cores of GPU to calculate the fitness values for optimization.

Performance evaluation of PSO,GA based on GPU – Kawano et al.[337] in 2018 proposed to execute PSO,GA using the original code on the processor against the modified algorithm where the certain process of the algorithm are integrated on video card to compare the execution time. Also graphical interface was made for both algorithms to facilitate the process of handling the parameters.

Integrated motor optimization and Route planning for EV using GPU –Roberge et al. [338] in 2019 proposed PSO and the Bellman-Ford (BF) routing for minimizing energy consumption for EV and are implemented in CUDA..PSO is used to calculate magnetic flux settings for an Induction motor for various operating points and losses are also calculated prior to trip.BF is used to calculate optimized routes and Parento front of routes are prepared.

2.6.3 Cloud computing using PSO

Cloud computing technologies provide method to deal with massive data,delivering a flexible, pay-as-you-go for [339,340] and are needed for high performance complex application. Cloud computing helps user applications dynamically provision as many compute resources at specified locations (currently US east1 a-d for Amazon [341]) as and required. Applications can choose the storage locations to host there data (Amazon S3) [342] at global locations. These services are called Infrastructure as a service (IaaS), Platform as service (PaaS) and Software as a service (SaaS).Cloud computing has four layered architecture as data centre layer, platform layer, infrastructure layer and application layer. There are four types of cloud such as Public cloud, Private cloud, hybrid cloud and community cloud. It is found that there has been frequent use of PSO to solve all problems of cloud computing liketask scheduling, Energy optimization, Load balancing and workflow scheduling problems to get efficient solutions over the different virtual machines on the cloud environment.

PSO based heuristic for Scheduling workflow in cloud computing environments – Pandey et al.[343] in2010 proposed to minimize the total cost of execution of application of workflows on cloud computing environments by varying communication cost between resources and the execution cost of compute resources. The results showed three times saving in cost as compared to Best Resource Selection (BRS) heuristic.

Discrete PSO for cloud workflow scheduling – Wu et al. [344] in 2010 proposed to schedule applications among cloud services by combining the data transmission cost and computational cost for optimal cost to user. Moreover this algorithm is not better for larger search space.

Set-based discrete PSO for cloud workflow scheduling with user defined Qos constraints – Chen and Zhang [345] in 2012 proposed to optimize the user Qos such as make span, user cost and reliability separately.

Sort based PSO in Cloud computing – Guo et al. [346]in 2012 adopted a small position value rule by sorting all the dimensions in position according to the real value and giving each dimension an integer value rank number and then map this value to the cloud resource index . A single objective was formed by combining the data transmitting time and the user cost.

PSO for energy aware virtual machine placement optimization – Wang et al. [347] in 2013 proposed for lowering the energy consumption of a virtualized data centres by means of virtual machine placement optimization while keeping in view the necessary requirements of cloud services .An improved version of PSO is used by redefining the parameters and operators, then adopting an energy aware fitness strategy and coding scheme.

Round based PSO in cloud computing – Rodriguez and Buyya [348] in 2014 proposed to round the real number to integer number to indicate the resource index that the workflow was scheduled on but it does not reflect the features of resources.

Energy efficient resource allocation of Virtual machine– Xiong and Xu [349] in 2014 suggested this algorithm by using Energy efficient resource allocation model and PSO method in cloud data centre to reduce the energy consumption. The fitness function of PSO is defined as the total Euclidean distance to determine the optimal point between resource utilization and energy consumption.

Task based system load balancing using PSO (TBSLB-PSO) – Ramezani et al. [350] in 2014 suggested for system load balancing by only transferring extra tasks from an overloaded virtual machine (VM) instead of migrating the entire overload VM. Then PSO is applied to migrate these extra tasks to the new host VMs for reducing the downtime, cost and amount of memory involved in this process.

Renumber strategy enhanced PSO in cloud computing – Li et al. [351] in 2015 suggested a number strategy to use the metric of the price per unit time to record the resources and thus making the learning among the particles more efficient.

Cloudlet scheduling with PSO – Al-Olimat et al.[352] in 2015 proposed a hybrid of PSO and Simulated Annealing is implemented inside the CloudSim is used to minimize the makespan and maximize the resource utilization.

Dynamic Power saving Resource allocation using PSO (DPRA) – Chou et al. [353] in 2018 suggested this mechanism based on PSO which consider the energy consideration of physical machine (PM)and virtual machine (VM) and also take care of energy efficiency of air-conditioners ,total electricity bill, VM migration, and number of shut downs of VMs.

Quantum PSO (QPSO) Based Load Balancing – Sivakumaret. al [354]in 2019 proposed to decrease the traffic surrounded by the incoming requests to the server which is protected by firewalls ,sends to the load balancer that acts as reverse substitute and distributes network transversely to servers .This algorithm consider data dependences in cloud environment and data intensive workflow features.

2.7 **Hybrid PSO using Parallel Implementation**

Parallel hybrid Moving boundary PSO(hmPSO)– Zhang et al. [355] in 2009 proposed that this algorithm consists of three components global bpso, local bpso and direct local search by Nelder-Mead method. The hardware for parallel implementation is a LINUX cluster consisting of 96 dual processor dual-core operation.This hybrid model improves the efficiency and avoid premature convergence to local minima's.

Parallel PSO with Genetic Migration – Jin and Lu [356] in 2012 proposed coarse grained parallel PSO on GPU and implemented genetic strategy for communication using selection, crossover and mutation operators on the particles and after competition of migration among swarms, new swarms run on PSO.

Comparison Parallel GA and PSO – Roberge et al. [357] in 2013 proposed the hybrid GA and PSO to reduce the execution time for the solutions by using single programming, multiple data parallel programming. By using parallel implementation on multi-core CPUs, a real time path planning for UAV is possible with a quasi-linear speedups 0f 7.3 to 8 cores with low execution time.

Hybrid approach based on neighbourhood search and PSO for parallel machine (VNPSO) – Chen et al.[358] in 2013 proposed this algorithm to multi stage problem and formulated as a mixed integer linear programming model. The algorithm addresses both sequence independent as well as sequence dependent setup time.

Parallel Co-operative Co-evolution based PSO (PCCPSO) – Yuan et al. [359] in 2015 proposed this technique to solve conditional nonlinear optimal perturbation (CNOP) problem. A hybrid using advancement in PSO with Tabu search algorithm was used and then parallelizing was performed.

Spark based Parallel C0-operative Co-evolution PSO – Cao et al.[360] in 2016 introduced a hybrid algorithm by combing probability distribution functions (Gaussian, Cauchy and Lcvy distribution functions) with the global and local version of PSO and implemented on spark platform for solving high dimension problems in parallel.

Parallel Quantum-behaved PSO with neighbourhood search – Long et al.[361] in 2016 suggested to use global search and local search neighbourhood strategy in quantum behaved

PSO and employ parallel technique for reducing runtime and increased the diversity of the population.

Hybrid of Multi Swarm PSO and GA – Franz and Thulasiraman [362] in 2016 proposed this algorithm by parallelizing the hybrid algorithm on an accelerating processing unit (APU) which is a hybrid multi core computer to improve performance and close coupling between GPU and CPU.

Hybrid Iterative Truncated singular value decomposition (TSVD) and Parallel PSO – Ge et al. [363] in 2016 suggested the new inversion algorithm can achieve favourable results for signals with signals to noise ratio larger than 10 by inversing the relaxation time (T1) and transversal time (T2) spectrum in a low field nuclear magnetic resonance to obtain optimal truncated position with high computational speed.

Multi-Core Parallel PSO (PPSO) -Peng et al.[364] in 2017 proposed three multi-core parallel PSO algorithms (PPSO_ring, PPSO_star, PPSO_share) based on Fork/Join framework and concurrency in Java for exchange of information among the threads (sub-swarms).Fork/Join framework assigns threads to different CPU cores ,whereas synchronization and communication mechanisms are employed exchanging information among the threads.

Hybrid GA-PSO in Cloud Computing – Manasrah and Ali[365] in 2018 proposed to allocate the task to resources efficiently and to insure the fair distribution of the workload among the available virtual machines so as to reduce the make span and the processing cost of the workflow applications with minimum time in the cloud computing environments.

Hybrid GA-PSO in Cloud Computing – Senthil et al. [366] in 2019 proposed to combine GA and PSO to minimize the execution time for task scheduling. Initially GA will randomly generate the population and encoding the chromosomes is done with mapping task and matched resources. Fitness is calculated and elite are divided into two halves.GA is applied to best elite first half. Anew population is resulted after applying crossover and mutation. With PSO, pbest and gbest are evaluated with every iteration for particle position and velocity. Results are combined both of GA and PSO. Finally results are sorted based on fitness values and global best is the optimal solution.

2.8 **Multi-Objective Particle Swarm optimization using Parallel Implementation**

Parallel Vector evaluated PSO (VEPSO) – Vlachogiannis and Lee [367]in 2005 implemented this algorithm which contains equal number of objective functions and number of swarms with the same number of PCs working in parallel for solving the multi-objective optimization problems in short computing time with precise results.

Parallel PSO for multi-objective problems – Fan and Chan [368] in 2009 developed the idea based upon the concept of parento-dominance as well as parallel computing in which after standard PSO run, each swarm shares a fixed number of crowed member after the migration period .Fixed number of non-dominated solutions were obtained by external archive which keeps on updating after each cycle.

GPU based Parallel multi-objective PSO- Zhou and Tan[369] in 2011 firstly proposed this approach for optimizing parallel multi-objective problems via PSO using the GPU is more efficient in running time and speed range from 3.74 to 7.92 times as compared to CPU sequential platform.

Parallel implementation of MOPSO on GPU using OpenCL and CUDA – Arun et al. [370] in 2011 implemented this technique on the popular GPU frameworks which results in 90% improvement in performance as compared to sequential implementation.

Multi-objective parallel PSO – Soares et al. [371] in 2013 proposed to solve the dual objective of Vehicle to grid scheduling by applying parallel computing parento weights to multi-objective parallel PSO.

Multi objective Parallel PSO-SA (P-PSOSA)- Khoshahval et al.[372]in 2014 proposed two different fitness function were defined considering multiplication factor maximizing and power peaking factor minimizing objectives simultaneously ,thus achieving near global core pattern.

Parallel multi-objective PSO based on Decomposition – Li et al. [373] in 2015 proposed to use both MPI and OPENMP to implement the algorithm with a hybrid of distributed and shared memory programming models.

Weighted sum approach using Parallel PSO–Borges et al. [374] in 2016 implemented parallel PSO to the non- linear multi-objective combinational resources scheduling problem of distributed energy in which single objective function is formed by the weighted sum of two objectives.

Workload Distributor with a Resource Allocator (WDRA) – Alsubhai and Gaudiot[375] in 2017 proposed to combine workload distribution, core scaling, and thread allocation into a multi-objective optimization problem using PSO in order to reduce the execution time, energy consumption, under peak power and peak CPU temperature constraints.

Scalable Parallel co-operative Co-evolutionary PSO - Atashpender et al. [376] in 2018 firstly suggested this variant of speed constrained multi-objective PSO and scalability analysis in the terms of number of variables and parallelization. This method gives high computation speed ups and higher convergence speed with quality solutions.

Parallel Multi-objective PSO for large swarm and high dimensions (MOPSO) – Hussian and Fujimoto[377]in 2018 introduced parallel implementation MOPSO on a GPU based CUDA architecture using coalescing memory access ,pseudo random number generator, thrust library, atomic function, parallel archiving and so on. This implementation uses master slave model provides up to 182 times speedup as compared to CPU MOPSO.

Bi-Objective PSO – Varshney and Singh[378] in 2018 proposed this technique in which two swarms are used one for each objective such that information of one swarm is used to update the velocity of the other and both swarm co-operate each other to get better optimal solutions and is better in terms of reliability and execution time in cloud computing environment.

Parallelized Multi-objective Cultural algorithm PSO (CAPSO) – Stanley et al. [379] in 2019 proposed a parallelized hybrid optimization system by combining elements from cultural algorithm(CA), PSO and Vector Evaluated Genetic algorithm(VEGA).This algorithm works by dividing the search space within multiple swarms joined by sharing of CA knowledge among themselves.

Parallel Multi-swarm PSO strategies for Multi-objective – Campos Jr. et al. [380] in 2019 proposed two strategies, firstly based on parento dominance and the second on decomposition. Multi-swarms execute on independent processors and communicate on a fully connected network. Parallelization has more impact on convergence and diversity on multi-objectives.

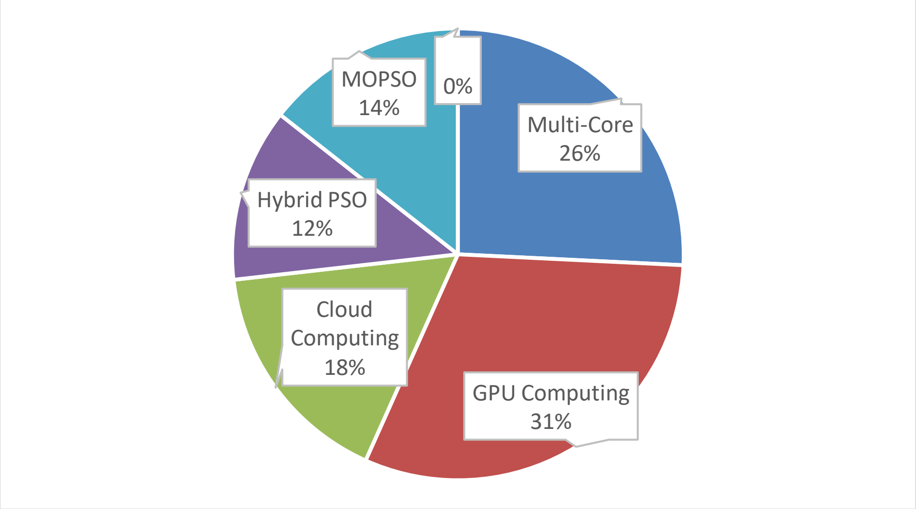


Fig 2.2 Pie chart represents the distribution of the published research papers using PSO Parallel Implementation.

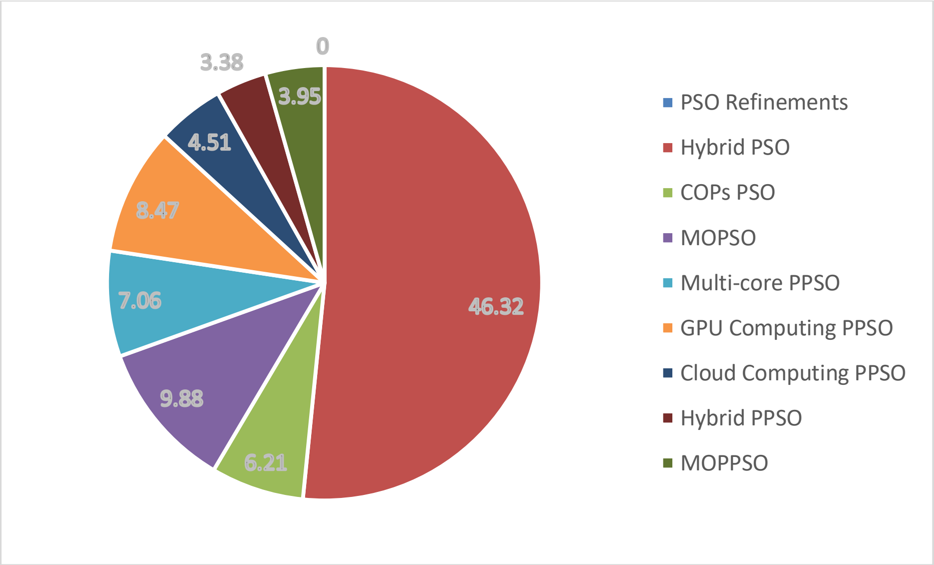


Fig 2.3 Pie chart represents the complete distribution of published research papers using both PSO and Parallel Implementation of PSO.

Fig2. Shows the distribution of all research papers published from 2003 to end 2019 for Parallel implementation of PSO using Multi-core, GPU Computing, Hybrid versions of PSO, and Multi-objective PSO. Whereas Fig 3.represents completes distribution of research papers published from 1995 to 2019 of all PSO papers.

2.9 **Conclusion**

PSO is computationally an intensive method and suffers with long run time while solving real world large optimization problems. The review article focus on PSO based algorithmic approach, communication topologies and parameter setting based approaches, hybridized approaches and multi-objective approach for its robustness and efficient for solving large optimization problems. For the present review article we have studied 1050 research articles from IEEE Journals, IEEE Transactions on Evolutionary Computation, Nature Computing, Soft computing, IEEE Access, Springer Nature, IEEE Transactions on Parallel Distribution, Proceedings of the IEEE International Conference on Control and Automation, Int. IEEE Conf. on Systems, Man, and Cyber; IEEE International Conference on Machine Learning and Cyber, Neuro Computing, Journal of Innovative Computing, Applied soft computing, IEEE

Latin America Transactions, IEEE Symposium Series on Computational Intelligence (SSCI), : IEEE Symposium Series on Computational Intelligence (SSCI), International Journal of Communication Systems, International IET Conference on Software Intelligence Technologies and Applications, The Scientific World Journal, and more. In review we found that search performance and convergence speed and run time are effected by different strategy. This paper gives a through survey with more emphasis from its development, improvements from its basic form, various methods derived from this algorithm are introduced. PSO as single objective optimizer motivated many researchers to extend to multi-objective and constrained optimization. The review will help the researchers to choose the correct constraint handling strategy for optimization problems in real life.

The publication chronological review of parallel PSO based on the parallelization strategy suggests descending order strategy MPI, GPU, Multi-core, OpenMP, Hadoop and MATLAB and then cloud computing. Literature survey on communication based suggest master-slave is most popular parallelization approach then coarse grained and finally fine grained approach. Hybrid approaches have little share as compared parallel PSO parallelization strategy. Multi-objective using parallelization strategy has also low share but is very effective in the real life optimization problems as this reduces computational time significantly.

CHAPTER 3

DEVELOPMENT OF NEW VERSION OF PARTICLE SWARM OPTIMIZATION

3.1 Introduction

Particle Swarm Optimization (PSO), one of the Bio-Inspired algorithm, was firstly introduced by Kennedy and Eberhart [6]. It is very simple, population- based search algorithm and is motivated from simulation of social and cognitive behaviour (Eberhart and Kennedy [7], Couzin et al. [382], Conradt et al.[381], Nagy et al.[387], zoltan toth et al [392]) of particles in the swarm. PSO is initialized with population of solutions called particles having random velocities and keep track while moving on its co-ordinates in the specified search space which are the personal best solution of the particles achieved so far called (Pbest). Among all the personal best values is the global best which is experienced by the particles in the swarm known as Gbest. PSO is found to be fast, robust, and less susceptible to entrapment due to its nature and can easily solve non-linear, non-differentiable multi-modal optimization problems even of higher dimensions. (Engelbrecht [383], Jain et al. [384], Jain et al. [386])

PSO has a long dynamic journey and have many variants. Shi and Eberhart [11, 14] outlined the selection criterion for inertia weights &velocity and proposed empirical study of PSO with linearly decreasing inertia weight from 0.9 to 0.4, keeping both acceleration constants c1 & c2 equal to 2 with asymmetric initial range. Clerc and Kennedy [25] suggested that PSO has common problem of stagnation because of premature convergence, especially in multi-modal functions. Ajith Abraham et al. [40] proposed various Inertia weight strategies that effect the convergence and exploration-exploitation trade off in PSO. The premature convergence behavior in PSO is a major problem and is studied by many researchers; however the particles tend to converge before true global minimum. Van den Berg [389] suggested GCPSO (Guaranteed Convergence PSO) for guaranteed convergence to local minimum by using different velocity equations. Further addressing to this problem Van den Berg (2002) developed MPSO (Multi-start PSO) which restarts itself whenever stagnation is detected by applying various criteria for detecting premature convergence. A different approach called Opposition based learning has been originally proposed by Tizhoosh [84] by considering counter estimates, opposite numbers, anti-chromosomes, counter actions, and opposite's weights by making revolutionary jumps in starting. Wang et al. [390] used the opposition based learning approach along with Cauchy mutation on the best particle to accelerate the convergence and avoid to get trapped in local minima. Cui Z et al. [391] proposed a dynamic adjustment strategy of optimum radius to improve the global search ability by updating the position of optimal solution by adaptive bat search algorithm with better accurate results as compared to other algorithms. A new mechanism called re-grouping (Reg PSO)/re-organize proposed by Evers and Ghalia [142] can efficiently re-group the swarm when premature convergence is detected and enables them to move towards global minimum, but it uses only global best of the swarm without further exploration. In this paper, a new approach PCPSO (Perfectly convergent PSO) is proposed which has both the qualities of exploration and faster convergence to escape from many local minima's even with noisy environment. The new algorithm differs from the previous work (Van den Berg [389], Tizhooh [84], Wang et al.[390], Evers and Ghalia [142]) opposition based PSO, opposition based PSO with Cauchy Mutation, GCPSO and MPSO(using Reg PSO). It was tested on the uni-modal, multi- modal with local minima and noisy environment and results were compared.

3.2 **Motivation**

Particle swarm optimization is originally inspired from social & personal behaviour, and is now being used widely for optimizing purposes. Several approaches are applied to achieve efficiency, but many a times they go away from original idea. Psychological findings suggest that the living organisms go for personal experience and they find the similarities in living organism who had experienced personally in life time. This thought gave me the inspiration to use Personal best in place of global PSO mechanism. Opposition based PSO, opposition based PSO with Cauchy Mutation, GCPSO and MPSO, as all have tendency of being trapped into the local minima whether few or many. Stagnation can be eliminated if premature convergence is diagnosed at appropriate time.

3.3 **Particle Swarm Optimization**

3.3.1 Standard Particle Swarm Optimization

PSO came into the existence in 1995 by Kennedy and Eberhart [6] derived from the basic Physics displacement equations called "Lbest" PSO was developed by Kennedy and Eberhart considered only information between close neighbourhood of two and six particles and investigated its effects on convergence. Eberhart et al. [7] developed a velocity clamping technique for controlling the initial rapid growth of velocity to prevent particles to leave the search space by choosing some maximum velocity in each dimension as $v_{\max j}$ .Each dimension j of velocity vector is checked and if the absolute exceeds then the j-th component velocity is revised. The velocity update is as follows:

$$v(k+1)_{i,j} = \begin{cases} v(k+1)_{i,j} \; if \quad -v_{max,j} \leq v(k+1)_{i,j} \leq v_{max,j} \\ v_{max,j} \; if \qquad v_{max,j} < v(k+1)_{i,j} \\ -v_{max,j} \; if \qquad v(k+1)_{i,j} < -v_{max,j} \end{cases} \quad (3.1)$$

Whereas $v_{max,j} \;=\; \lambda(u_j - l_j)$ (3.2)

λϵ (0, 1)

$v_{max,j}$ is fraction of the search space ,u_j, l_j are the upper and lower limits of the search space in j dimension and λ is the velocity clamping percentage, is usually lie between 0 and 1.Another technique for velocity clamping is based on the magnitude of velocity vector is to maintain the search direction of the particle and maintains the overall velocity vector magnitude. If the magnitude of the particle velocity reaches a selected limit, then the whole velocity vector is modified such that its magnitude is within limits and its direction is retained and is as follows:

$$v(k+1)_i = \begin{cases} v(k+1)_i & if\ \|v(k+1)_i\| \le v_{max} \\ \dfrac{v_{max}v(k+1)_i}{\|v(k+1)_i\|} & if\ \|v(k+1)_i\| > v_{max} \end{cases} \quad (3.3)$$

$$v_{max} = \lambda \sqrt{\sum_{j=1}^{n} \left(u_j - l_j\right)^2} \quad (3.4)$$

$$= \lambda\|U - L\| \qquad (3.5)$$

Whereas ||·|| represents Euclidean norm, $U = [u_1, u_2, \dots, u_n]^T$ and $L = [l_1, l_2, \dots, l_n]^T$ are the n dimension vectors which are bound in search space for dimension j. The value of v_{max} is calculated on some fraction of the maximum step size. This strategy does not alter the velocity direction but retains the information and components of velocity vector can be modified even though there are large components resulting in the particles to move slowly in most directions due to large velocity in other dimension. Kennedy [9] carried out an analysis of this algorithm for social interaction with the latest four types of models as full model, cognition model only, social model only and selfless model and indicated that cognition and social model only worked well but did not want to substitute the core algorithm as it would result in premature convergence. In starting PSO was without inertia weight but in (1998), first time Shi and Eberhart [11] introduced constant inertia weight and this algorithm was called standard PSO.PSO is initialized by initial solutions of the particles moving in the search space, each particle is represented by a position and velocity and keep updating as follows:

$$x_j(k+1) = x_j(k) + v_j(k+1)(3.6)$$

$$v_j(k+1) = \omega v_j(k) + c_1r_1\big(p_j(k) - x_j(k)\big) + c_2r_2\big(g(k) - x_j(k)\big) \quad (3.7)$$

Where, j =1, 2, 3,…i

k+1 denotes next iteration, k is the current iteration number, v_j is velocity of the particle j,x_j is position of the particle j, ω is Inertia weight factor, c_1,c_2 are acceleration factors,$p_j$is personal best of particle j, g is the global best of the entire swarm,r_1,r_2 are pseudo random numbers between 0 and 1.

Eberhart and Shi [14] suggested a better strategy of inertia weight in a linearly decreasing way from 0.9 to 0.4 for improved exploration and optimal global optima Maurice Clerc [15] proposed to use constriction factor χ which ensures better convergence and ability to control velocity in PSO algorithm as follows:

$$v_j(k+1) = \chi[v_j(k) + c_1r_1\left(p_j(k) - x_j(k)\right) + c_2r_2\left(g(k) - x_j(k)\right)] \qquad (3.8)$$

$$X = \frac{2}{\left|2-\phi-\sqrt{\phi^2-4\phi}\right|} \qquad \text{where } \phi = c_1+c_2,\ \phi > 4 \qquad (3.9)$$

.Again, Eberhart and Shi [18] proposed the comparison of two techniques using constriction factor χ and inertia weight ω are used which were found to be mathematically equivalent and linearly decreasing inertia weight from 0.9 to 0.4 was used for better results. Clerc and Kennedy [25] suggested to use constriction factor χ which eliminate the need for velocity clamping. Trelea [30] investigated the convergence boundaries, convergence point and procedure for parameter setting with inertia weight ω =0.6 and ϕ1 = ϕ2 =1.7, resulting in improved performance of the PSO. Ratnaweera [32] suggested adaption rule for reinitialize the velocity of particles with the values of ϕ1 to be decreased and ϕ2 is increased for better exploration and exploitation in search space. Bratton and Kennedy [34] defined a standard PSO as a basis for researchers to act as common grounding to work from. Nakagawa [37] suggested by adding a random number to the particle velocity for better control on velocity depending on the distance from global best particle location. Bonyadi and Michalewicz [45] examined the behaviour of particles and proposed that during the exploration the exploration the particles oscillate in various patterns in four classes based on maximum oscillation frequency and there boundaries do not depend on the number of dimensions. The standard PSO flow chart is shown in fig. 1 as follows:

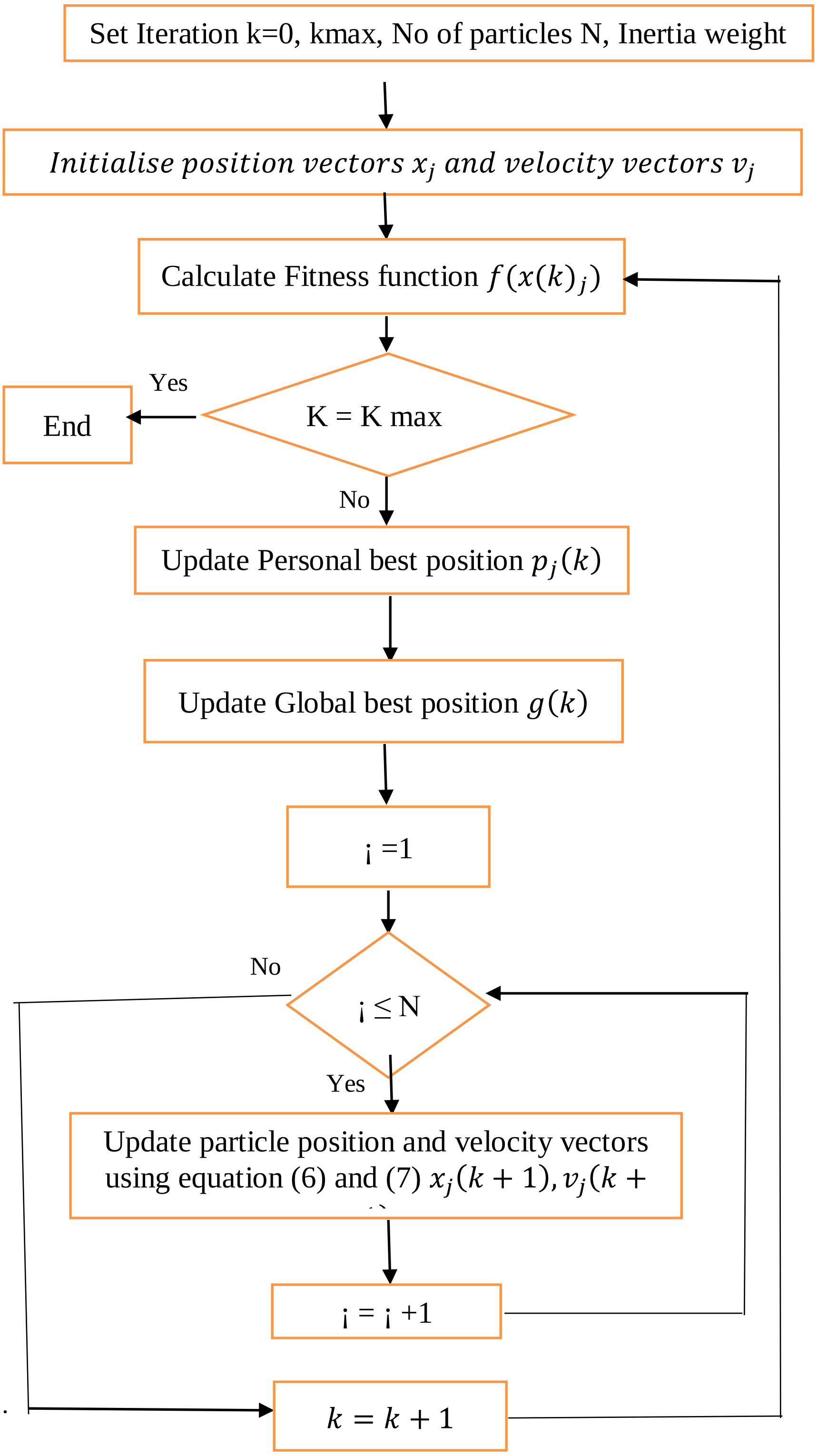

Set Iteration k=0, kmax, No of particles N, Inertia weight
$Initialise\ position\ vectors\ x_j\ and\ velocity\ vectors\ v_j$
Calculate Fitness function $f(x(k)_j)$
Yes
End
K = K max
No
Update Personal best position $p_j(k)$
Update Global best position $g(k)$
¡ =1
No
¡ ≤ N
Yes
Update particle position and velocity vectors using equation (6) and (7) $x_j(k+1), v_j(k+$
¡ = ¡ +1
$k = k + 1$

Fig.3.1 Standard PSO flow chart

Again Bonyadi and Michalewicz [47] investigated the relationship between the base frequency and correlation between particle positions and suggested that the particle with small base frequency will show smooth trajectories while particles with high base frequency will show more oscillations with large steps between positions.

3.3.2 **Guaranteed convergence particle swarm optimization (GCPSO)**

The concept behind the GCPSO is to induct a particle whose personal best is equal to the global best and that particle will search the global minimum in that part of the search space, so the velocity eq. will be as follows:

$$v_j'(k+1) = -x_j{}'(k) + g(k) + \omega v_j'(k) + \rho(k)(1-2r) \qquad (3.10)$$

$$x_j' = g(k) + \omega v_j'(k) + \rho(k)(1-2r) \qquad (3.11)$$

Where,

$-x_j{}'(k)$ Resets the particle position to global best position g(k) and $\omega v_j'(k)$ help in searching current direction,

r is a vector randomly generating numbers between 0 and 1,

ρ(k) (1-2r) generates a random search in the neighborhood area of global best particle,

ρ(k) is the diameter of random search space defined as follows:

$$\rho\,(k) = \rho(k+1) = \begin{cases} 2\rho(k) & \text{Successes} > \text{sc} \\ (0.5)\rho(k) & \text{failure} > \text{fc} \\ \rho(k) \quad \text{otherwise} & \end{cases} \qquad (3.12)$$

# failures>f_c

#successes (k+1)> #successes (k) => # failures (k+1) =0 and,

# failures (k+1)> # failures (k)=> #successes (k+1)=0

Where, #successes & # failures are the number of consecutive successes or failures, s_c= 15 &f_c=5 are the threshold parameters.

The strategy of ρ setting is to punish the low performers and reward to good performers. This algorithm do well with uni-modal functions with low number of particles, but get trapped to local optima in case of multi modal functions. But it is faster and has better convergence than standard PSO.

3.3.3 **New variant PCPSO (Perfectly Convergent PSO)**

The purpose of this new variant is to avoid premature convergence which leads to stagnation and to give opportunity to the personal best particles in place global particle as they give better exploration in the search space. In this new variant, I have introduced an additional particle as used in GCPSO but it will search around personal best position in place of global position. Searching areas near to global position lacks in exploration and faces to get trapped in multi-modal problems with one or many local minima, keeping in view the new velocity update is as follows:

$$v_j'(k+1) = -x_j{}'(k) + pbest(k) + \omega v_j'(k) + \rho(k)(1-2r) \qquad (3.13)$$

Whereas other particles in the swarm will update the velocity as per this new variant:

$$v_j'(k+1) = \omega x_j{}'(k) + c_1 r_1\big(p_j(k) - x_j(k)\big) + c_2 r_2\big(-x_j(k)\big) \qquad (3.14)$$

Where, $-x_j{}'(k) + pbest(k)$ component will make the search in the personal best region, $\omega v_j'(k)$ gives the momentum to search in current direction, ρ(k) (1-2r) generates a random search in the neighborhood area of personal best particle with side length of $2\rho(k)$, ρ(k) is the diameter of random search space defined as follows:

$$\rho(k+1) = \begin{cases} 2\rho(k) & Successes > sc \\ (0.5)\rho(k) & failure > fc \\ \rho(k) & otherwise \end{cases} \qquad (3.15)$$

#successes (k+1)> #successes (k) => # failures (k+1) =0 and,

# Failures (k+1)> # failures (k) => #successes (k+1) =0

Where, #successes & # failures are the number of consecutive successes or failures, s_c= 15 &f_c=5 are the threshold parameters and can be finely adjusted.This technique uses an adaptive ρ for obtaining the optimal sampling volume in its current state of this method. If a particular value of repeatedly results in a success then a large sampling volume is chosen to increase the maximum distance travelled in one step. On the other hand when ρ produces consecutive failures fc then sampling volume is too big and must therefore be reduced. In the end stagnation is absolutely prevented if ρ> 0 for all steps.

This following is convergence criteria requirements for the Perfectly convergent PSO which works on the basis as defined by Solis and Wets [388] They investigated the convergence of stochastic search algorithms, in particular those of pure random search algorithm providing conditions by which the called global search or local search algorithm.

Global search convergence theorem: Suppose that f is a measurable function, S is a measurable subset of R^n and H1 and H2 are satisfied. Let $(y^k)_{k=0}^{\infty}$ be a sequence generated by algorithm, then $\lim_{k\to\infty} P\,[y^k \ \in R_M] = 1$

Where $P[y^k \ \in R_M]$ is the probability that at step k, the point y^k generated by the algorithm is in R_M.

Proposition: Given a function f from $R^n\ to\ R$ and S is a subset of R^n. We seek a point y in S which minimizes f on S or at least which yields an acceptable approximation of the *infimum* of f on S.

This proposition sets out the concept of what the global optimizer must generate as output, provided the function f and search space S. The stochastic algorithm uses basic random search algorithm to perform this role.

Basic Random search Algorithm

Step 0.Find y_0 in S and set k= 0

Step 1.Generate ξ_k vector in sample space $(R^n, \mathcal{B}, \mu_k)$.

Step 2. Set$y_{k+1} = D(y_k, \xi_k)$, chooseμ_{k+1} and set k= k+1 and return to step 1.

Where y_0a random initial starting point in S,μ_k is a probability measure (corresponding to distribution function on R^n) on $\mathcal{B}$ and $\mathcal{B}$ is anσ-algebra of subsets ofR^n, D is a function that builds the better solution to problem than current solution and satisfies the following condition:

H1$f(D(y,\xi)) \leq f(y)$ $and\ if\ \xi \epsilon S\ then, f(D(y,\xi)) \leq f(\xi)$ (3.16)

Global convergence mean that with probability 1, the sequence $f(y^k)_{k=1}^{\infty}$ converges to *infimum* of f on S but if it occurs at a point at which f is singularly discontinuous, then we cannot find minimum point. Now replace the search for infimum by essential infimum δ as follows:

$\delta = \inf(t: v[y \epsilon S|f(y) < t| > 0)$ (3.17)

Whereas $v\ |A|$ is the Lebesgue measure on set A.and typically is the n-dimensional volume of set A. Therefore we aim aspire to produce convergence towards a small region a minimum solution. The optimum region can be defined as follows:

$R_M = (x \epsilon S|f(y) < \delta + \epsilon)$ (3.18)

Where δ > 0 and the point found in this region is good approximation to the true global minima .In the execution of this algorithm, we distinguish between local and global search algorithm based on the $\mu_k$properties of sequence of probability measures used. In local search algorithms μ_k with bounded supportM_k, such that $v|S \cap M_k| < v|S|$ possibily with all values of k.Second assumption for global search is as follows:

H 2 For any (Borel) subset of A of S with $v[A] > 0$, we have

$\prod_{k=0}^{\infty}(1 - \mu_k[A]) = 0$ (3.19)

Whereas $\mu_k[A]$ is the probability of A generated by μ_k.This implies that for any subset A of S with positive v, the probability of repeatedly missing the set A with random samples must be zero and the likelihood of sampling a point in the optimum region $R_M \subset S$ must be non-zero.

H3. To any $y_0\ \epsilon S$, there corresponds a β >0 and a$0 < \eta \leq 1$ such that:

$\mu_k[(dist(D(y,\xi), R_M) < dist(y, R_M) - \beta)\quad or\ (D(y,\xi), R_M)] \geq \eta$ (3.20)

For all k and all y in the compact set $L_o = [y\ \epsilon S|f(y) \ll f(y_o)]$

Whereas $dist(D(y,A))$denotes the distance between y and set A defined as:

$$dist(y, A) = inf_{c\epsilon A}^{inf} dist(y, c)$$

At every step in a local optimization method can move y in close approximation to the optimum region by at least distance β or y is in optimum region with a probability greater or equal to η

Local search convergence theorem: Suppose that f is a measurable function, S is a measurable subset of R^n and H1 and H3 are satisfied. Let $(y^k)_{k=0}^{\infty}$ be a sequence generated by algorithm, then $\lim_{k\to\infty} P\,[y^k \in R_M] = 1$

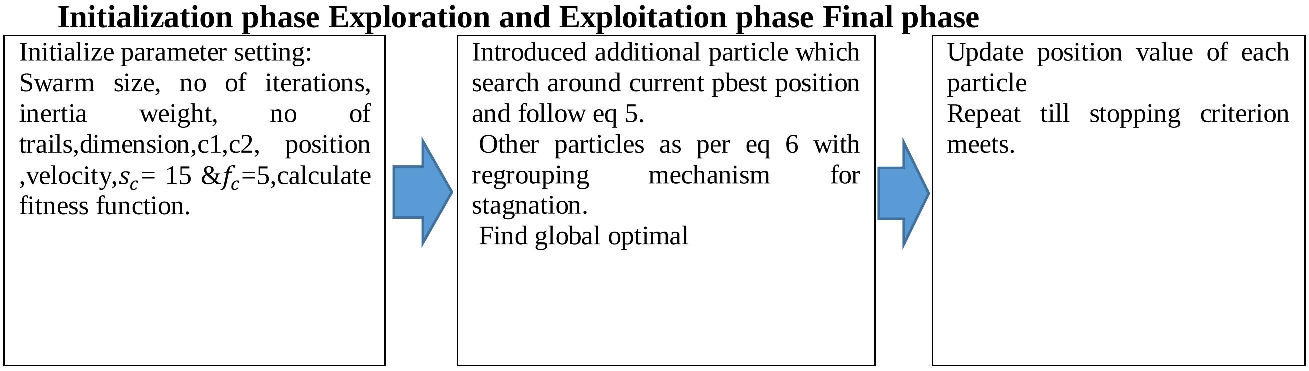
Where $P[y^k \in R_M]$ is the probability that at step k, the point y^k generated by the algorithm is in the optimum region R_M.

The Perfect convergent PSO (PCPSO) satisfies the necessary and sufficient condition to converge because:

1. PCPSO can always generate a sample around a point.

2. PCPSO algorithm make a non-degenerate sampling volume with non- zero probability of sampling a point near to optimum region, irrespective of the initial state of the particles.

Following is the block diagram explanation of PCPSO procedure as shown in fig 3.2.



Initialization phase Exploration and Exploitation phase Final phase

Initialize parameter setting: Swarm size, no of iterations, inertia weight, no of trails,dimension,c1,c2, position ,velocity,s_c= 15 &f_c=5,calculate fitness function.

Introduced additional particle which search around current pbest position and follow eq 5. Other particles as per eq 6 with regrouping mechanism for stagnation. Find global optimal

Update position value of each particle Repeat till stopping criterion meets.

Fig3.2 Block diagram of PCPSO

Basically, this version gives chance to all particles in participating whether they are in exploring stage /got better personal best than the previous iteration or at the verge of global optima and makes it true global search algorithm. This algorithm overcomes the limitation of GCPSO.

3.4 Experiments

3.4.1 Benchmark Problems

Five of the well-known benchmark functions are used in this paper. Two functions f1, f2 are uni-modal functions & f3, f4 are multi-modal functions while f5 is multi-modal with noise. Generally, multi-modal functions are the most difficult functions for optimization. Symmetric initialization is used in this paper, where the initial population is uniformly distributed in the entire search space. All the functions used in this paper are to be minimized. Table 3.1 shows the detail of these functions.

Table 3.1 Benchmark Dimensionality

Benchmark	Objective function	Search Space (Initialization Range)	Optimal Function value	Number of Dimensions
Sphere	Min f1(x) = $\sum_{i=1}^{n} x_i^2$	-5.12≤ x_i ≤ 5.12	0	30
Weighted Sphere	Min f2(x) =$\sum_{i=1}^{n} i.x_i^2$	-5.12≤ x_i ≤ 5.12	0	30
Ackley	Min f3(x) = 20+e-20$e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}}$ -$e^{\frac{\sum_{i=1}^{n}\cos(2\pi x_i)}{n}}$	-30≤ x_i ≤ 30	0	30
Griewenk	Min f4(x) = 1+$\sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	-600≤ x_i ≤ 600	0	30
Quartic with Noise	Min f5(x) = $random[0,1) + \sum_{i=1}^{n} i.x_i^4$	-1.28≤ x_i ≤ 1.28	0	30

3.4.2 Parameter Settings:

The selection of parameters have strong impact on the performance of PSO algorithms .In this paper, following parameters were set as shown in Table 3.2 for two groups of experiments. First group uses linearly decreasing inertia weights from 0.9 to 0.4 and in the second group constant inertia weight $\omega = 0.729844$ were used without and with 0%, 15% velocity clamping in their respective ranges. OPSO is without mutation whereas OPSO with Cauchy Mutation uses 0.5 mutation probability.

Table 3.2 The Parameter setting for experiments

Parameter Name	Setting	Reference
Acceleration constants c1	1.49618	[8]

and c2		
Inertia weights, ω	linearly decreasing from 0.9 to 0.4	[8]
	0.729844	[8]
Swarm size	10	
Maximum iterations	5000 per trial	
Dimensions	30	
Number of trials	05	
Velocity clamping	$\lambda = 0\%$ and $\lambda = 15\%$,	[17]
Stagnation threshold,ϵ	1×10^{-6}	[17]
regrouping factor =1.2ε^{-1}	50 for uni-modal functions	[17]
	1,20,000 for multi modal functions	[17]
Mutation Probability	0 for only OPSO	[16]
	0.5 for OPSO with Cauchy mutation	[16]
No of Starts in MPSO	100	[14]

3.4.3 Experimental setup

This section will examine behavior of PCPSO algorithm by systematically manipulating the parameter that affects the rate of convergence. Each algorithm was tested on all the benchmarks listed in table 3.1.For each experiment the MatLab programme was run 05 times of 5000 iteration each with all statistical values like mean ,median, best and worse were recorded on MATLAB-2015a,excuted on Compaq 6720s,intel core2 Duo. PCPSO uses only 500 iterations rest all the PSO variants uses 5000 iterations.

3.5. Computational results and discussion

Table 3&4 show the comparison of proposed algorithm PCPSO with the OPSO, OPSO with Cauchy mutation, GCPSO (using RegPSO) and MPSO (using RegPSO) for the function f1 to f5 using linearly decreasing inertia weight and constant inertia weight with velocity clamping at 0% and 15% respectively. Whereas "Mean", "Best", "Worse" are the mean of the last function value of the 5 trails conducted, Best is the minimum function value among 5 trials and worse is the maximum function value achieved over 5 trials. Table 3.3 &3.4 also shows highly impressive optimal solution with merely 500 iterations as compared to other variants which have 5000 number of iterations as stopping criteria. Bold values are the best values in the Table 3.3 & Table 3.4 respectively. Fig3.5 and Fig 3.6 shows the plot between Mean

Function value and the number of iterations with linear inertia weight and with constant inertia weight with and without velocity clamping (0% and 15%) respectively.

3.5.1 Algorithms using linearly decreasing inertia weight

The idea behind using the linearly decreasing inertia weight is that it gives more exploration during the initial stages of the iteration before they start to converge. Fig3.5 and Fig3.6 shows the characteristic of various algorithms between mean function value and the number of iterations applied to different benchmarks function. Moreover, the different algorithms fail to detect stagnation stage at appropriate time when applied on the benchmark function. Following table 3.3 shows the statistical results using linearly decreasing inertia weight

The two uni-modal functions f1 & f2 with all the statistical values (Median, Mean, Best& Worse) show the stronger property of convergence of PCPSO with velocity clamping 15% as compared to other algorithms. The performance of GCPSO, MPSO, and OPSO &OPSO with Cauchy mutation showed the results for 5000 iterations but PCPSO was able to converge in merely 500 iterations. Function f1(spherical) without velocity clamping shows OPSO is worse among the competitive performer, PCPSO detected stagnation in just 100 iterations without getting trapped and acted too fastly but when 15% velocity clamping is applied to all algorithms, it again showed excellent statistical values. In case of Function f2 (weighted sphere) without velocity

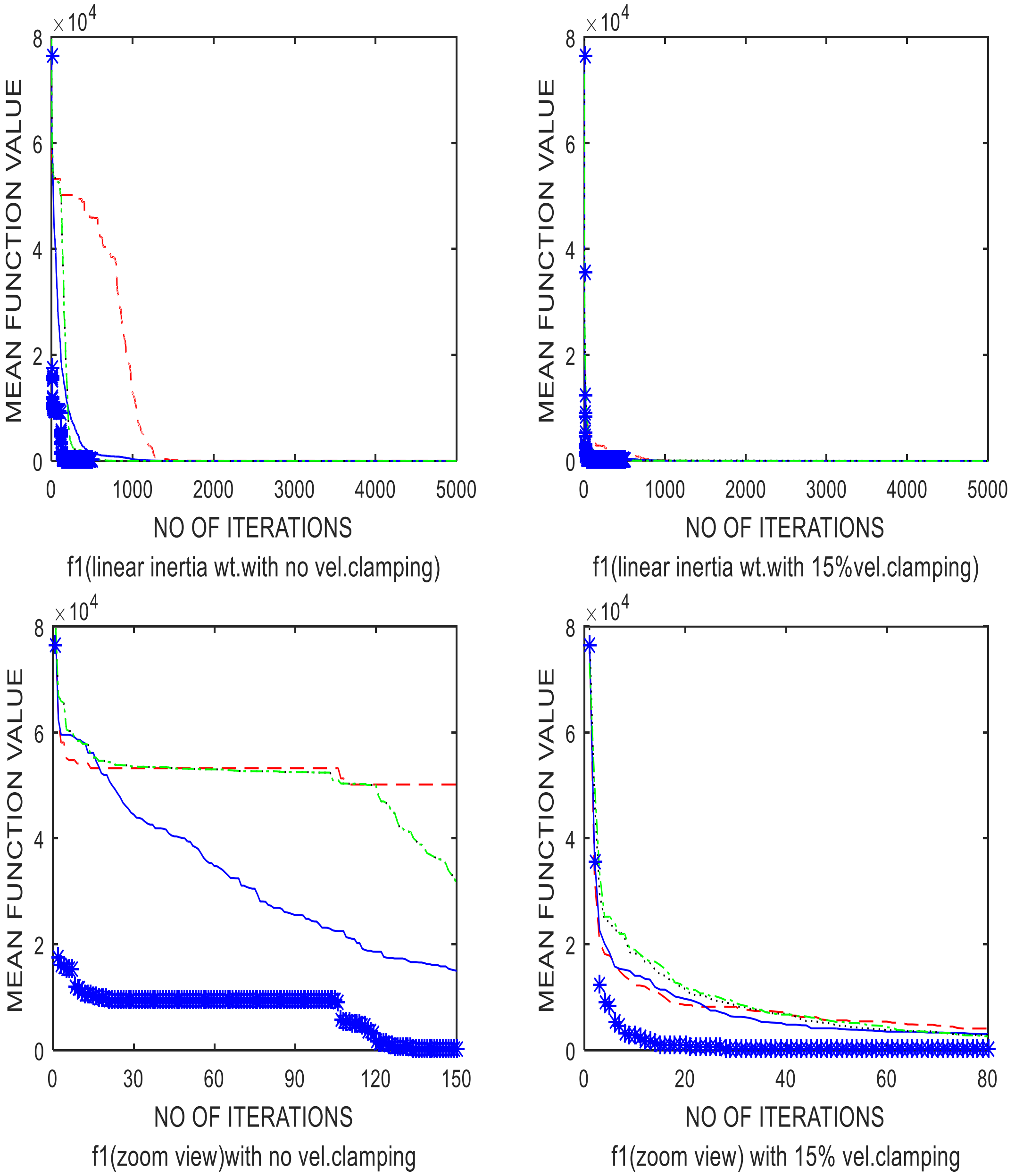
Table 3.3 Results using linearly decreasing inertia weight from 0.9 to 0.4

Function	**Statistics**	**Opposition based PSO**		**OPSO with Cauchy Mutation**		**GCPSO**		**MPSO**		**PCPSO**	
		λ = 0%	λ=15%	λ=0%	λ=15%	λ=0%	λ=15%	λ=0%	λ=15%	λ=0%	λ=15%
f_1	Median	$3.4823e^{-20}$	$2.0634e^{-20}$	$3.5559e^{-22}$	$3.1153\ e^{-20}$	$4.1189\ e^{-24}$	$3.2356\ e^{-27}$	$5.7145\ e^{-22}$	$1.6391\ e^{-22}$	$2.4009e^{-53}$	$\mathbf{1.2209e^{-56}}$
	Mean	$5.5831e^{-25}$	$8.3488e^{-20}$	$4.9064e^{-22}$	$9.5368\ e^{-22}$	$7.7595\ e^{-24}$	$1.5818\ e^{-22}$	$8.5252\ e^{-22}$	$5.0625\ e^{-5}$	$7.2725e^{-53}$	$\mathbf{3.4897e^{-55}}$
	Best	$6.6045e^{-27}$	$1.8298e^{-24}$	$4.1694e^{-22}$	$3.5758\ e^{-25}$	$3.4937\ e^{-26}$	$2.9763\ e^{-22}$	$1.6433\ e^{-24}$	$9.2299\ e^{-42}$	$4.0611e^{-55}$	$\mathbf{1.7138e^{-58}}$
	Worse	$2.7909e^{-24}$	$2.8776e^{-20}$	$2.3179e^{-22}$	$4.5747\ e^{-25}$	$2.4474\ e^{-22}$	$7.8928\ e^{-22}$	$4.2214\ e^{-20}$	$2.5303\ e^{-7}$	$2.5672e^{-52}$	$\mathbf{1.5571e^{-54}}$
f_2	Median	0.00017596	$2.1495e^{-7}$	$1.4849e^{-22}$	$1.7497\ e^{-27}$	$6.4827\ e^{-22}$	$1.9702\ e^{-9}$	$5.3606\ e^{-22}$	$1.0372\ e^{-9}$	$4.5032e^{-53}$	$\mathbf{1.7859e^{-56}}$
	Mean	0.064589	0.006511	$2.2212e^{-22}$	$1.3407\ e^{-26}$	$6.553\ e^{-22}$	$1.2997\ e^{-5}$	$6.9548\ e^{-22}$	$2.2595e^{-5}$	$7.4025e^{-53}$	$\mathbf{1.3262e^{-56}}$

	Best	$3.6428e^{-6}$	$4.8283e^{-3}$	$1.3726e^{-24}$	$2.6444\ e^{-27}$	$1.0525\ e^{-22}$	$5.3936\ e^{-27}$	$1.7164\ e^{-22}$	$2.1909\ e^{-24}$	$1.0972e^{-53}$	$\mathbf{8.7345e^{-60}}$
	Worse	0.31705	0.032554	$1.0723e^{-20}$	$5.4714\ e^{-26}$	$1.6167\ e^{-22}$	$6.4735\ e^{-5}$	$2.3343\ e^{-22}$	0.00010813	$2.3649e^{-52}$	$\mathbf{2.5644e^{-56}}$
f_3	Median	18.3748	0.0050057	8.8898	0.0007911	0.00036866	0.00010449	0.0033141	0.00012259	$\mathbf{2.6166e^{-7}}$	0.028696
	Mean	18.2143	0.14123	8.827	0.13462	0.0004432	0.0018528	0.36086	0.00045146	$\mathbf{9.4964e^{-6}}$	0.09583
	Best	16.8906	0.00067494	8.2601	0.00010873	$7.7437e^{-5}$	$2.4579\ e^{-5}$	0.00021775	$6.8047\ e^{-5}$	$\mathbf{1.518e^{-8}}$	0.0077893
	Worse	19.2307	0.67394	9.1203	0.66892	0.0010145	0.0086033	1.7769	0.0012416	$\mathbf{4.6203e^{-5}}$	0.22102
f_4	Median	0.059178	0.085436	0.078426	0.09117	0.093275	0.03677	0.01478	0.06671	**0**	0.00020535
	Mean	0.074378	0.091973	0.080624	0.09583	0.072113	0.041237	0.025541	0.08348	**0**	0.0099559
	Best	0.049064	0.0098769	$1.6403e^{-2}$	0.041667	$7.7805\ e^{-7}$	0.012321	0.0098677	0.034421	**0**	1.7131
	Worse	0.13677	0.1688	0.16414	0.16316	0.12279	0.0809901	0.075957	0.1693	**0**	0.024901
f_5	Median	0.011207	0.011963	0.0086609	0.010514	0.016763	0.0060504	0.015058	0.0063105	0.00079656	**0.00049606**
	Mean	0.012741	0.012125	0.010822	0.010982	0.019234	0.0062488	0.016059	0.00715	0.00067099	**0.00048536**
	Best	0.0091362	0.0074706	0.0062412	0.0066133	0.011164	0.0043831	0.0046659	0.0045855	6.0567	**0.00013364**
	Worse	0.016367	0.016588	0.017634	0.013956	0.030663	0.0091254	0.030966	0.010239	0.0013707	**0.00081387**

Clamping GCPSO and MPSO performed better in about 200 iterations than OPSO and OPSO with Cauchy mutation are the worse while PCPSO performed in about 100 iterations with remarkable statistical results. More over with 15% velocity clamping PCPSO is ahead of other algorithms. Function f3 & f4 are multi- modal and are considered as the difficult benchmark because of many wells with local minima, but PCPSO results shows highly impressive even by using the high dimensions of 30.Function f3 (Ackley) has the lowest mean and median values without velocity clamping in the case of PCPSO in 200 iterations followed by MPSO and GCPSO but OPSO with Cauchy mutation and OPSO faced stagnation problem and got trapped in local minima. Function f4 (Griewenk function) shows all statistical values higher in other algorithms but in our case it is perfectly 0 without velocity clamping and is very impressive as it is capable to overcoming the stagnation problem without getting trapped in the local minima. It shows the stronger convergence property of PCPSO and the rate of convergence of any stochastic algorithm is dependent on the volume of sample space as the number of points in the sample space increases exponentially with the dimension in search space, then other algorithm takes longer to find the global minima in a finite number of iterations and fails very rapidly as the dimension increases.

Fig.3.5 Comparison of Mean Function value v/s Number of Iterations for linearly decreasing inertia weight with and without velocity clamping.

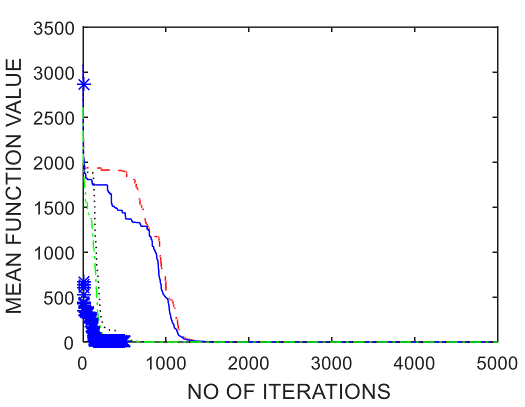


f1(linear inertia wt.with no vel.clamping)

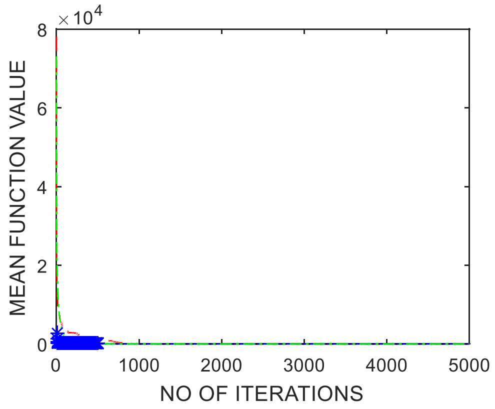
f1(linear inertia wt.with 15%vel.clamping)

f1(zoom view)with no vel.clamping

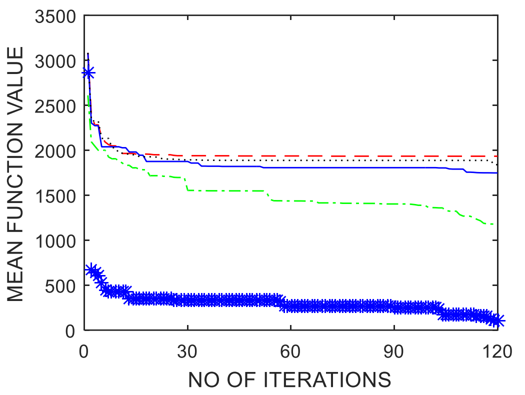
f1(zoom view) with 15% vel.clamping



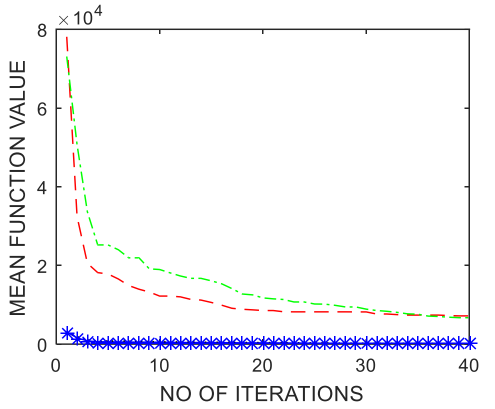
f2(linear inertia wt.with no vel.clamping)



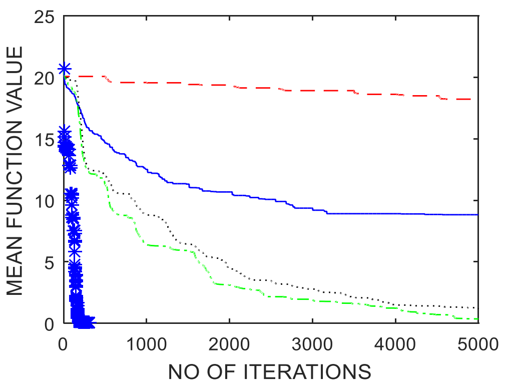
f2(linear inertia wt.with15%vel.clamping)



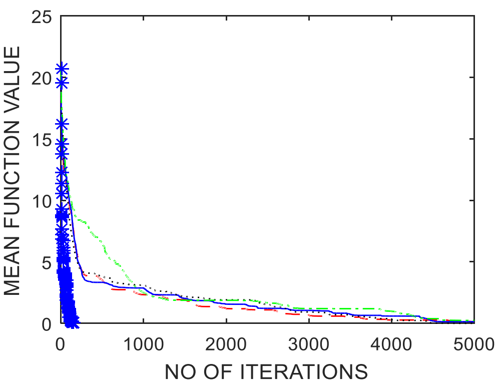
f2(zoom view)with no vel.clamping



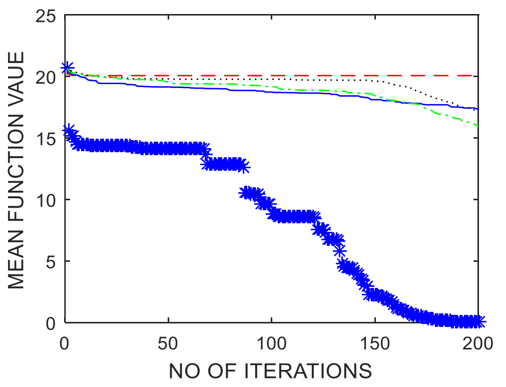
f2(zoom view)with 15%vel.clamping



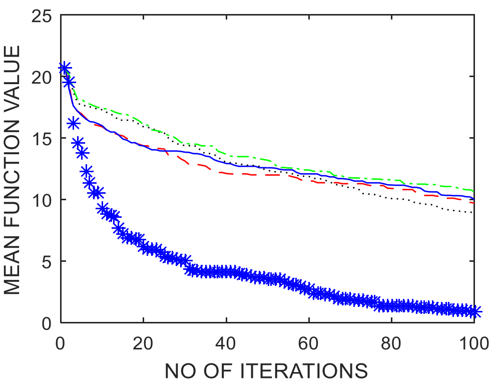
f3(linear inertia wt.with no vel.clamping)



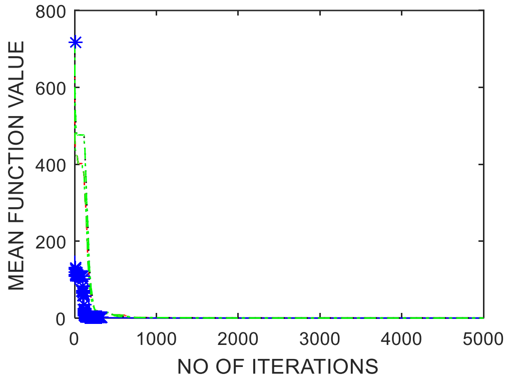
f3(linear inertia wt.with 15%vel.clamping)



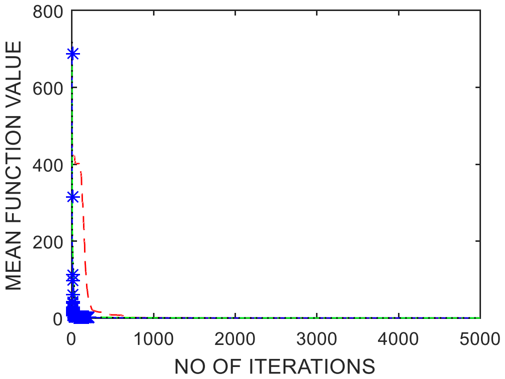
f3(zoom view) with no vel.clamping



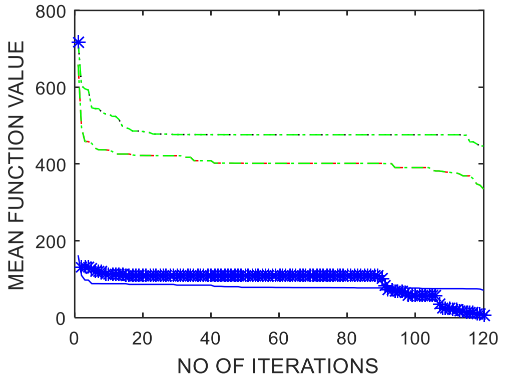
f3(zooom view)with 15%vel.clamping



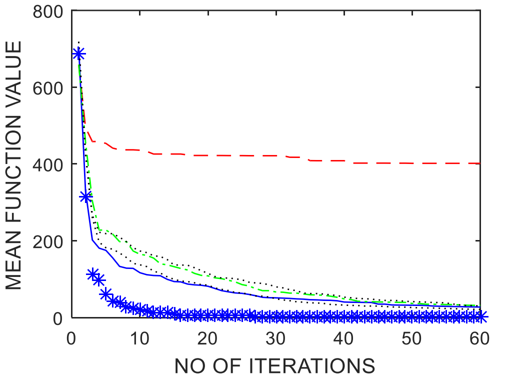
f4(linear inertia wt.with no vel.clamping)



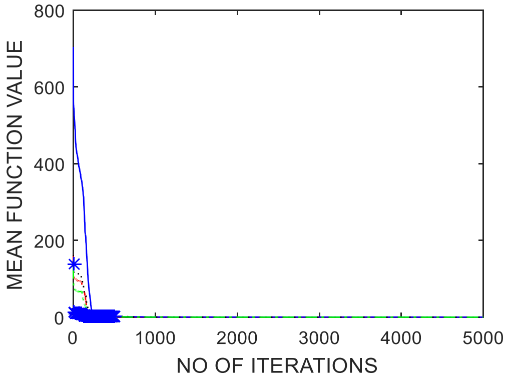
f4(linear inertia wt.with 15%vel.clamping)



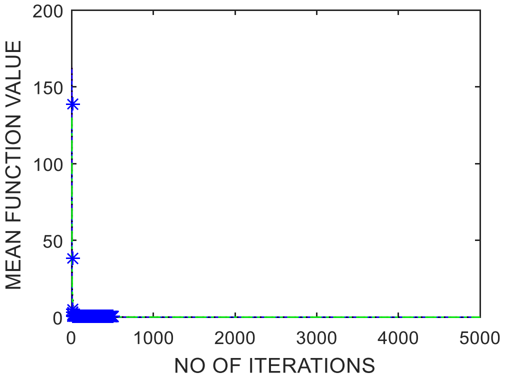
f4(zoom view)with no vel. clamping



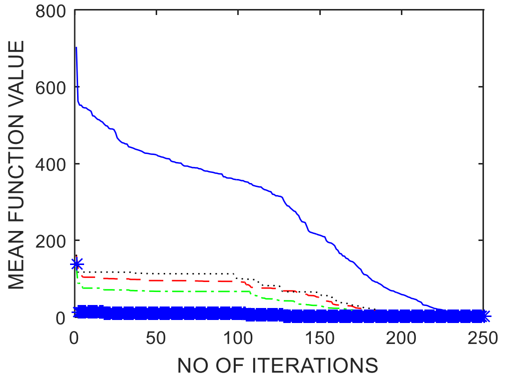
f4(zoom view)with 15%vel.clamping



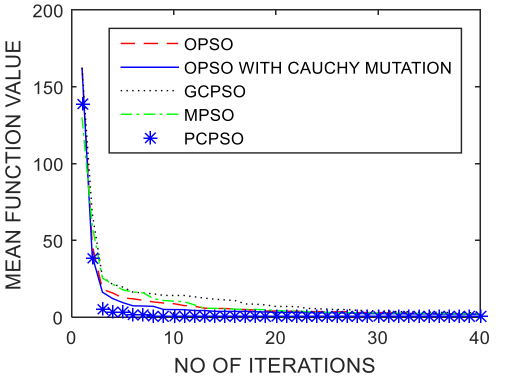
f5(linear inetia wt with no vel.clamping)



f5(linear inertia wt with 15%vel.clamping)



f5(zoom view) with no vel.clamping



f5(zoom view) with 15% vel.clamping

. Function f5(Quartic with noise) which is noisy in nature, but PCPSO again showed even good results at 15% velocity clamping as compared to GCPSO and MPSO (both using RegPSO) using linear decreasing inertia weight.Figure3.5.shows superior performance between PCPSO and comparison algorithms.

3.5.2 Algorithms using constant inertia weight

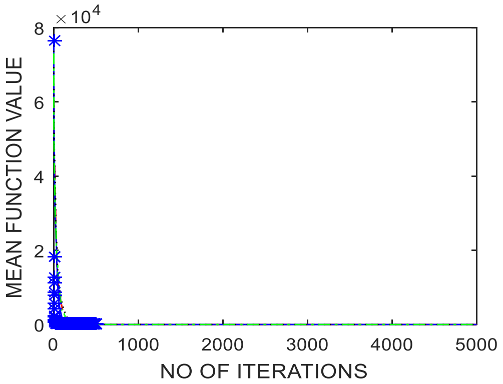
Table 3.4 Results using Constant inertia weight $\boldsymbol{\omega = 0.729844}$

Function	**Statistics**	**Opposition based PSO**		**OPSO with Cauchy Mutation**		**GCPSO**		**MPSO**		**PCPSO**	
		λ = 0%	**λ=15%**	**λ=0%**	**λ=15%**	**λ=0%**	**λ=15%**	**λ=0%**	**λ=15%**	**λ=0%**	**λ=15%**
f_1	Median	0.50512	2.3584 e^{-6}	1.7504 e^{-30}	2.7199 e^{-27}	5.9203 e^{-13}	0.0045084	7.8125 e^{-25}	1.1436 e^{-8}	$\mathbf{2.3726e^{-}}$	$4.7686e^{-33}$
	Mean	0.42057	1.429 e^{-5}	1.9704 e^{-25}	1.4058 e^{-26}	9.7924 e^{-11}	0.055115	2.2379 e^{-23}	0.0017072	$\mathbf{1.1076e^{-}}$	$5.798e^{-32}$
	Best	0.0065457	9.7844 e^{-7}	9.1656 e^{-32}	1.5526 e^{-32}	1.1651 e^{-13}	1.0952 e^{-7}	2.6748 e^{-25}	1.8587 e^{-13}	$\mathbf{5.8641e^{-}}$	$9.7633e^{-37}$
	Worse	0.72107	5.3353 e^{-5}	9.852 e^{-25}	6.1768 e^{-26}	4.8781 e^{-10}	0.23801	1.08 e^{-22}	0.0075333	$\mathbf{2.7403e^{-}}$	$2.6407e^{-31}$
f_2	Median	7.9933 e^{-22}	1.5052 e^{-14}	4.4835 e^{-25}	9.0918 e^{-20}	2.6643 e^{-16}	9.9429 e^{-5}	3.6526 e^{-14}	2.278 e^{-14}	$9.5597e^{-3}$	$\mathbf{1.2316e^{-35}}$
	Mean	3.8258 e^{-20}	3.4048 e^{-10}	1.2779 e^{-23}	3.0692 e^{-14}	2.2606 e^{-14}	0.046509	4.2234 e^{-11}	9.3119 e^{-13}	$1.6567e^{-3}$	$\mathbf{1.5745e^{-33}}$
	Best	1.2528 e^{-22}	3.4531 e^{-18}	8.535 e^{-29}	2.1222 e^{-22}	1.7906 e^{-18}	1.5483 e^{-5}	9.2341 e^{-18}	9.3474 e^{-19}	$1.5093e^{-3}$	$\mathbf{5.1378e^{-37}}$
	Worse	1.6686 e^{-22}	1.7023 e^{-9}	5.5919 e^{-23}	1.5338 e^{-13}	5.6876 e^{-14}	0.23198	2.0971 e^{-10}	4.3858 e^{-12}	$7.876e^{-33}$	$\mathbf{7.7236e^{-33}}$
f_3	Median	1.1552	0.0011664	0.72782	2.4671 e^{-5}	1.3404	0012304	0.93142	2.5985 e^{-5}	$1.6692e^{-5}$	$\mathbf{1.625e^{-11}}$
	Mean	0.9547	0.60724	0.61036	0.0080875	0.83846	0.42031	0.83492	0.27705	$7.1372e^{-5}$	$\mathbf{4.2919e^{-10}}$
	Best	6.4216 e^{-5}	3.9595 e^{-6}	6.1661 e^{-6}	1.786 e^{-6}	0.0026449	6.8812 e^{-5}	0.0011311	2.2421 e^{-6}	$3.4887e^{-7}$	$\mathbf{3.5527e^{-15}}$
	Worse	1.3451	1.5331	1.1569	0.040361	1.5029	1.1552	1.1555	1.3404	0.00019627	$\mathbf{1.9138e^{-9}}$
f_4	Median	0.029459	0.041794	0.066173	0.012321	0.11799	0.044127	0.012321	0.046483	**0**	$7.5873e^{-11}$
	Mean	0.034354	0.058924	0.060203	0.019139	0.10866	0.045037	0.026556	0.046037	**0**	10.9232
	Best	0.017226	0.012316	1.036 e^{-10}	5.7732 e^{-15}	0.0098573	8.2934 e^{-14}	1.2517 e^{-12}	2.7978 e^{-14}	**0**	$2.1649e^{-14}$
	Worse	0.061127	0.15211	0.13949	0.056289	0.19819	0.093027	0.076154	0.088246	**0**	54.6153
f_5	Median	0.012674	0.017612	0.015082	0.017821	0.011793	0.0080003	0.0050304	0.0093713	0.00064179	**0.00042271**
	Mean	0.011904	0.018594	0.016964	0.017589	0.013309	0.0075724	0.0082326	0.0077502	0.0005975	**0.00058713**
	Best	0.0085033	0.012361	0.0077755	0.012093	0.010956	00052398	0.0040762	0.0041969	0.00014112	**0.00013636**
	Worse	0.013768	0.02903	0.027866	0.023442	0.016151	0.0091191	0.01438	0.010272	0.00095718	**0.0011979**

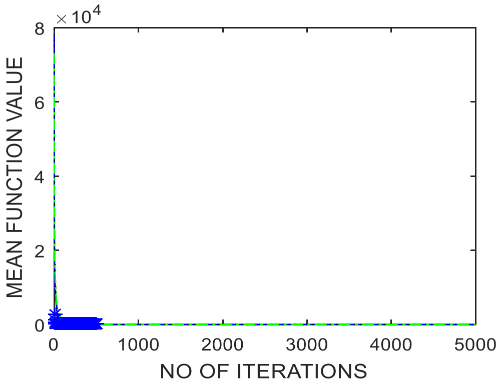
In this experimental set up inertia weight $\boldsymbol{\omega = 0.729844}$ is kept constant throughout the simulation run using MATLAB. Table3.4 clearly indicate that PCPSO shows the stronger convergence property whether it is uni-modal or multi-modal with many minima due to its better exploration & good strategy. The mechanism behind is very powerful which easily liberate particles in all the conditions and easily avoids premature convergence. Following table 3.4 shows the statistical results using linearly decreasing inertia weight.

The two uni-modal functions f1with 0% & f2 with15% velocity clamping shows all the statistical values (Median, Mean, Best& Worse) of PCPSO are lowest as compared to other algorithms. Multi-modal function f3 with 15%& f4 without velocity clamping again shows remarkable perforance.f4 (Griewenk function) shows quality solutions as again it has achieved perfectly zero as function values by using PCPSO showing convergent behaviour irrespective of the type of inertia weight used. Noisy function f5 again showed PCPSO with good results as compared to GCPSO, MPSO (both using RegPSO) and other algorithm using constant inertia weight.Figure2.shows efficient performance between PCPSO and comparison algorithm.

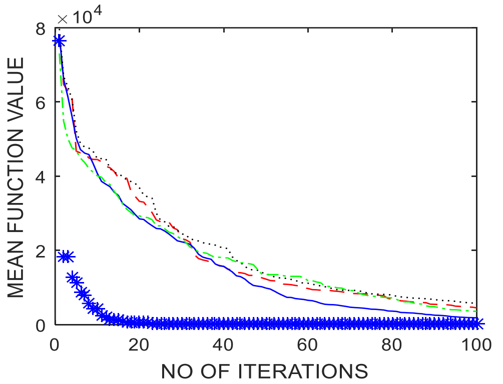
Fig3.6 Comparison of Mean Function value v/s Number of Iterations for Constant inertia weight with and without velocity clamping.



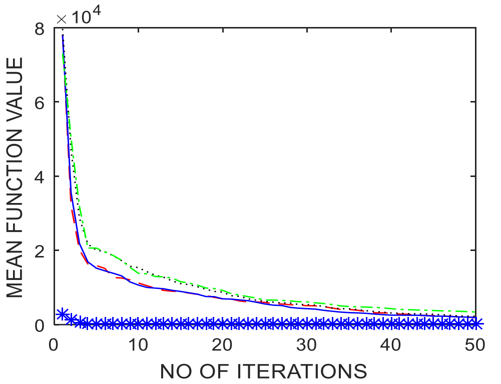
f1(const. inertia wt with no vel.clamping)



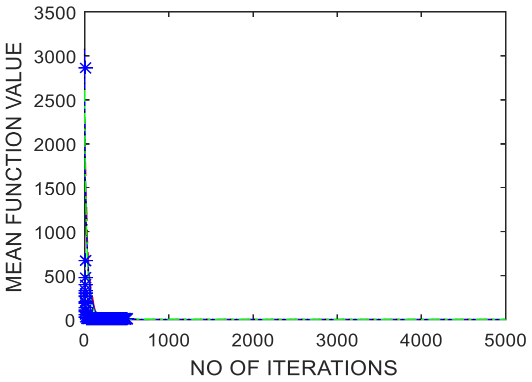
f1(const. inertia wt with 15% vel.clamping)



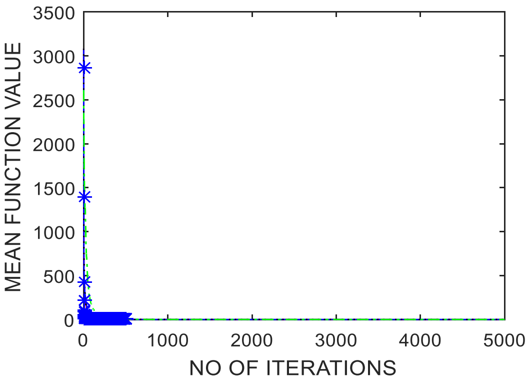
f1(zoom view) with no vel. clamping



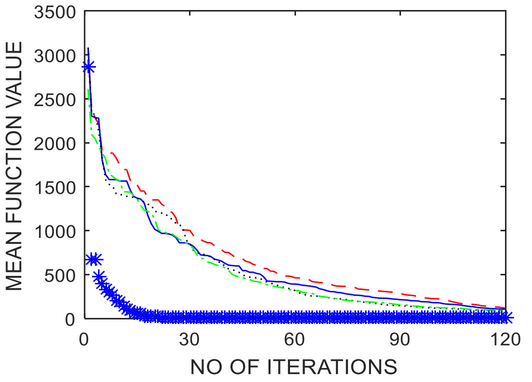
f1(zoom view) with15%vel.clamping



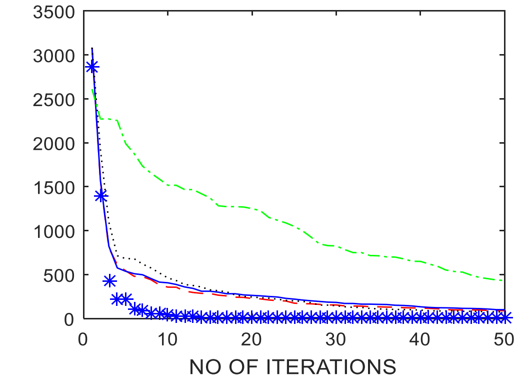
f2(const.inertia wt.with no vel.clamping)



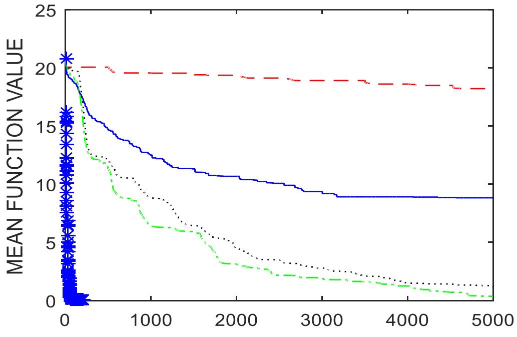
f2(const.inertia wt. with 15%vel.clamping)



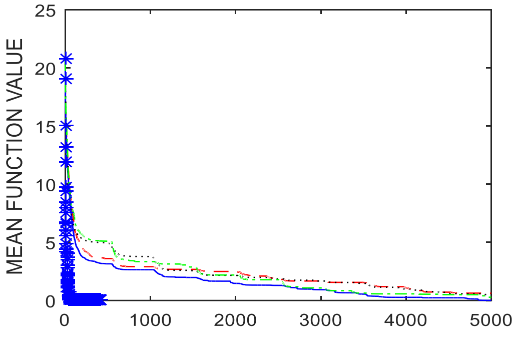
f2(zoom view)with no vel.clamping



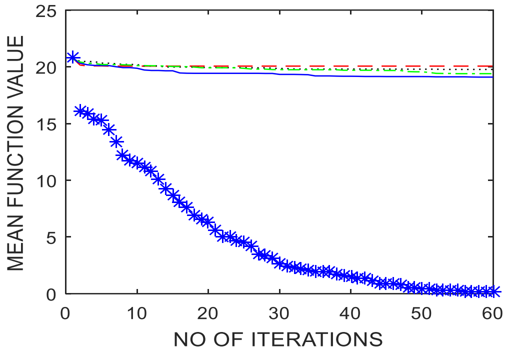
f2(zoom view)with 15%vel.clamping



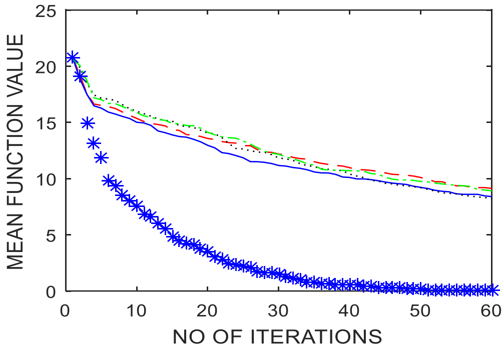
f3(const.inertia wt.with no vel.clamping)



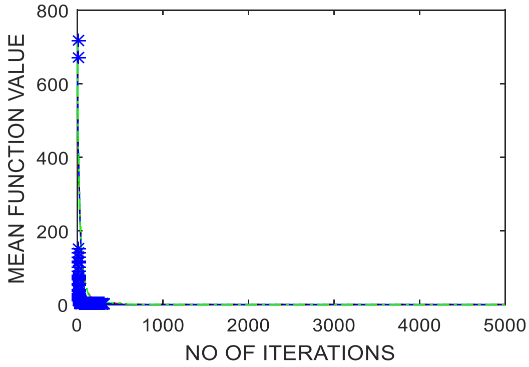
f3(const.inertia wt.with 15%vel.clamping)



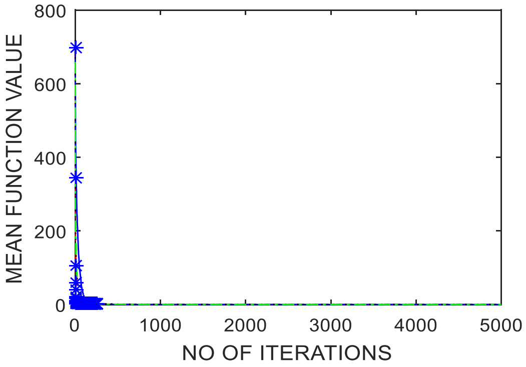
f3(zoom view)with no vel.clamping



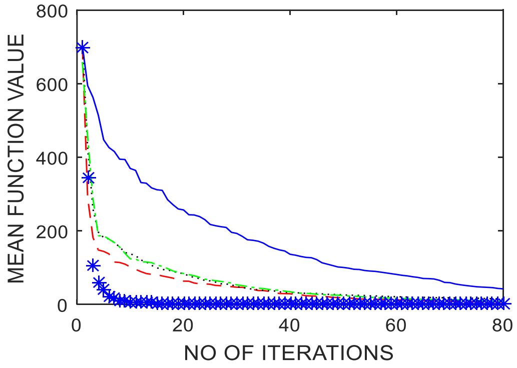
f3(zoom view)with 15%vel.clamping



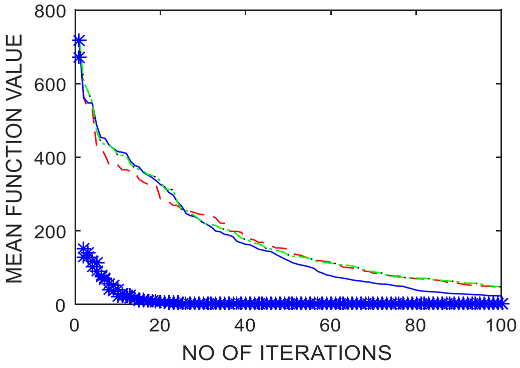
f4(const.inertia wt.with no vel.clamping)



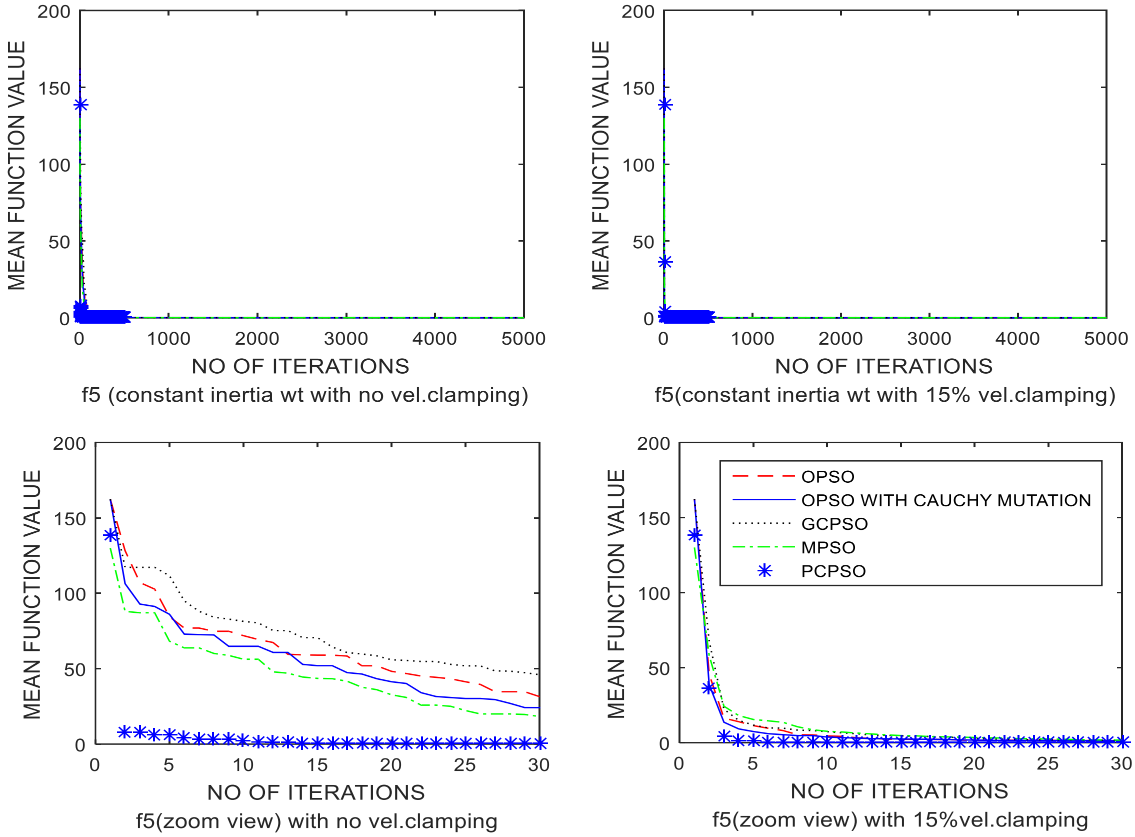
f4(const.inertia wt.with 15%vel.clamping)



f4(zoom view)with no vel.clamping



f4(zoom view)with 15%vel.clamping



f5 (constant inertia wt with no vel.clamping)

f5(constant inertia wt with 15% vel.clamping)

f5(zoom view) with no vel.clamping

f5(zoom view) with 15%vel.clamping

Performance results of PCPSO are compared with Koyuncu and Ceylan(2018) with nine variants(Global PSO-w, Global PSO-cf, Local PSO-w, Local PSO-cf, UPSO,CLPSO,SG-PSO,SP-PSO, Scout PSO) of PSO,GA(Genetic Algorithm) and four variants (norm ABC, Bin ABC, Dis ABC, Bit ABC) of ABC (Artificial Bee Colony optimization on the benchmark functions. PCPSO outperformed in achieving true global minima in all the variants of PSO and ABC in the high dimension search space due to better convergence characteristics and avoids in getting trapped in local minima due to premature convergence and stagnation.

3.6 Conclusion

A new algorithm Perfectly Convergent Particle Swarm Optimization has been developed. It has been implemented on uni-modal, multi-modal with local minima and noisy function. The proposed algorithm has been compared with the OPSO, OPSO with Cauchy mutation, GCPSO and MPSO existing variants of PSO for linearly decreasing inertia weight and constant inertia weight with and without velocity clamping. Five trials were conducted for each case and were compared statistically in terms of mean, median, best and worse .It is observed that PCPSO achieved the global minima in much less number of iterations as compared to other algorithms by a rather large margin. The problem behind the entire algorithm gave an approach to solve plagued PSO from premature convergence for overcoming the stagnation in multi-modal with few minima and many minima and the noisy function. PCPSO can also be used to solve multi-objective optimization problems. PCPSO consistently outperformed with comparison algorithm with low iterations and approximate the true global minimize. This algorithm helps and gives chance to all particles which face premature convergence and stagnation by automatically triggering a new particle mechanism .PCPSO gives a penalty if the best particle changes its position each time for$\rho(k)$to readjust and stabilize. Velocity clamping restricted the particles in initial explosion by taking small step size and selecting the values of inertia weight with acceleration co-efficient results in smooth trajectories movement in high dimension search spaces. PCPSO results are quite impressive in the case of Griewenk where all the statistical values like mean, median, best and worse are zero, irrespective of the type of inertia weight. The additional particle helps the PCPSO to solve the problem fast with large diversity. PCPSO was able to give balanced exploration and exploitation in the search space by helping particles in premature convergence states and opportunity to all particles to reach true global minima.

CHAPTER 4

MACHINE LEARNING THROUGH BACK PROPAGATION NETWORKS USING PCPSO IN HIGHER DIMENSIONS

4.1 Introduction

Particle swarm optimization (PSO) was firstly developed by Kennedy and Eberhart [6] in 1995 for solving optimization problems motivated by the behavior of a flock of birds, fishes or the social behavior of the group of people. It is a stochastic population[382,383] based approach that maintains a set of solutions within a search space called particles. Particles freely fly over the search space called exploration and gathers information like local best (Pbest) and global best (Gbest) before converging to optimum point [385,393]. PSO has a long journey with changes and have many variants. Shi and Eberhart outlined the selection criterion for inertia weights &velocity and proposed empirical study[394,14]of PSO with linearly decreasing inertia weight from 0.9 to 0.4, keeping both constants c1 & c2 equal to 2 with asymmetric initial range. PSO has common problem of stagnation because of premature convergence, especially in multi-modal function [25]. The premature convergence behavior in PSO is a major problem and is studied by many researchers, however the particles tend to converge before true global minimum. Van den Berg suggested GCPSO (Guaranteed Convergence PSO) for guaranteed convergence to local minimum[389]. Wang et al. used the opposition based learning approach along with Cauchy mutation on the best particle to accelerate the convergence and avoid to get trapped in local minima[390] .This stagnation problem was overcooked by regrouping mechanism when premature convergence is detected[142].

Kennedy and Eberhart [395] in 2001 implemented the PSO to train the Artificial Neural Networks (ANN) due to its simplicity, strong learning rate and when the data relationship is unknown. It processes the information fast through its multi- layer perceptron (MLP) having three separate layers: input, output and hidden layers. Each neuron of the layer is connected to all the neurons of next layer. Performance of training level dependent on the architecture which determines the number of hidden layers and neurons [396].The training is stopped once the inputs are equal to targets or reach to specified errors. Many algorithms are used for training purposes [397] Lavenberg-Marquardt Back propagation [398] and Bayesian Regularization Back Propagation algorithms [399] are the most powerful techniques used in this research. The sufficient layers of hidden layers are able to approximate to any degree of accuracy by suitable training.

4.2 **Artificial Neural Network:**

A two layered feed forward network with sigmoid hidden neurons and linear output neurons can fit multi-dimensional mapping problems easily. Supervised learning finds the correct weights that minimize the mapping error. The data sets used to train the network contains the input vectors and their corresponding output values. The aim is to train the network with minimum error called training data set. Second data set is called test set which has both the input and output values then it are possible to make approximation in error with respect to new data. The mapping error over test set is called test set error. Data here is randomly divided into three samples as training, validation and testing samples. Training samples are presented to the network during training and the network is adjusted as per the error. Validation is used to network generalization and to stop training when generalization does not show any improvement. Testing do not affect the training but it is independent measure of network performance during and after training.

4.3 **Proposed steps to train PSO using Artificial Neural Networks:**

1) Collect data.
2) Create the network.
3) Configure the network

4) Initialize the weights and biases.

5) Train the network using back propagation algorithms.

6) Validate the network.

7) Use the network

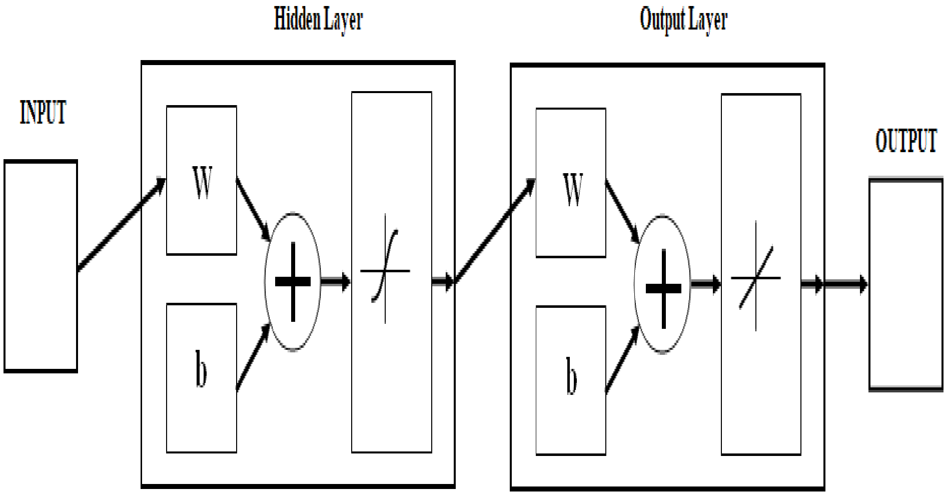


Fig 4.1 A typical artificial neural network structure.

4.4 Training of PSO variants (OPSO with Cauchy mutation & PCPSO) from following algorithms:

4.4.1 Lavenberg-Marquardt Back propagation Algorithm (LMBP):

This algorithm is an approximation to the Newton method and is more efficient used for training of moderate sized feed forward ANNs [16].The updated weights and biases are calculated as follows:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \qquad (4.1)$$

Where J is the Jacobin matrix which contains first derivatives of network errors with respect to weights and biases, x_k is the current weight and biases, e is the network errors, µ is a scalar and I is an identity matrix.

During the training, weights and biases are adjusted in such a way to minimize the network performance function that is mean square error given as:

Mean square error = $\frac{1}{N}\sum_{i=1}^{N}\left(F_i - F_i^{'}\right)^2$ (4.2)

$\frac{1}{N}\sum_{i=1}^{N}(e_i)^2$ (4.3)

Where F_i ,F_i' are the input function value and target or output values of neural network, N is the size of training data set and e_i is the error due to difference between input and output of the neural network.

4.4.2 Bayesian Regularization Back Propagation Algorithm (BRBP):

This algorithm minimizes the combination of squared errors and weights and determines a correct value for the network [17]. The performance function is given as:

$F_{reg} = \beta\sum_{i=1}^{n} e_i^2 + \alpha\sum_{i=1}^{n} w_i^2$ (4.4)

Where 'n' is the total number of weights and biases,e_i is the errors. The function controls the weights and biases to be small for good network response. Depending on the values of α and $\beta$the training errors are decided. If α<< β errors will be smaller otherwise training will pay more emphasis on weight reduction.

4.5 Experiments

4.5.1Parameter Settings:

The selection of parameters have strong impact on the performance of PSO algorithms, parameters were set as shown in table 4.1 and benchmark as taken in previous chapter table 3.1 for one set I group of experiments for OPSO with Cauchy mutation and PCPSO and uses linearly decreasing inertia weights from 0.9 to 0.4.

Table 4.1 The Parameter setting for OPSO with Cauchy mutation and PCPSO experiments

Parameter Name	Setting	Reference
Acceleration constants c1 and c2	1.49618	[8]
Inertia weights, ω	linearly decreasing from 0.9 to 0.4	[8]
Swarm size	10	
Maximum iterations	5000 per trial	
Dimensions	30	

Number of trials	05	
Stagnation threshold,ϵ	1×10^{-6}	[17]
regrouping factor =$1.2\varepsilon^{-1}$	50 for uni-modal functions	[17]
	1,20,000 for multi modal functions	[17]
Mutation Probability	0 for only OPSO	[16]
	0.5 for OPSO with Cauchy mutation	[16]

Set II group of experiments were conducted on Lavenberg-Marquardt Back propagation and Bayesian Regularization Back Propagation Algorithms using the following fixed parameters.

Table4.2 The Experimental setting for training of PSO variants using ANN Back Propagation Training Algorithms:

Parameter Name	Setting	Reference
Maximum number of Iterations	1000	
Number of Hidden Layer Neurons	4,8 and 12	
Number of Training data samples	3500 ;70% of 5000	
Number of Validation data samples	750;15% of 5000	
Number of Testing data samples	750;15% of 5000	
Number of Validation Checks	6	

4.5.2 Experimental setup:

A Intel(R) core(TM) 2 Duo CPU T7250 @ 2.00GHz was used to train network. Both training algorithms were tested on all the parameters listed in table 3.1. All the experiments were conducted for one trail consisted of maximum 1000 iterations and the stat ices were recorded as follows:

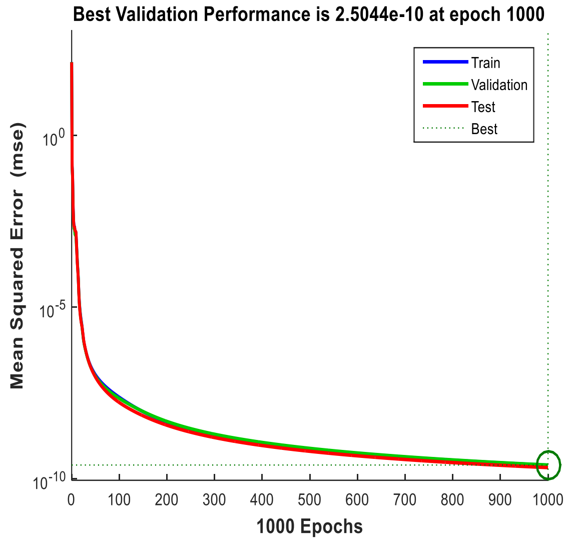
4.6 Computational results:

Training capabilities of two algorithms (Lavenberg-Marquardt and Bayesian Regularization Back propagation) are compared in the table 4.3 and table 4.4 respectively. Here the numbers of hidden layer neurons are taken as 4, 8 and 12 with training data samples of 3500, validation sample of 750 and testing sample of 750 respectively. Mean square error (MSE) and Regression (not shown in table, but its value is 9.99999e-1 for all test in table 4.3 and table4.4) is calculated and is shown table 4.3 and table 4.4. Bold values are the best values in the Table 4.3. & Table 4.4.respectively.

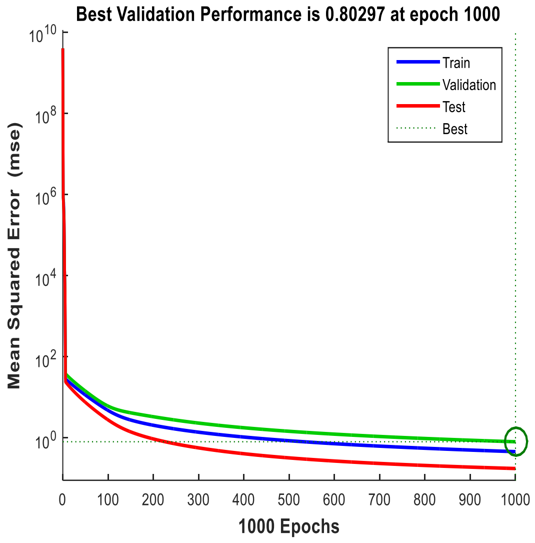
Table 4.3 Results obtained using Lavenberg-Marquardt Back propagation Algorithm

Function	**Statistics**	**OPSO with Cauchy Mutation**			**Perfectly convergenct Particle swarm Optimization (PCPSO)**		
		Mean Square Error(MSE) **Number of Hidden units**			**Mean Square Error(MSE)** **Number of Hidden units**		
		4 units	**8 units**	**12 units**	**4 units**	**8 units**	**12 units**
		MSE	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**
f_1	**Training**	4.84094e^{-8}	2.92557e^{-9}	**2.47267 e^{-10}**	1.54905 e^{-1}	**4.57828 e^{-1}**	1.75369 e^{-0}
	Validation	4.48375e^{-8}	3.60944 e^{-8}	**2.50441 e^{-10}**	1.82873 e^{-1}	**8.02968 e^{-1}**	33025.08624 e^{-6}
	Testing	4.36761e^{-8}	2.66474 e^{-8}	**2.10818 e^{-10}**	1.94060e^{-1}	**1.75773 e^{-1}**	*1.51892* e^{-0}
	Time (Sec)	126	140	**169**	136	**150**	5
f_2	**Training**	5.63367 e^{-3}	*1.85586*e^{-0}	***5.12020**e^{-5}*	*4.10265*e^{-4}	*1.66592*e^{-6}	***3.33124**e^{-8}*
	Validation	*5.65288*e^{-3}	*15.20507*e^{-0}	***6.18798**e^{-5}*	*1.20584*e^{-3}	*2.38399*e^{-6}	***3.23339**e^{-7}*
	Testing	*1.39606*e^{-0}	*28.12770*e^{-0}	***11.41788**e^{-0}*	*1.08926*e^{-3}	*1.71753*e^{-6}	***10273.4493**e^{-8}*
	Time (Sec)	130	139	**164**	129	146	**172**
f_3	**Training**	4.79715e^{-9}	2.93883e^{-9}	***3.65410**e^{-10}*	6.97988 e^{-9}	5.33756 e^{-8}	**7.90251 e^{-9}**
	Validation	5.00058e^{-9}	3.48987 e^{-9}	**2.9472e^{-7}**	8.2438 e^{-9}	1.08536 e^{-7}	**1.15509 e^{-8}**
	Testing	4.63391e^{-9}	5.37890 e^{-8}	**5.09331e^{-10}**	6.09155 e^{-9}	5.36647 e^{-8}	**7.52407 e^{-6}**
	Time (Sec)	143	160	**145**	123	133	**155**
f_4	**Training**	3.0076 e^{-6}	3.9787 e^{-7}	**6.32707e^{-8}**	1.42249e^{-4}	3.66950 e^{-5}	**1.52101 e^{-6}**
	Validation	.34462e^{-6}	5.97617e^{-7}	**1.14552 e^{-6}**	2.81389e^{-4}	3.45382 e^{-5}	**2.81896e^{-6}**
	Testing	2.68597 e^{-6}	1.86605 e^{-7}	**1.19177 e^{-6}**	1.45248e^{-4}	3.88082e^{-5}	**2.12479 e^{-6}**
	Time (Sec)	119	132	**156**	139	152	**159**
f_5	**Training**	*1.57805*e^{-6}	*2.73600*e^{-4}	***3.62640**e^{-7}*	*2.39091*e^{-5}	*4.80165*e^{-7}	***6.02854**e^{-8}*
	Validation	*2.78513*e^{-6}	*5.07811*e^{-4}	***5.54823**e^{-7}*	*1.68116*e^{-5}	*3.46217*e^{-6}	***6.80893**e^{-8}*
	Testing	*2.23884*e^{-6}	*3.25443*e^{-4}	***5.02938**e^{-1}*	*1.52752*e^{-5}	*6.17321*e^{-6}	***2.58184**e^{-8}*
	Time (Sec)	170	154	**165**	139	165	**182**

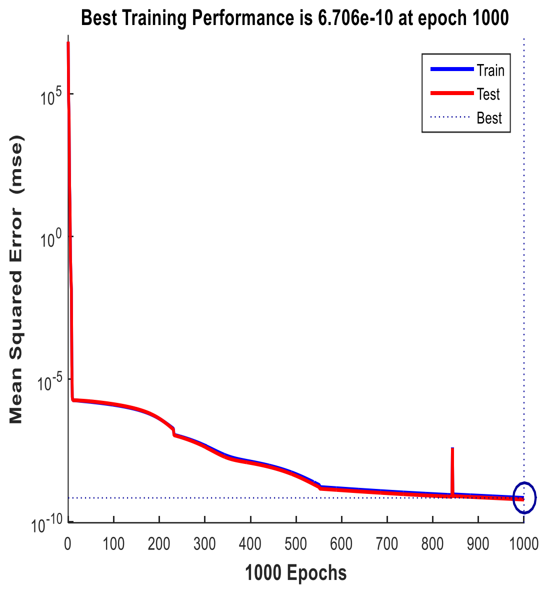
The uni-modal functions f1 with all the statically values (Training, validation, testing and time in seconds) in both the algorithms shows the time period is almost same in both cases but remarkable low error results using LMBP(with 12 neurons for OPSO with Cauchy mutation and 8 neurons for PCPSO) whereas BRBP shows (with 8 neurons for OPSO with Cauchy mutation and 12 neurons for PCPSO).The learning ability for function f2 again shows LMBP MSE goes on decreasing as the neurons increases to 12 but BRBP learning rate is good but at the cost of time. The functions f3 and f4 are multi modal in nature with many minima BRBP showed very less time period with low MSE (for f3 with 12 neurons for OPSO with Cauchy mutation and 4 neurons for PCPSO) as compared to LMBP which uses (for f3 with 12 neurons each for OPSO with Cauchy mutation PCPSO) and same happened to function f4 (with 8 neurons each for OPSO with Cauchy mutation and PCPSO).So, BRBP had shown excellent learning rate with lesser number of neurons .Function f5 being noisy in nature showed promising results by using BRBP (with 12 neurons each for OPSO with Cauchy mutation and 4 neurons PCPSO)



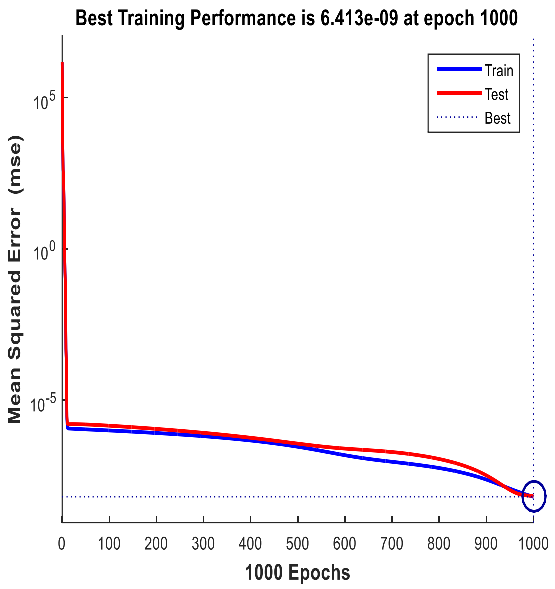
f1



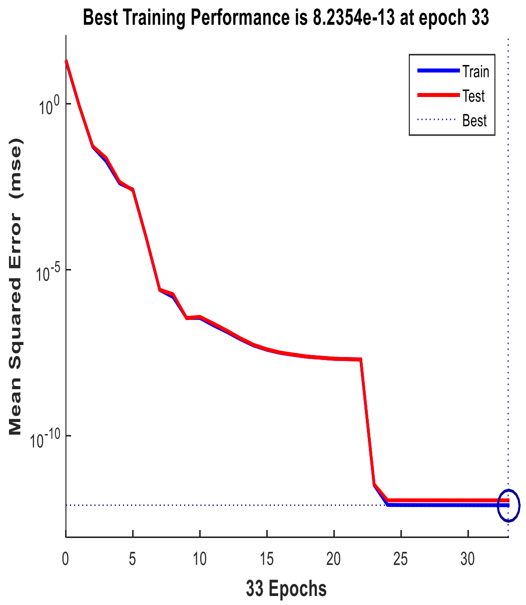
f1


Best Training Performance is 6.706e-10 at epoch 1000
Mean Squared Error (mse)
10^{5}
10^{0}
10^{-5}
10^{-10}
0
100
200
300
400
500
600
700
800
900
1000
1000 Epochs
Train
Test
Best

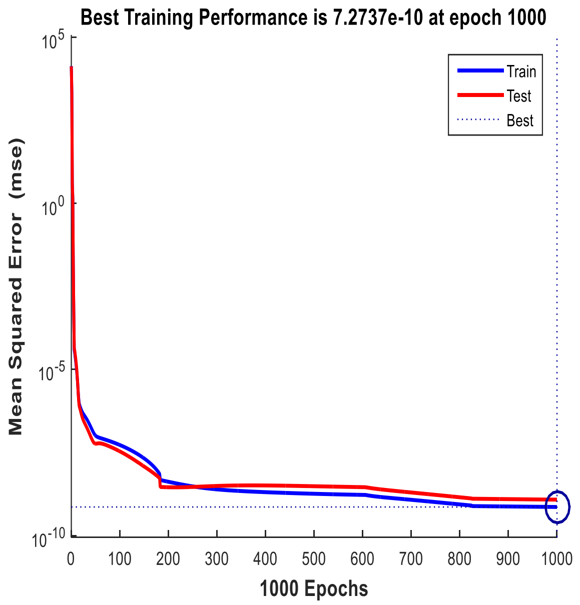
f2


Best Training Performance is 6.413e-09 at epoch 1000
Mean Squared Error (mse)
10^{5}
10^{0}
10^{-5}
0
100
200
300
400
500
600
700
800
900
1000
1000 Epochs
Train
Test
Best

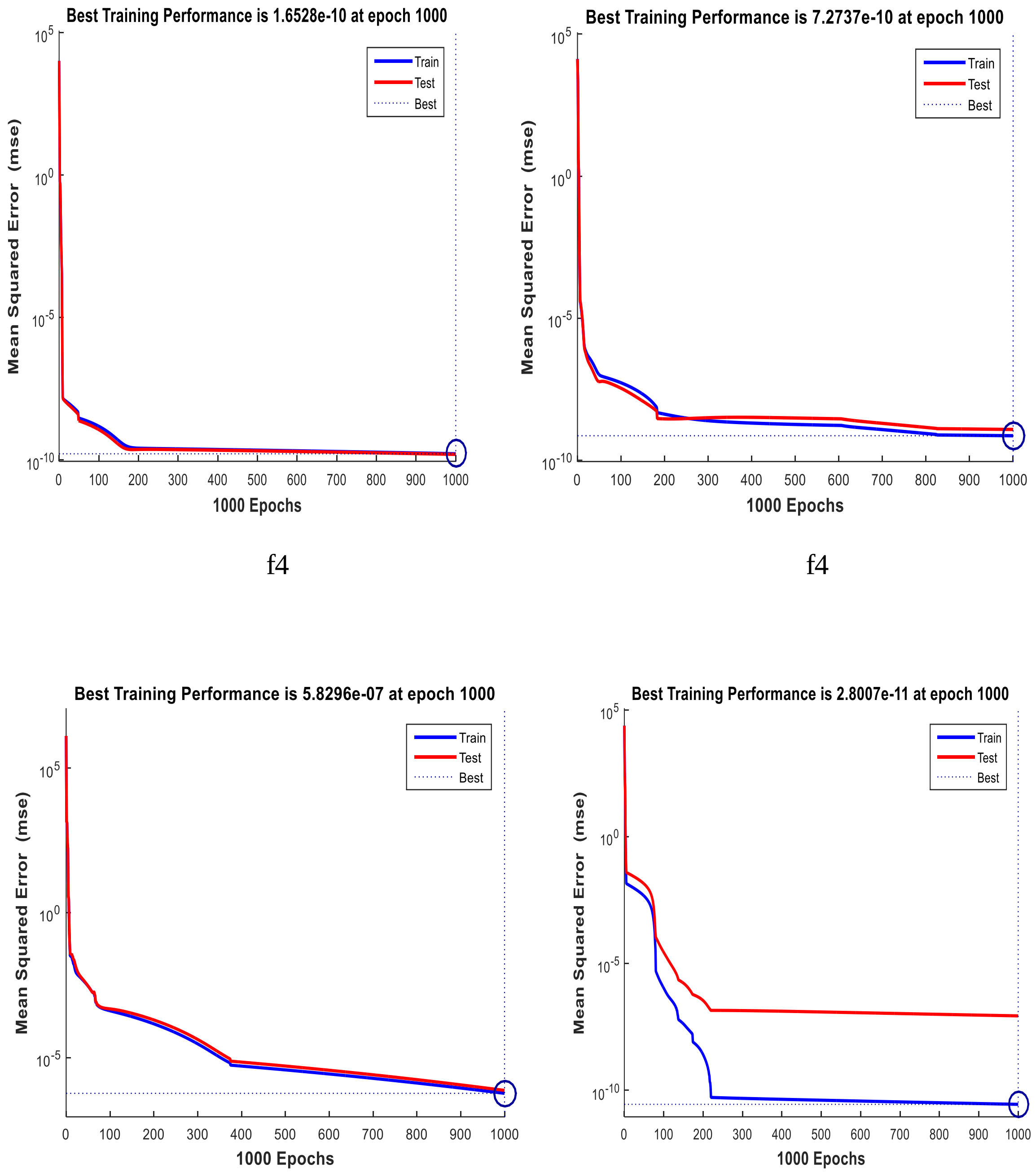
f2


Best Training Performance is 8.2354e-13 at epoch 33
Mean Squared Error (mse)
10^{0}
10^{-5}
10^{-10}
0
5
10
15
20
25
30
33 Epochs
Train
Test
Best

f3


Best Training Performance is 7.2737e-10 at epoch 1000
Mean Squared Error (mse)
10^{5}
10^{0}
10^{-5}
10^{-10}
0
100
200
300
400
500
600
700
800
900
1000
1000 Epochs
Train
Test
Best

f3



f4

f4

f5

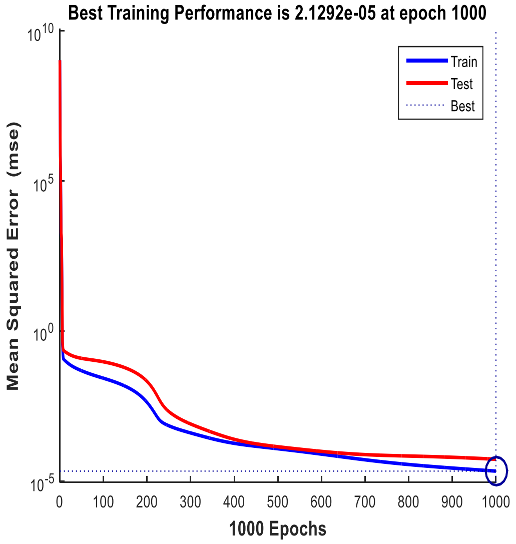
f5

Fig4.2 Comparison of Mean Square Error v/s Number of Iterations/Epochs for OPSO with Cauchy mutation & PCPSO using Lavenberg-Marquardt Back propagation for benchmark functions listed in table 4.4

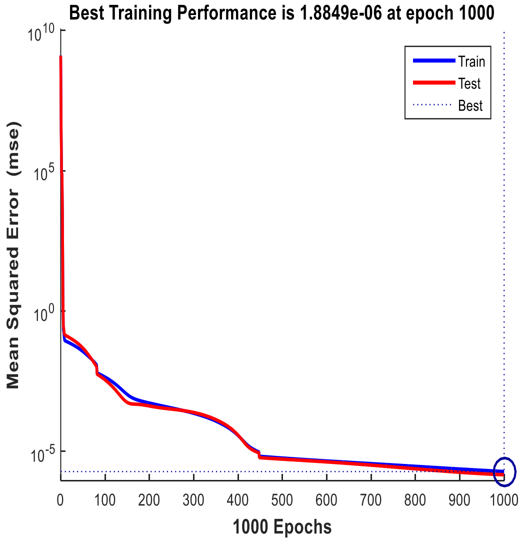
Table4.4. Results obtained using Bayesian Regularization Back Propagation Algorithm

Function	Statistics	OPSO with Cauchy Mutation			Perfectly convergent Particle swarm Optimization (PCPSO)		
		Mean Square Error(MSE) Number of Hidden units			Mean Square Error(MSE) Number of Hidden units		
		4 units	8 units	12 units	4 units	8 units	12 units
		MSE	MSE	MSE	MSE	MSE	MSE

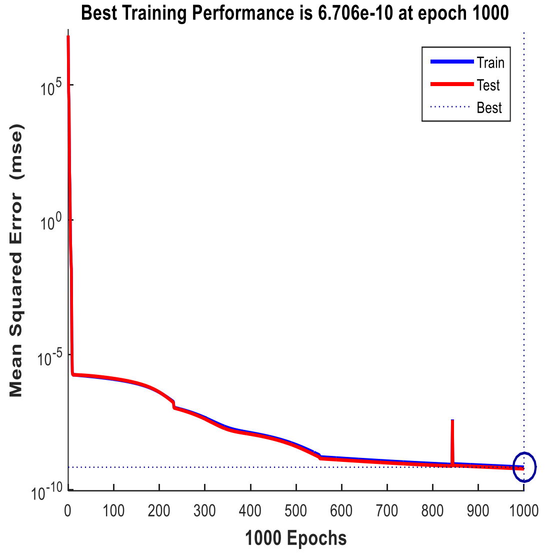
f_1	**Training**	1.93397e^{-4}	**2.12916e^{-5}**	1.47216e^{-5}	3.33201e^{-5}	1.44089 e^{-3}	**1.88491 e^{-6}**
	Validation	0.00000e^{-0}	**0.00000e^{-0}**	0.00000e^{-0}	0.00000e^{-0}	0.00000e^{-0}	**0.00000e^{-0}**
	Testing	1.46312e^{-4}	**5.28909e^{-8}**	1.29301e^{-5}	1.73145e^{-5}	2.05511 e^{-3}	***1.425256e^{-6}***
	Time (Sec)	126	**150**	168	124	144	**166**
f_2	**Training**	*1.90360e^{-7}*	*1.77918e^{-6}*	***6.70603e^{-10}***	*4.49739e^{-5}*	*3.37153e^{-9}*	***6.41304e^{-9}***
	Validation	0.00000e^{-0}	0.00000e^{-0}	**0.00000e^{-0}**	0.00000e^{-0}	0.00000e^{-0}	**0.00000e^{-0}**
	Testing	*1.51232e^{-3}*	*9.19846e^{-7}*	***5.87142e^{-7}***	*7.58700e^{-5}*	*3.52374e^{-9}*	***6.98132e^{-9}***
	Time (Sec)	162	161	**176**	177	209	**247**
f_3	**Training**	1.52452e^{-10}	2.59183e^{-12}	***8.23540e^{-13}***	**7.95311 e^{-12}**	1.53066 e^{-10}	7.99911 e^{-11}
	Validation	0.00000e^{-0}	0.00000e^{-0}	**0.00000e^{-0}**	**0.00000e^{-0}**	0.00000e^{-0}	0.00000e^{-0}
	Testing	1.43423e^{-10}	2.42614e^{-12}	**1.15359 e^{-12}**	**2.59923 e^{-11}**	1.49225 e^{-10}	1.10975 e^{-10}
	Time (Sec)	123	19	**5**	**44**	149	164
f_4	**Training**	4.50967e^{-9}	**1.65281e^{-10}**	1.45932e^{-10}	5.26674e^{-8}	**7.27369 e^{-10}**	4.84695 e^{-10}
	Validation	0.00000e^{-0}	**0.00000e^{-0}**	0.00000e^{-0}	0.00000e^{-0}	**0.00000e^{-0}**	0.00000e^{-0}
	Testing	8.84201e^{-9}	**1.57775 e^{-10}**	2.75283 e^{-11}	5.17896e^{-8}	**1.20945e^{-9}**	2.96973 e^{-2}
	Time (Sec)	128	**132**	162	133	**152**	175
f_5	**Training**	*5.03515e^{-7}*	*2.97393e^{-7}*	***5.82964e^{-7}***	***2.80067e^{-11}***	*1.01210e^{-11}*	*2.63593e^{-11}*
	Validation	0.00000e^{-0}	0.00000e^{-0}	**0.00000e^{-0}**	**0.00000e^{-0}**	0.00000e^{-0}	0.00000e^{-0}
	Testing	*8.26916e^{-7}*	*1.06588e^{-7}*	***7.36863e^{-7}***	***8.61405e^{-8}***	*1.02099e^{-10}*	*9.21393e^{-11}*
	Time (Sec)	142	163	**181**	**129**	162	174



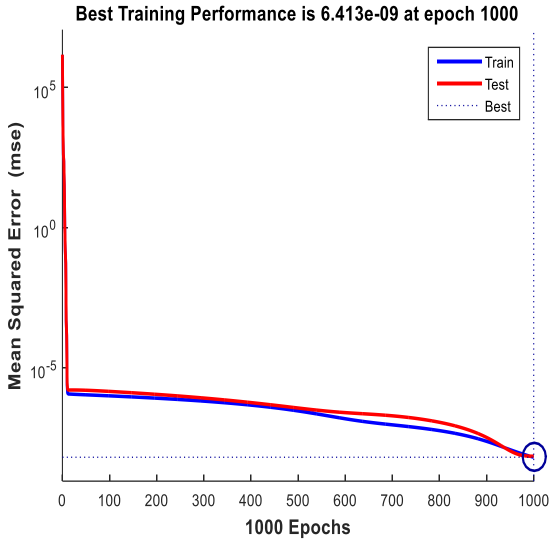
f1



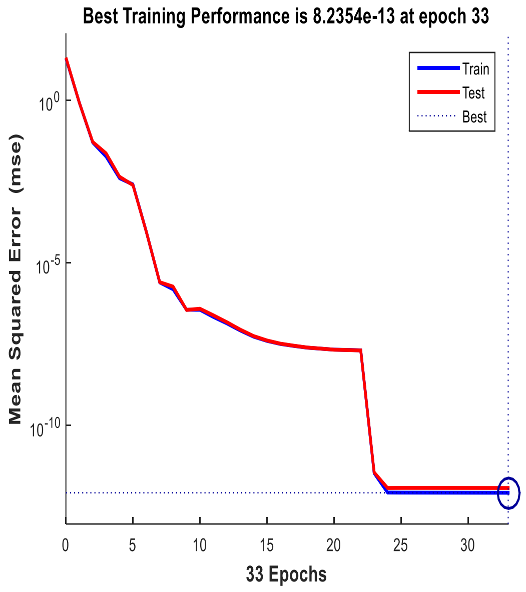
f1


Best Training Performance is 6.706e-10 at epoch 1000
Train
Test
Best
Mean Squared Error (mse)
10^{5}
10^{0}
10^{-5}
10^{-10}
0
100
200
300
400
500
600
700
800
900
1000
1000 Epochs

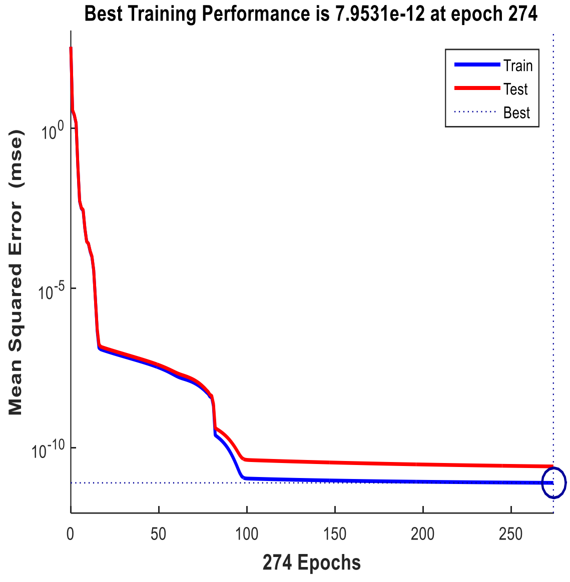
f2


Best Training Performance is 6.413e-09 at epoch 1000
Train
Test
Best
Mean Squared Error (mse)
10^{5}
10^{0}
10^{-5}
0
100
200
300
400
500
600
700
800
900
1000
1000 Epochs

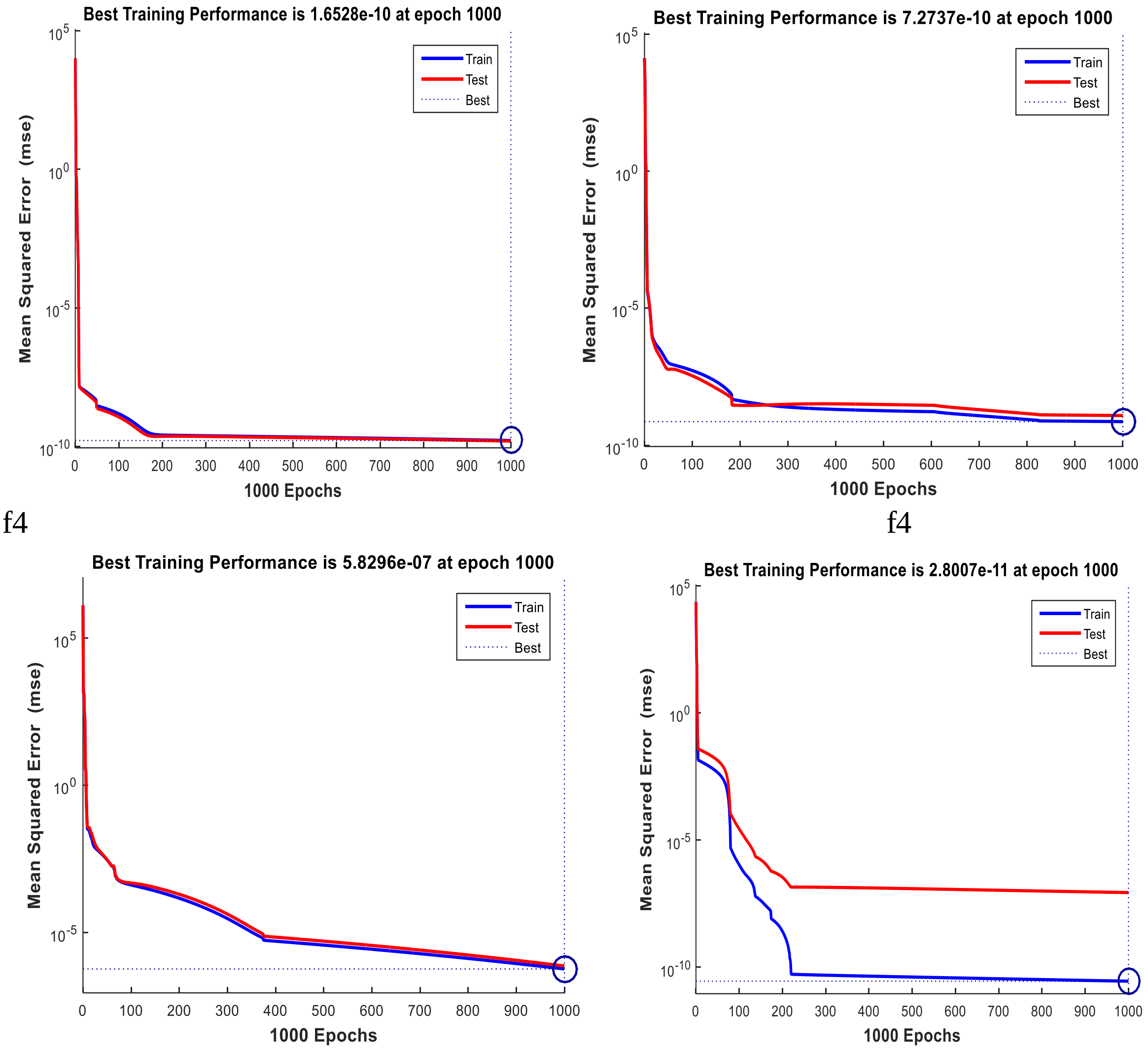
f2


Best Training Performance is 8.2354e-13 at epoch 33
Train
Test
Best
Mean Squared Error (mse)
10^{0}
10^{-5}
10^{-10}
0
5
10
15
20
25
30
33 Epochs

f3


Best Training Performance is 7.9531e-12 at epoch 274
Train
Test
Best
Mean Squared Error (mse)
10^{0}
10^{-5}
10^{-10}
0
50
100
150
200
250
274 Epochs

f3



f4 f4

f5 f5

Fig.4.3.Comparision of Mean Square Error v/s Number of Iterations/Epochs for OPSO with Cauchy mutation & GCPSO using Bayesian Regularization Back propagation Algorithm for benchmark functions listed in table 4.4

4.7 **Conclusions**

Detailed experiments were conducted to train the PSO variants (OPSO with Cauchy mutation & PCPSO) using Back Propagation algorithms (Lavenberg-Marquardt and Bayesian Regularization) on the bench mark functions in 30 dimensions. By changing number of neurons in hidden layer from 4,8,12 learning by Lavenberg-Marquardt algorithm showed remarkable accuracy to train. ANN is able to train uni-modal, multi modal with many minima as well as noisy functions with very low (near to zero) using PSO. The simulation results shows that Lavenberg-Marquardt algorithm is more efficient for uni-modal functions and Bayesian Regularization out performs for multi modal /noisy functions. The training of ANN using PCPSO is very fast and has efficient learning rate when high precision is required.

CHAPTER 5

COMBINED ECONOMIC EMISSION DISPATCH WITH QUADRATIC FUNCTION WITH POINT VALVE LOADING USING PERFECTLY CONVERGENT PARTICLE SWARM OPTIMIZATION

5.1 Introduction

Fossil fuels are one of the most major part of power generation, make up a majority of global generation. Sulphur-di-oxide, nitrogen dioxide, carbon dioxide, ozone, and other hazardous gases and particles are released into the air, contributing to global warming. The final Affordable Clean Energy regulation (ACE), which repealed the Clean Power Plan [400] for generating units, was published by the Environmental Protection Agency (EPA) in June 2019. As a result, an innovative technique is designed to decrease emissions from thermal power stations.

Numerous approaches have been proposed and introduced to overcome the power system's economic dilemma. Linear programming, Lagrangian relaxation, and the Lagrange multiplier are some of the early methodologies that have been used. To enhance existing techniques, such as the genetic algorithm (GA), evolutionary programming(EP), particle swarm optimization(PSO), Biogeography Based Optimization(BBO) , harvest season artificial bee colony, differential evolution(DE), Backtracking search algorithm(BSA), Gravitational search algorithm(GSA), epsilon-multi-objective genetic algorithm variable(ev-MOGA), Flower pollination algorithm(FPA), quasi oppositional teaching learning based optimization(QOTLBO), modified artificial bee colony algorithm (MABC/D/Cat , MABC/D/Log) , Kernel search Optimization (KSO) and more alternative generations composed by intelligent techniques have been developed. In comparison to other options, very few of the heuristic search algorithms has yet to be able to provide adequately high performance to resolve all optimization problems. As a result,creating a population-based heuristic search technique capable of preventing premature convergence while maintaining the rapid converging feature remains a difficulty. The use of this technique yielded good optimal results in a short amount of time.

5.2 Formulation of Combined Economic Emission Dispatch problem

The mathematical formulation of the CEED problem is presented in this section, which includes the quadratic fuel cost function model, quadratic emission model, and max-max price penalty function.

5.2.1 Quadratic fuel cost function

As the initial objective of the committed generating units, coupled with equality and inequality requirements, the significant portion of the operating cost of thermal power plants is described as a second order of quadratic function:

$$Min\, F_{CT} = \sum_{i=1}^{n} a_i P_i^2 + b_i P_i + c_i + \left| \alpha_i \sin\left(\beta_i\left(P_{i,min} - P_i\right)\right)\right| \frac{\$}{h} \quad (5.1)$$

Subject to constraints:

Power balance constraint: The total real power generation is equal to the sum of total power demand and transmission losses.

$$\sum_{i=1}^{n} P_i = P_D + P_L \quad (5.2)$$

Generator limit constraint: The real power generation of i_{th} committed generating unit should be within following limit.

$$P_{i\,min} \ll P_i \ll P_{i\,max} \quad (5.3)$$

Transmission loss constraint: The total transmission loss $P_L$should be minimum and is given as George’s formula:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j \quad (5.4)$$

Where $F_{CT}$is the fuel cost of all generators in \$/h,P_i is the real output power in MW of i_{th} generator,P_D , P_L are total demand and transmission losses in MW,$P_{i\,min}$, $P_{i\,max}$are the minimum and maximum power limits of i_{th} generator, n is the number of committed generating units,a_i, b_i, c_i, d_i are the fuel cost curve co-efficient of the i_{th} generators respectively.$B_{ij}$is the matrix of transmission loss coefficient of generating units.

5.2.2 Quadratic Emission function

Due to the burning of fossil fuels, all thermal power plants create hazardous gases such as SO2, NOx, and CO2, which add to the overall emissions and must be reduced individually. All three emissions are mathematically defined in this model using quadratic polynomials as follows:

$$E_T = \sum_{i=1}^{n}(d_i P_i^2 + e_i P_i + f_i) + \gamma_i exp(\delta_i P_i) \ \ Kg/h \qquad (5.5)$$

Whereas E_T is the total emission with valve loading effect in ton/h, d_i, e_{i,f_i} are coefficients of emission of i^{th}generating unit in ton/M$W^2h,$ton/MWh and ton/h , $\gamma_i \ and \ \delta_i$ are the valve point loading effect emission coefficient of i^{th}generating unit.

5.2.3 Price Penalty Factors (PPF)

Price penalty factors are calculated by dividing the fuel cost by the emission value and are used to transform emission criteria into equivalent fuel costs. The Max-Max price penalty factor, h_i employed is as follows.

$$h_i = \frac{(a_i P_{i\,max}^2 + b_i P_{i\,max} + c_i) + \left|\alpha_i \sin\left(\beta_i(P_{i,min} - P_i)\right)\right|}{(a_i P_{i\,max}^2 + b_i P_{i\,max} + c_i) + \gamma_i exp(\delta_i P_i)} \qquad (5.6)$$

5.2.4 Bi-objective CEED

The bi-objective CEED equations are shown below, which incorporate fuel cost with each emission and are then converted to a single objective by multiplying a price penalty factor to emissions independently.

$$F_T = \sum_{i=1}^{n}[(a_i P_i^2 + b_i P_i + c_i) + \left|\alpha_i \sin\left(\beta_i(P_{i,min} - P_i)\right)\right|$$

$$+ h_{i\,SO2}\left(a_i\,P_i^2 + b_i\ P_i + c_i\ + \gamma_i exp(\delta_i P_i)\right)]\frac{\$}{h} \qquad (5.7)$$

5.3 Particle Swarm Optimization

Kennedy and Eberhart [6] established PSO without inertia weight in 1995, but for the first time in 1998, they developed it with constant inertia weight, and this method became known as conventional PSO. Initialized with candidate solutions of particles moving through the search space, each particle having a position and velocity, and updates as follows:

$$x_j(k+1) = x_j(k) + v_j(k+1) \quad (5.8)$$

$$v_j(k+1) = \omega v_j(k) + c_1(K)r_1\big(p_j(k) - x_j(k)\big) + c_2(K)r_2\big(g(k) - x_j(k)\big) \quad (5.9)$$

Where, j =1, 2, 3 … n

$$\omega(K) = \omega_{max} - k \times (\omega_{max} - \omega_{min}) \div K_{max} \quad (5.10)$$

$$C_1(K) = 1.167 \times \omega(K)2 - 1.167 \times \omega(K) + 0.66 \quad (5.11)$$

$$C_2(K) = 3 - C_1(K) \quad (5.12)$$

$C_1(K), C_2(K)$are asynchronous learning factors with non-linear dynamic adjustable features which have high chances of converging to optimal global solution.

k+1 denotes next iteration, k is the current iteration number, v_j is velocity of the particle j,x_j is position of the particle j, ω is Inertia weight factor,c_1,c_2 are acceleration factors,$p_j$is personal best of particle j, g is the global best of the entire swarm,r_1,r_2 are pseudo random numbers between 0 and 1.$\omega_{max}, \omega_{min}$ are having maximum and minimum value of 0.9 and 0.4 of inertia weights

Proposed algorithm as Perfectly Convergent Particle Swarm Optimization (PCPSO)

The purpose of this proposed version [401-403] in our scenario is to eliminate premature convergence, which contributes to stagnation, and to allow personal best particles to replace global particles because they provide more search space exploration. I've introduced an additional particle in this new form, similar to the one used in GCPSO [404], but instead of searching for global position, it will hunt for personal best position. Searching areas close to global position, while taking into consideration the current velocity update, lacks exploration and risks entrapment in multi-modal situations with one or more local minima:

$$v_j'(k+1) = -x_j{}'(k) + pbest(k) + \omega v_j'(k) + \ \rho(k)(1-2r) \quad (5.13)$$

Other particles in the swarm, on the other hand, will adjust their velocity according to this new variant:

$$v_j'(k+1) = \omega(k)x_j{}'(k) + c_1r_1\big(p_j(k) - x_j(k)\big) + c_2r_2\big(-x_j(k)\big) \quad (5.14)$$

$$\omega(k) = \omega_{max} - k \times (\omega_{max} - \omega_{min}) \div K_{max}$$

$$C_1(k) = 1.167 \times \omega(k)2 - 1.167 \times \omega(k) + 0.66$$

$$C_2(k) = 3 - C_1(k)$$

$C_1(k), C_2(k)$ are asynchronous learning factors with non-linear dynamic adjustable features which have high chances of converging to optimal global solution.

Where, $-x_j{}'(k) + pbest(k)$ component will make the search in the personal best region, $\omega v_j'(k)$ gives the momentum to search in current direction, ρ(k) (1-2r) generates a random search in the neighborhood area of personal best particle with side length of $2\rho(k)$, ρ(k) is the diameter of random search space defined as follows:

$$\rho(k+1) = \begin{cases} 2\rho(k) & Successes > sc \\ (0.5)\rho(k) & failure > fc \\ \rho(k) & otherwise \end{cases} \qquad (5.15)$$

#successes (k+1)> #successes (k) => # failures (k+1) =0 and, # failures (k+1)> # failures (k) => #successes (k+1) =0

Where, #successes & # failures are the number of consecutive successes or failures, s_c= 15 & f_c=5 are the threshold parameters and can be finely adjusted.

In its current version, this method employs an adaptive for determining the best sampling volume. If a certain value of consistently yields a positive result, a high sampling volume is used to increase the maximum distance walked in a single step. When, on the other hand, yields multiple failures in a row, the sampling volume is too large and must be lowered. At the conclusion of the day, if > 0 for all stages, there will be no standstill.

This variant, in essence, enables all particles to compete, irrespective whether they're in the exploratory stage/have a better personal best than the previous iteration or are on the edge of global optima, resulting in a true global search algorithm. This technique gets over GCPSO's restrictions

5.4 **PCPSO execution in CEED**

Step1. Specify the lower and upper limitations for each unit's generation, as well as the area load demand and tie line transfer restrictions.

Step2. For a size of population S in the jth-dimensional space, produce particles at random between the min and max operating limits of the N units, using the $i^{th}\,particle \quad P_i = [(P_{i1}^n, P_{i2}^n, P_{i3}^n \ldots P_{iN}^n)]$where i=1, 2...S.

To meet the generation limit criteria of (5.3), in this case, r is a uniformly distributed random number between 0 and 1.

$$P_{ij}^n = P_{min} + r\left(P_{ij\,max} - P_{ij\,min}\right) \qquad (5.16)$$

Step3: Constraints imposed by prohibited operating zones

If any element P_{ij} of the starting population (or updated population) is determined to be within the kth forbidden operating zone, it is adjusted and given the generation value corresponding to the zone's (P_{ij}^{lower}) or higher (P_{ij}^{upper}) boundary, as stated by the logic. $P_{mid,k}$.is the kth restricted zone's midpoint.

$$P_{ij} = \begin{cases} P_{ij}^{lower} & if\ P_{ij}^{lower} \le P_{ij} < P_{mid,k} \\ P_{ij}^{upper} & if\ P_{mid,k} \le P_{ij} < P_{ij}^{upper} \end{cases} \qquad (5.17)$$

Step4. Set particle velocity in the $[v_i^{min} v_i^{max}]$ in N-dimensional space.

Step5. Evaluate the fitness of each individually using the equation (5.1, 5.5, 5.7).

Step6: The parameters are iteratively changed to improve fitness. The parameters of PCPSO are updated using equations (5.8, 5.9, 5.13-5.15).

Step7: The evaluation function values for the changed particle positions are found. PCPSO sets the new value to pbest if it is better than the old pbest. Similarly, gbest's value is changed to reflect its position as the best vector among pbest.

Step8. Stop criteria

The position of particles is denoted as Gbest for the optimal solution and stop if equation (17) is less than the stagnation threshold of $\varepsilon = 1x10^{-6}$

5.5 Simulation results and discussion

The CEED problem was solved using the proposed PCPSO methodology using three different test platforms. To do this, we created a software in the MATLAB 2015a platform on Compaq 6720s lab-top with 4GB RAM and tested it on three different IEEE test unit systems with six units, ten units, and forty units, including for the losses in variability transmission and other constraints. Number of particles in swarm is 20, number of iterations are 250, number of trails is 5, linearly decreasing inertia weight with maximum and minimum inertia w=0.9 to 0.4, acceleration constants c1 = c2 =2 are some of the parameters of proposed PCPSO

5.5.1 Test system 1

To highlight the effectiveness of the PCPSO approaches for addressing the CEED problem with line flow limits, the CEED problems are investigated and validated on the IEEE 30-bus, 6-generator system[405] at a demand load of 283.4 MW, also in appendix. Using the PCPSO approach including all system limitations, optimal generator scheduling was accomplished. All buses have a lower and upper voltage limit of 0.94 p.u. and 1.06 p.u, respectively, with a maximum voltage variation of 6%.The proposed PCPSO algorithm is compared with latest research papers algorithms BBO [406], GA[407], EP[407], PSO[407] and DE[407] with lowest F_{CEED} of 2004.30$/h as shown in table 5.1.The simulation results shows power loss $P_L$of 5.78MW with computational time of 6.08 seconds resulting in excellent convergence characteristics.

Table5.1Comparision of CEED results for six generating system with other techniques.

	PCPSO	BBO[406]	GA[407]	EP[407]	PSO[407]	DE[407]
P_1	120.28	127.84	58.41	114.29	107.73	121.65
P_2	48.40	42.41	76.27	50.38	46.60	56.58
P_3	30.62	31.19	47.82	30.19	27.93	36.30
P_4	31.44	33.27	33.44	32.78	35.00	28.91
P_5	29.00	29.97	28.75	29.36	30.00	22.88
P_6	29.43	29.40	39.98	30.66	40.00	21.90
F_{CEED}	**2004.30**	2084.78	2107.19	2094.39	2109.47	2122.53

5.5.2 Test system 2.

This scenario involves a thermal system with 10 units of generation and valve point effects. The fuel cost coefficients matrix, generator constraint matrix, emission coefficient matrix, and transmission loss coefficient matrix are taken from [408] and in appendix.. Table 5.2 displays the outcomes of using PCPSO to solve CEED for a 2000MW load demand and contrasts them with other approaches.

The proposed PCPSO algorithm shows the lower fuel cost with emission cost along with lowest combined economic emission dispatch of 216166.43 $/h which is lower than BSA, MODE[408], PDE[408], GSA[418], QOTLBO[412], NGPSO, FPA[410], ev-MOGA[420], ABCDP-LS[417], by 2948.85$/h, 2015.07$/h, 1701.25$/h, 1687.24$/h, 1624.31$/h, 4.11$/h, 2761.69$/h and 1848.61 $/h respectively in achieving optimal global minimum solution in low iteration and computing time. This shows the excellent performance of PCPSO without getting trapped in minima solution

5.5.3 Test system 3.

The actual Tai power system, which is a large-scale and varied generating system with coal-fired, oil-fired, gas-fired, diesel, and combined cycle all present, has 40 generating units

[408]. The ramp rate limit (RRL), banned operating zones (POZ), non-smooth cost function with valve point effects, and emission function all contribute to the system's load demand of 10500 MW given in appendix. The PCPSO's best simulation results as shown table 5.3. are compared to with MODE[408], PDE[408], MABC/D/Cat[409], MABC/D/Log[409], FPA[410], KSO[411], QOTLBO[412], GQPSO[413], SAIWPSO[414], PSOGSA[415] ,MA θ-PSO[416], HPSOGSA[415], IABC[417], GSA[418,427], NSGA-II[419] , SPEA2[408], IABC-LS[417] , TLBO[425] , ev-MOGA[420] , ABCDP[417] , MBFA[421] , DE-HS[422] , MLTBO[423] , RCCRA[424] and BPO[425] is lowest in fuel cost 12430.00 $/h along with emission cost .The F_{CEED} is 220810.00 $/h with low power loss P_L of 81.59MW which is also lower from the results in a very low computation time of 3.16 seconds in few iterations. Following table 5.2 and table 5.3 shows the comparison of fuel cost and emission cost of all the recent algorithms

Table 5.2 Shows the comparison of fuel cost, emission cost and F_{CEED} of PCPSO with the other algorithms.

	BSA[426]	MODE[408]	PDE[408]	GSA[427]	QOTLBO [412]	NGPSO[428]	FPA[410]	ev-MOGA[420]	**PCPSO**
P_1	55.00	54.9487	54.9853	54.9992	55.0000	55.00	53.188	54.1807	**55.00**
P_2	80.00	74.5821	79.3803	79.9586	80.0000	80.00	79.975	78.4981	**80.00**
P_3	86.53	79.4294	83.9842	79.4341	84.8457	81.2398	78.105	84.7653	**78.2489**
P_4	86.98	80.6875	86.5942	85.0000	83.4993	80.8334	97.119	81.3502	**81.8443**
P_5	129.15	136.89551	144.4386	142.1063	142.9210	160.00	152.740	138.0526	**157.8900**
P_6	146.92	172.6393	165.7756	166.5670	163.2711	235.0087	163.080	166.2667	**231.8700**
P_7	300.00	283.8233	283.2122	292.8749	299.8066	289.3507	258.610	295.466	**290.0735**
P_8	323.90	316.3407	312.7709	313.2387	315.4388	297.4542	302.220	326.7642	**299.2467**
P_9	435.99	448.5923	440.1135	441.1775	428.5084	401.5072	433.210	428.9338	**403.6784**
P_{10}	440.01	436.4287	432.6783	428.6306	430.5524	401.4275	466.070	429.6309	**403.5242**
F_C	112807.37	1.1348e5	1.1351e5	1.1349e5	113460	116179.6487	1.1337e5	113422.34	**113360.46**
E_C	4188.0	4124.9	4111.4	4111.4	4110.2	3939.227	4147.1	4120.52	**3910.7**

	9					8	7	04	**8**
F_{CEED}	219115.28	218181.5	217867.68	217853.24	217790.74	216170.54	218928.12	218015.04	**216166.43**

Table5.3. Fuel cost F_C and Emission cost E_C of PCPSO is compared to other latest algorithms

	PCPSO	MODE[408]	PDE[408]	MABC/D/Cat[409]	MABC/D/Log[409]	FPA[410]	KSO[411]
P_1	**114.00**	113.53	112.15	110.79	110.79	43.40	112.80
P_2	**114.00**	114.00	113.94	110.79	110.79	113.95	112.68
P_3	**108.98**	120.00	120.00	97.39	97.39	105.86	119.67
P_4	**166.32**	179.80	180.26	174.55	174.55	169.65	179.66
P_5	**97.00**	96.77	97.00	87.79	97.00	96.65	96.68
P_6	**131.44**	139.27	140.00	105.39	105.39	139.02	139.72
P_7	**286.80**	300.00	299.88	259.59	259.59	273.28	298.30
P_8	**300.00**	298.91	300.00	284.59	284.59	285.17	284.60
P_9	**300.00**	290.77	289.89	284.59	284.59	241.96	284.60
P_{10}	**168.40**	130.90	130.57	130.00	130.00	131.26	130.00
P_{11}	**215.72**	244.73	244.10	318.19	318.21	312.13	311.46
P_{12}	**213.76**	317.82	318.28	243.59	243.59	362.58	315.59
P_{13}	**291.35**	395.38	394.78	394.27	394.27	346.24	394.28
P_{14}	**320.91**	394.46	394.21	394.27	394.27	306.06	394.28
P_{15}	**320.34**	305.81	305.96	394.27	394.27	358.78	394.28
P_{16}	**320.34**	394.82	394.13	394.27	394.27	260.68	394.28
P_{17}	**465.50**	487.98	489.30	399.51	399.51	415.19	488.33
P_{18}	**467.26**	489.17	489.64	399.51	399.51	423.94	497.57
P_{190}	**509.65**	500.52	499.98	506.19	506.17	549.12	487.59
P_{20}	**509.65**	457.00	455.41	506.19	506.22	496.70	421.52
P_{21}	**550.00**	434.60	435.28	514.14	514.11	539.17	433.54
P_{22}	**550.00**	434.53	433.73	514.14	514.14	546.46	433.54
P_{23}	**550.00**	444.67	446.24	514.52	514.56	540.06	433.62
P_{24}	**550.00**	452.03	451.88	514.53	514.48	514.50	433.57
P_{25}	**550.00**	492.78	493.22	433.51	433.51	453.46	433.52
P_{26}	**550.00**	436.33	434.74	433.51	433.51	517.31	433.52
P_{27}	**14.03**	10.00	11.80	10.00	10.00	14.88	10.00
P_{28}	**14.03**	10.39	10.75	10.00	10.00	18.79	10.00
P_{29}	**14.03**	12.31	10.30	10.00	10.00	26.61	10.00
P_{30}	**97.00**	96.90	97.00	97.00	87.80	59.58	97.00
P_{31}	**176.59**	189.77	190.00	159.73	159.73	183.48	187.91
P_{32}	**176.59**	174.23	175.30	159.73	159.73	183.39	186.12
P_{33}	**176.59**	190.00	190.00	159.73	159.73	189.02	188.51
P_{34}	**90.00**	199.65	200.00	200.00	200.00	198.73	199.72
P_{35}	**90.00**	199.86	200.00	200.00	200.00	198.77	200.00
P_{36}	**90.00**	200.00	200.00	200.00	200.00	182.23	200.00
P_{37}	**110.00**	110.00	109.94	89.11	89.11	39.67	110.00
P_{38}	**110.00**	109.94	109.88	89.11	89.11	81.59	110.00
P_{39}	**110.00**	108.17	108.96	89.11	89.11	42.96	110.00
P_{40}	**509.65**	422.06	421.37	506.18	506.19	537.17	421.52
F_C	**12430.00**	12579.00	12573.00	12490.90	124491	123170.00	125491.00
E_C	**76610.00**	211190.00	211770.	256560.	256560.0	208460.00	199591.00

			00	67	0		
F_{CEED}	**220810.00**	NA	NA	NA	NA	NA	NA
Time	**3.16**	5.39	6.15	NA	NA	4.92	NA

Table5.4 Fuel cost F_C and Emission cost E_C of PCPSO is compared to other latest algorithms.

Algorithm	Fuel Cost F_C	Emission Cost E_C
PCPSO	**12430.00**	**76610.00**
QOTLBO[412]	125161.00	206490.00
GQPSO[413]	146121.50	270192.37
SAIWPSO[414]	121676.23	177276.36
PSOGSA[415]	129987.00	176678.00
MA θ-PSO[416]	129995.00	176682.00
HPSOGSA[415]	129997.00	176684.00
IABC[417]	129995.00	176682.00
GSA[418]	125782.00	210933.00
NSGA-II[419]	125825.00	210949.00
SPEA2[408]	125808.00	211098.00
IABC-LS[417]	12995.00	176682.00
TLBO[425]	125602.00	206648.00
ABCDP[417]	129995.00	176682.00
ABCDP-LS[417]	129995.00	176682.00
MBFA[421]	129995.00	176682.00
DE-HS[422]	129994.00	176682.00
MLTBO[423]	127283.87	99127.70
RCCRA[424]	124250.95	229395.90
BPO[425]	127335.40	97848.41

5.6 Conclusion

In order to solve CEED issues in power systems, PCPSO has been created in this study. The effectiveness of the PCPSO was evaluated for a number of test cases and contrasted with the recent research papers. It is confirmed that PCPSO is preferable an alternative algorithms for solving CEED issues, especially in large-scale power systems with valve point impact. Additionally, PCPSO shows avoiding to struck in the premature convergence in local minima which results in better economic and emission impact, computational effectiveness, and its convergence feature. As a result, PCPSO optimization is a viable method for addressing challenging issues in power systems. The future scope of this work includes applications of the proposed algorithm to multi-area power systems integrated with wind farms and PV systems. The PCPSO outperforms in effectively resolving bi- and multi-objective power system optimization issues with optimal outcomes in a minimal amount of compute time and iteration.

CHAPTER 6

COMBINED ECONOMIC EMISSION DISPATCH USING PERFECTLY CONVERGENT PARTICLE SWARM OPTIMIZATION

6.1 Introduction

One of the most important elements of power generation is fossil fuels and mainly accounts for global generation. These thermal plants release harmful gasses like Sulphur dioxide, nitrogen dioxide, carbon-di-oxide, ozone etc. and particles into the atmosphere causing global warming. In June2019, [400] Environment Protection Agency (EPA) published the final Affordable Clean Energy regulation (ACE) which abolished the Clean Power Plan for generating units. As a result it requires an intelligent technique to minimize the emissions from the thermal power plants.

Multi objective combined economic emission power dispatch (CEED) optimization problem are broadly classified into deterministic and meta-heuristic methods. Deterministic methods such as Newton-Raphson, Langrage relaxation methods were fast and can easily consider emissions constraints also whereas meta-heuristic techniques like Genetic algorithm (GA),firefly algorithm (FA),simulated annealing(SA),Particle swarm optimization (PSO),Quantum PSO (QPSO), Grasshopper optimization algorithm(GOA),Pattern search (PS),Biogeography based optimization, artificial bee colony(ABC), Gravitational search algorithm(GSA),differential evolution (DE),non- dominated sorting genetic algorithm(NSGA-II) and various hybrid models of heuristic techniques were used to solve CEED optimization problems. Literature survey of recent and previous cubic CEED research papers were conducted using Langrage relaxation [428], PSO [429], SA [430,431], Modified Bio-geography based optimization (MBO)[432],GOA algorithm [433], Artificial ecosystem based optimization (AEO)[434],Multi objective 4th chaotic function Artificial ecosystem based optimization (CAE04)[435],Quantum Particle swarm optimization (QPSO)[436],Sine-cosine algorithm(SCA)[437].

Cubic cost and Cubic emission functions [438-443] are considered which represent the correct operating cost of generating units as compared to quadratic functions using PCPSO. This adoption of this technique resulted in excellent optimal solutions with low computational time.

6.2 Combined economic emission Problem formulation

This section presents the mathematical formulation of CEED problem including the cubic fuel cost function model, cubic multiple emission models and six price penalty functions.

6.2.1 Cubic fuel cost function model

The major portion of the operating cost of thermal power plants is described as a third order of cubic function as the first objective of the committed generating units along with equality and inequality constraints as follows:

$$Min\ F_{CT} = \sum_{i=1}^{n} F_i P_i \qquad (6.1)$$

$$F_{(i)} P_i = \sum_{i=1}^{n} a_i P_i^3 + b_i P_i^2 + c_i P_i + d_i \qquad (6.2)$$

Subject to constraints:

Power balance constraint: The total real power generation is equal to the sum of total power demand and transmission losses.

$$\sum_{i=1}^{n} P_i = P_D + P_L \qquad (6.3)$$

Generator limit constraint: The real power generation of i_{th} committed generating unit should be within following limit.

$$P_{i\,min} \ll P_i \ll P_{i\,max} \qquad (6.4)$$

Transmission loss constraint: The total transmission loss $P_L$should be minimum and is given as George’s formula:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j \qquad (6.5)$$

Where $F_{CT}$is the fuel cost of all generators in \$/h,P_i is the real output power in MW of i_{th} generator,P_D , P_L are total demand and transmission losses in MW,$P_{i\,min}$, $P_{i\,max}$are the minimum and maximum power limits of i_{th} generator, n is the number of committed generating units,a_i, b_i, c_i, d_i are the fuel cost curve co-efficient of the i_{th} generators respectively.B_{ij} is the matrix of transmission loss coefficient of generating units.

6.2.2 Cubic gas emission model:

All the thermal power plants emit toxic gases like S02, NOx and CO2 due to burning of fossil fuel contributing to global emissions and needs to be minimized individually. In this model all the three emissions are individually mathematically formulated by cubic polynomial as follows:

$$E_{i(so2)}(P_i) = \ a_{i(so2)}P_i^3 + b_{i(so2)}P_i^2 + c_{i(so2)}P_i + d_{i(so2)} \qquad (6.7)$$

$$E_{i(NOx)}(P_i) = a_{i(NOx)}P_i^3 + b_{i(NOx)}P_i^2 + c_{i(NOx)}P_i + d_{i(NOx)} \qquad (6.8)$$

$$E_{i(CO2)}(P_i) = \ a_{i(CO2)}P_i^3 + b_{i(CO2)}P_i^2 + c_{i(CO2)}P_i + d_{i(CO2)} \qquad (6.9)$$

$$E_t = E_{i(SO2)}(P_i) + E_{i(NOx)}(P_i) + E_{i(CO2)}(P_i) \qquad (6.10)$$

Where $E_{i(SO2)}(P_i), E_{i(NOx)}(P_i), E_{i(CO2)}(P_i), E_t$ are total SO2, NOx, CO2 emissions and combined emissions in Kg/h.

6.2.3 Price Penalty Factors (PPF):

Price penalty factors [6,11,444] are formed by taking the ratio of fuel cost to the emission value and is used to convert the emission criteria into equivalent fuel cost for the emission. Following are the seven types of price penalty factors which are used in this thesis.

1. Min-Max price penalty factor, h_1

$$h_{1\,SO2} = \frac{(a_iP_{i\,min}^3 + b_iP_{i\,min}^2 + c_iP_{i\,min} + d_i)}{(a_{i(SO2)}P_{i\,max}^3 + b_{i(SO2)}P_{i\,max}^2 + c_{i(SO2)}P_{i\,max} + d_{i(SO2)})} \text{ \$/Kg} \qquad (6.11)$$

$$h_{1\,NOx} = \frac{(a_iP_{i\,min}^3 + b_iP_{i\,min}^2 + c_iP_{i\,min} + d_i)}{(a_{i(NOx)}P_{i\,max}^3 + b_{i(NOx)}P_{i\,max}^2 + c_{i(NOx)}P_{i\,max} + d_{i(NOx)})} \text{ \$/Kg} \qquad (6.12)$$

$$h_{1\,CO2} = \frac{(a_iP_{i\,min}^3 + b_iP_{i\,min}^2 + c_iP_{i\,min} + d_i)}{(a_{i(CO2)}P_{i\,max}^3 + b_{i(CO2)}P_{i\,max}^2 + c_{i(CO2)}P_{i\,max} + d_{i(CO2)})} \text{ \$/Kg} \qquad (6.13)$$

Whereas $h_{1\,SO2}$, $h_{1\,NOx}$, $h_{1\,CO2}$ are the min-max price penalty factors for SO2, NOx, CO2 emission.

2. Max-Max price penalty factor,h_2

$$h_{2\,SO2}=\frac{(a_iP_{i\,max}^3+b_iP_{i\,max}^2+c_iP_{i\,max}+d_i)}{(a_{i(SO2)}P_{i\,max}^3+b_{i(SO2)}P_{i\,max}^2+c_{i(SO2)}P_{i\,max}+d_{i(SO2)})}\ \text{\$/Kg} \qquad (6.14)$$

$$h_{2\,NOx}=\frac{(a_iP_{i\,max}^3+b_iP_{i\,max}^2+c_iP_{i\,max}+d_i)}{(a_{i(NOx)}P_{i\,max}^3+b_{i(NOx)}P_{i\,max}^2+c_{i(NOx)}P_{i\,max}+d_{i(NOx)})}\ \text{\$/Kg} \qquad (6.15)$$

$$h_{2\,CO2}=\frac{(a_iP_{i\,max}^3+b_iP_{i\,max}^2+c_iP_{i\,max}+d_i)}{(a_{i(CO2)}P_{i\,max}^3+b_{i(CO2)}P_{i\,max}^2+c_{i(CO2)}P_{i\,max}+d_{i(CO2)})}\ \$/Kg \qquad (6.16)$$

Whereas $h_{2\,SO2}$, $h_{2\,NOx}$, $h_{2\,CO2}$ are the max-max price penalty factors for SO2, NOx, CO2 emission.

3. Min-Min price penalty factor,h_3

$$h_{3\,SO2}=\frac{(a_iP_{i\,min}^3+b_iP_{i\,min}^2+c_iP_{i\,min}+d_i)}{(a_{i(SO2)}P_{i\,min}^3+b_{i(SO2)}P_{i\,min}^2+c_{i(SO2)}P_{i\,min}+d_{i(SO2)})}\ \text{\$/Kg} \qquad (6.17)$$

$$h_{3\,NOx}=\frac{(a_iP_{i\,min}^3+b_iP_{i\,min}^2+c_iP_{i\,min}+d_i)}{(a_{i(NOx)}P_{i\,min}^3+b_{i(NOx)}P_{i\,min}^2+c_{i(NOx)}P_{i\,min}+d_{i(NOx)})}\ \text{\$/Kg} \qquad (6.18)$$

$$h_{3\,CO2}=\frac{(a_iP_{i\,min}^3+b_iP_{i\,min}^2+c_iP_{i\,min}+d_i)}{(a_{i(CO2)}P_{i\,min}^3+b_{i(CO2)}P_{i\,min}^2+c_{i(CO2)}P_{i\,min}+d_{i(CO2)})}\ \text{\$/Kg} \qquad (6.19)$$

Whereas $h_{3\,SO2}$, $h_{3\,NOx}$, $h_{3\,CO2}$ are the min-min price penalty factors for SO2, NOx, CO2 emission.

4. Max-Min price penalty factor,h_4

$$h_{4\,SO2}=\frac{(a_iP_{i\,max}^3+b_iP_{i\,max}^2+c_iP_{i\,max}+d_i)}{(a_{i(SO2)}P_{i\,min}^3+b_{i(SO2)}P_{i\,min}^2+c_{i(SO2)}P_{i\,min}+d_{i(SO2)})}\ \text{\$/Kg} \qquad (6.20)$$

$$h_{4\ \ NOx}=\frac{(a_iP_{i\,max}^3+b_iP_{i\,max}^2+c_iP_{i\,max}+d_i)}{(a_{i(NOx)}P_{i\,min}^3+b_{i(NOx)}P_{i\,min}^2+c_{i(NOx)}P_{i\,min}+d_{i(NOx)})}\ \text{\$/Kg} \qquad (6.21)$$

$$h_{4\,CO2}=\frac{(a_iP_{i\,max}^3+b_iP_{i\,max}^2+c_iP_{i\,max}+d_i)}{(a_{i(CO2)}P_{i\,min}^3+b_{i(CO2)}P_{i\,min}^2+c_{i(CO2)}P_{i\,min}+d_{i(CO2)})}\ \text{\$/Kg} \qquad (6.22)$$

Whereas $h_{4\,SO2}$, $h_{4\,NOx}$, $h_{4\,CO2}$ are the max-min price penalty factors for SO2, NOx, CO2 emission.

5. Average penalty factor,h_5

$$h_{5\,SO2}=(h_{1\,SO2}+h_{2\,SO2}+h_{3\,SO2}+h_{4\,SO2})\div 4\ \ \text{\$/Kg} \qquad (6.23)$$

$$h_{5\,NOx}=(h_{1\,NOx}+h_{2\,NOx}+h_{3\,NOx}+h_{4\,NOx})\div 4\ \text{\$/Kg} \qquad (6.24)$$

$h_{5\,CO2} = (h_{1\,CO2} + h_{2\,CO2} + h_{3\,CO2} + h_{4\,CO2}) \div 4$ \$/Kg (6.25)

Whereas $h_{5\,SO2}$, $h_{5\,NOx}$, $h_{5\,CO2}$ are the average price penalty factors for SO2, NOx, CO2 emission.

6. New average Min-Max and Max-Max price penalty factor,h_6

$h_{6\,SO2} = (h_{1\,SO2} + h_{2\,SO2}) \div 2$ \$/Kg (6.26)

$h_{6\,NOx} = (h_{1\,NOx} + h_{2\,NOx}) \div 2$ \$/Kg (6.27)

$h_{6\,CO2} = (h_{1\,CO2} + h_{2\,CO2}) \div 2$ \$/Kg (6.28)

Whereas $h_{6\,SO2}$, $h_{6\,NOx}$, $h_{6\,CO2}$ are the new average price penalty factors for SO2, NOx, CO2 emission.

7. Common price penalty factor,h_7

$h_{7\,SO2} = h_{5\,SO2} \div n$ \$/Kg (6.29)

$h_{7\,NOx} = h_{5\,NOx} \div n$ \$/Kg (6.30)

$h_{7\,CO2} = h_{5\,CO2} \div n$ \$/Kg (6.31)

Whereas $h_{7\,SO2}$, $h_{7\,NOx}$, $h_{7\,CO2}$ are the common price penalty factors for SO2, NOx, CO2 emission and n is the number of generating power units.

6.2.4 Bi-objective CEED optimization problem (BOCEED) using PCPSO

Following are the bi-objective CEED equations by combining fuel cost with individual emission and which are further converted into single objective by multiplying a price penalty factor for the three emissions separately.

$F_{T\,SO2} = \sum_{i=1}^{n}[(a_iP_i^3 + b_iP_i^2 + c_iP_i + d_i) + h_{i\,SO2}(a_{i\,SO2}P_i^3 + b_{i\ SO2}P_i^2 + c_{i\,SO2}P_i + d_{i\,SO2})]$

\$/h (6.32)

$F_{T\,NO2} = \sum_{i=1}^{n}[(a_iP_i^3 + b_iP_i^2 + c_iP_i + d_i) + h_{i\,NOX}(a_{i\,NO2}P_i^3 + b_{i\ NO2}P_i^2 + c_{i\,NO2}P_i +$

$d_{i\,NO2})]$ \$/h (6.33)

$F_{T\,CO2} = \sum_{i=1}^{n}[(a_iP_i^3 + b_iP_i^2 + c_iP_i + d_i) + h_{i\,CO2}(a_{i\,CO2}P_i^3 + b_{i\ CO2}P_i^2 + c_{i\,CO2}P_i +$

$d_{i\,CO2})]$ \$/h (6.34)

6.2.5 Formulation of four objectives CEED optimization problem (FOCEED) using PCPSO

Fuel cost and emissions can be minimized together by converting multi-objective problem into a single objective problem, by taking all the three emissions taken simultaneously SO2, NOx and CO2 with price penalty factors are combined together along with the fuel cost $F_c$of committed generating units is carried out for a particular demand. The total fuel cost $F_{T\,CEED}$ is expressed as follows:

$$F_{T\,CEED} = \sum_{i=1}^{n}[(a_iP_i^3 + b_iP_i^2 + c_iP_i + d_i) + h_{i\,SO2}(a_{i\,SO2}P_i^3 + b_{i\,SO2}P_i^2 + c_{i\,SO2}P_i + d_{i\,SO2}) + h_{i\,NO2}(a_{i\,NO2}P_i^3 + b_{i\,NO2}P_i^2 + c_{i\,NO2}P_i + d_{i\,NO2}) + h_{i\,CO2}(a_{i\,CO2}P_i^3 + b_{i\,CO2}P_i^2 + c_{i\,CO2}P_i + d_{i\,CO2})] \text{ \$/h} \quad (6.35)$$

$$F_{C=} \sum_{i=1}^{n}(a_iP_i^3 + b_iP_i^2 + c_iP_i + d_i) \text{ \$/h} \quad (6.36)$$

6.3 Results and Discussions

6.3.1 FOCEED Problem

The PCPSO [401] technique has been applied to IEEE 30bus system having 6 units of thermal power plants at bus1, bus2, bus5, bus8, bus11 and bus13 with four conflicting objective functions having cubic cost and cubic emissions functions. The simulations were performed on Matlab 2015a platform on Compaq 6720s lab-top with 4GB RAM. The cubic cost coefficients, cubic emission coefficients likeE_{SO2}, E_{NOX},E_{CO2}, real power generator limits are taken from respectively [428].Seven types of price penalty factors are used to solve CEED optimization problem, individual emission analysis for E_{SO2}, E_{NOX},E_{CO2} with fuel cost and emission cost for various load conditions from 150 MW to 300MW respectively. The total numbers of iterations were taken as 25 with 50 numbers of trails. The results obtained from intelligent PCPSO [401] are compared with various algorithms like SA, GOA, PSO, MBO, AEO, CAE04 algorithm and Lagrange method respectively published recently in various journals. Table6.1 shows best results of total fuel costF_T, Power loss P_L,total emission

E_T ,CT (computational time),iter (number of iteration)obtained from CEED problem using PCPSO with seven price penalty function with different load condition like 150MW,175MW,200MW,225MW,250MW and 300MW and are compared with Lagrange method [428], PSO [429],SA[430,431],MBO [432],GOA algorithm [433], Artificial ecosystem based optimization (AEO)[434],Multi objective 4th chaotic function Artificial ecosystem based optimization (CAE04)[435] ,Quantum Particle swarm optimization (QPSO)[436] and Sine-cosine algorithm(SCA)[437].

Table6.1 CEED results with seven price penalty factors with different load demands and its comparison with different algorithms.

P_D (MW)	PPF	P_1 MW	P_2 MW	P_3 MW	P_4 MW	P_5 MW	P_6 MW	P_L MW	F_C \$/h	E_{SO2} Kg/h	E_{NOx} Kg/h	E_{CO2} Kg/h	E_T Kg/h	$F_{T\ CEED}$ \$/h	CT Sec	Iter
150	h_1	50.00	33.21	18.79	13.19	19.15	15.66	1.33	2726.90	2777.60	2241.30	2377.10	7396.10	3642.80	0.51	12
	h_1 **[428]**	50.02	21.06	16.0	16.94	34.05	12.23	------	2780.09	3198.07	2250.75	2693.28	---	3768.28	----	--
	h_1 **[431]**	50.00	20.00	15.00	26.52	26.47	12.00	-------	2735.61	3165.29	2350.14	2553.17	----	3718.21	----	----
	h_2	50.00	25.10	16.71	21.06	14.26	22.87	1.26	2593.10	2810.60	2202.30	2411.80	7424.60	10110.00	0.48	14
	h_2 **[428]**	52.86	21.34	18.57	22.22	20.45	15.42	---	2729.34	3091.64	2448.21	2537.12	---	10264.56	---	---
	h_2 [43**0]**	50.00	20.17	15.00	24.10	19.82	20.88	-----	2705.21	3138.44	2379.35	2568.94	---	10261.49	34.15	--
	h_2 **[431]**	50.00	22.06	15.00	22.71	18.60	21.61	----	2714.63	3114.31	2375.64	2565.00	----	10270.90	---	---
	h_2 **[432]**	50.00	20.00	15.00	25.26	17.30	22.43	--	2704.92	3146.83	2406.23	2564.57	---	10255.21	3.37	--
	h_2 **[434]**	50.00	20.02	15.00	24.09	16.17	24.70	--	2703.68	2978.03	2349.85	2480.41	--	10180.80	--	--
	h_2 [435]	50.00	20.00	15.00	23.96	18.98	22.03	--	2702.94	2885.21	2249.10	2448.86	--	10179.35	--	--
	h_2 **[436]**	50.00	20.00	15.00	25.25	17.28	22.47	--	2704.89	3146.86	2406.37	2564.82	--	10255.25	--	--
	h_2 **[437]**	50.00	20.00	15.00	25.26	17.30	22.43	--	2704.92	3146.83	2406.23	2564.56	--	10255.20	0.20	--

	h_3	50.00	52.15	49.57	34.13	7.99	7.70	2.38	2857.30	2984.20	2379.20	2559.70	7923.10	12269.00	0.63	16
	h_4	53.07	12.58	8.35	31.10	5.18	41.50	1.79	4791.70	5447.70	4064.10	4401.30	13913.00	65171.00	3.62	19
	h_5	52.09	19.18	16.17	8.03	5.16	51.30	1.96	4804.00	5409.70	4028.50	4573.50	14012.00	22315.00	2.58	09
	h_6	50.00	20.37	17.73	25.72	21.96	14.22	1.12	2555.20	2799.70	2253.30	2247.60	7300.70	6191.00	0.60	17
	$h_{6]}$ **[428]**	51.34	22.67	15.02	18.32	30.42	13.25	---	2757.58	3177.53	2268.27	2660.42	---	7500.75	---	--
	h_6 **[431]**	50.00	23.28	15.00	21.72	20.51	19.47	--	2724.08	3097.39	2352.18	2558.96	---	7043.58	---	---
	h_7	50.00	24.82	18.61	26.27	21.28	21.73	1.41	2987.30	3391.20	2586.80	2798.30	8776.40	6023.70	1.59	05
175	h_1	54.48	22.46	24.34	31.30	21.62	22.36	1.57	3346.60	3781.30	2941.30	3136.90	9859.50	4399.80	5.49	13
	h_2	50.00	31.04	16.02	30.76	17.18	28.45	1.52	2844.40	3131.60	2447.90	2670.70	8250.20	11105.00	9.10	20
	h_2 **[430]**	50.00	30.03	15.00	28.59	25.42	25.95	---	3220.51	3763.47	2789.92	3094.68	---	12280.04	31.38	--
	h_2 **[432]**	50.00	23.77	15.00	32.64	23.91	29.66	--	3188.12	3859.48	2854.13	3129.20	---	12241.67	3.17	--
	h_2 [**433]**	50.00	23.77	15.00	32.65	23.92	29.66	--	3188.08	3859.37	2854.08	3129.07	9842.52	12241.41	--	--
	h_2 $\mathbf{[434]}$	50.07	22.68	15.99	30.15	24.74	31.33	--	3187.30	3829.72	2883.35	3136.05	--	12172.10	--	--
	h_2 **[435]**	50.00	21.24	15.01	32.97	26.27	29.48	--	3179.33	3891.63	2847.56	3154.93	--	12164.40	--	--
	h_2 $\mathbf{[436]}$	50.00	23.74	15.00	32.62	23.91	29.72	--	3187.93	3859.89	2854.22	3129.91	--	12241.71	--	--
	h_2 **[437]**	50.00	23.77	15.00	32.63	23.92	29.66	--	3188.15	3859.35	2854.00	3129.18	--	12241.66	0.21	--
	h_3	50.00	22.58	19.80	31.00	20.69	30.93	1.27	2898.40	873.40	648.76	2596.30	4118.40	12530.00	0.73	18
	h_4	56.28	28.66	9.75	31.63	5.34	45.69	2.37	4384.70	4351.70	3548.00	4073.80	11974.00	20580.00	0.53	13
	h_5	50.00	43.28	19.46	12.77	5.46	47.21	2.36	4051.90	4218.60	3406.70	3708.70	11334.00	10275.00	0.61	20
	h_6	54.00	30.18	15.20	25.10	19.22	33.16	1.89	3295.00	3863.10	2929.10	3242.60	10035.00	8340.20	0.53	08
	h_7	50.00	22.44	16.25	38.33	20.37	30.70	1.75	3252.90	3989.50	2991.50	3174.40	10155.00	6392.50	1.06	06
	h_1	52.02	32.29	23.31	32.10	28.42	33.88	2.06	3828.50	1723.30	1228.10	3740.10	6691.50	4605.40	0.52	20
	h_2	50.00	32.38	19.51	38.06	29.49	30.02	1.50	2904.80	3205.10	2532.50	2722.90	8460.40	11316.00	0.59	11

200	h_2 **[430]**	50.00	32.90	15.00	36.57	30.93	34.58	---	3735.73	4553.97	3285.64	3714.33	---	14421.30	39.48	--
	h_2 **[432]**	50.00	31.02	15.00	38.82	29.48	35.66	---	3727.42	4592.63	3325.34	3715.68	--	14413.71	3.29	--
	h_2 **[433]**	50.00	31.04	15.00	38.82	29.48	35.66	--	3727.40	4592.46	3325.31	3715.56	11633.33	14413.52	--	--
	h_2 **[434]**	50.00	29.90	15.21	40.22	27.35	37.29	--	3724.70	4618.24	3362.89	3719.47	--	14293.19	--	--
	h_2 **[435]**	50.00	29.60	15.00	38.67	30.32	36.39	--	3722.45	4589.86	3301.52	3725.20	--	14287.65	--	--
	$\boldsymbol{h_2}$ **[436]**	50.00	31.10	15.00	38.77	29.47	35.66	--	3727.61	4591.38	3324.62	3715.45	--	14413.77	--	--
	$\boldsymbol{h_2}$ **[437]**	50.00	31.03	15.00	38.82	29.47	35.66	----	3727.44	4592.49	3325.33	3715.63	--	14413.70	0.21	--
	h_3	52.75	73.34	34.29	17.45	9.55	13.52	2.91	4667.10	4750.00	3597.30	4247.70	12595.00	20531.00	0.71	10
	h_4	50.00	33.47	43.69	30.38	11.14	31.32	2.94	2934.10	3077.20	2370.90	2678.70	8126.80	13911.00	0.48	11
	h_5	50.00	31.21	32.95	26.13	5.28	55.37	2.89	4004.80	4403.40	3423.50	3787.30	11614.00	16075.00	0.52	12
	h_6	54.46	36.49	22.05	30.46	22.17	36.59	2.23	3870.40	4470.50	3398.40	3774.40	11643.00	9911.20	0.45	05
	h_7	50.42	50.69	22.38	32.57	10.27	36.06	2.41	3962.00	4416.30	3448.70	3689.80	11555.00	11059.00	0.80	11
225	h_1	61.98	22.37	29.25	41.27	49.17	23.38	2.44	4673.00	5507.70	3946.50	4658.00	14112.00	6093.80	1.77	16
	h_2	50.00	36.87	16.94	45.81	36.01	39.37	2.25	3723.10	3065.00	2365.50	2527.50	7958.10	10708.00	0.57	06
	h_2 **[430]**	50.00	38.90	18.69	42.05	35.35	40.00	---	4321.51	5287.30	3781.19	4324.30	----	16790.69	21.92	--
	h_2 **[432]**	50.00	37.24	18.05	44.32	35.37	40.00	--	4315.63	5335.80	3811.17	4328.07	---	16784.34	2.75	--
	h_2 **[433]**	50.00	37.66	17.83	44.72	34.79	40.00	--	4315.15	5338.42	3819.56	4322.26	13480.24	16783.68	--	--
	h_2 **[434]**	50.00	37.20	16.80	44.80	36.4	39.7	--	4315.00	5235.70	3758.10	4255.70	---	16617.10	--	---
	h_2 **[435]**	50.00	39.42	16.07	45.27	34.26	39.96	--	4314.38	5360.57	3832.06	4322.68	--	16603.90	--	--
	$\boldsymbol{h_2}$ **[436]**	50.00	37.68	17.75	44.77	34.81	40.00	--	4315.03	5340.20	3820.14	4322.70	--	16783.86	--	--
	$\boldsymbol{h_2}$ **[437]**	50.00	37.67	17.76	44.77	34.79	39.99	--	4315.04	5339.96	3820.28	4322.52	--	16783.78	0.21	--
	h_3	58.47	65.38	43.85	37.08	9.71	13.52	3.03	3915.20	4236	3254	374	1123	23093.	3.	18

										.60	.00	0.10	1.00	00	07	
	h_4	50.00	67.27	36.06	32.27	13.20	26.07	3.13	3582.80	3738.10	2953.20	3165.10	9856.40	17623.00	0.46	16
	h_5	50.00	21.56	59.03	41.98	27.63	25.24	2.40	3700.20	1337.80	941.99	3294.60	5574.40	31017.00	0.51	07
	h_6	50.38	50.13	27.25	34.81	25.71	37.35	2.62	4472.00	5067.50	3812.20	4286.60	13166.00	11653.00	0.55	19
	h_7	50.00	30.30	37.84	41.16	48.29	30.09	2.33	4796.60	5491.70	3944.40	4682.70	14119.00	8141.80	1.80	08
250	h_1	58.75	36.72	30.90	48.23	45.10	33.41	2.95	5159.90	6111.50	4438.10	5093.70	15643.00	6921.60	0.90	15
	h_1 **[429]**	57.62	41.00	20.98	50.00	49.03	35.59	4.24	5181.20	6352.70	4452.80	5217.90	---	7487.10	---	---
	h_2	50.00	43.26	31.11	48.78	40.41	36.44	2.42	4552.30	5359.60	3879.50	4489.50	13729.00	18887.00	1.87	14
	h_2 **[429]**	50.00	45.42	23.36	50.00	42.38	40.00	4.18	5081.10	6143.00	4376.40	5027.20	---	19832.00	---	---
	h_3	50.00	22.49	80.00	50.00	27.50	30.00	2.95	4974.00	2347.10	1559.40	4713.00	8619.50	23424.00	0.53	13
	h_4	50.00	29.70	73.10	49.46	27.74	30.00	2.84	3791.80	1412.20	952.83	3421.10	5786.10	94687.00	0.56	05
	h_5	50.00	37.16	64.84	50.00	28.00	30.00	2.77	3793.20	1499.00	1165.50	3394.40	6058.90	34145.00	0.52	02
	h_6	52.29	59.44	19.14	44.86	38.81	38.93	3.48	5080.10	6142.70	4366.40	5069.70	15579.00	13135.00	0.78	03
	h_6 **[429]**	50.01	48.60	23.37	50.00	42.24	40.00	4.23	5079.70	6181.10	4379.60	5041.40	---	14547.00	---	---
	h_7	75.91	41.01	42.98	34.11	25.46	33.81	3.31	5543.60	5942.80	4868.20	5369.80	16181.00	10667.00	1.66	06
300	h_1	67.80	74.79	43.03	48.41	34.95	35.52	4.53	6744.70	7611.90	5799.80	6602.50	20014.00	9337.70	0.72	08
	h_1 **[429]**	70.86	59.97	35.99	50.00	50.00	40.00	6.84	6772.80	7878.40	5771.10	6797.40	----	9938.50	---	---
	h_2	62.47	59.90	37.62	50.00	50.00	40.00	4.56	6731.40	7348.20	5610.90	6591.70	19551.00	17890.00	2.09	01
	h_2 **[429]**	58.39	65.75	42.55	50.00	50.00	40.00	6.71	6667.50	7665.70	5557.70	6591.40	---	26525.00	---	---
	h_3	89.08	62.51	73.33	40.55	9.81	30.30	5.60	5174.70	5394.90	4305.00	4721.90	14422.00	37458.00	2.10	10
	h_4	98.18	66.02	50.92	31.32	12.13	47.36	5.96	16479.00	10721.00	12504.00	20013.00	43238.00	68191.00	0.54	20
	h_5	98.31	65.41	53.17	49.63	12.05	26.95	5.55	10244.00	9937.20	9436.40	10987.00	30361.00	86380.00	0.48	13
	h_6	61.38	61.06	47.30	48.21	49.45	36.56	3.99	6642.70	7475.80	5545.90	6514.70	19536.00	17361.00	0.83	12
	h_6 **[429]**	52.14	74.20	40.35	50.00	50.00	40.00	6.70	6649.90	7727.70	5485.70	6605.50	----	19067.00	---	---
	h_7	64.45	81.84	60.42	43.91	14.89	39.38	4.92	10184.00	9929.60	9279.20	10893.00	30102.00	20717.00	1.32	13

For a power demand of 150MW in h_1 category the fuel cost is lowest as compared with Langrage method[428] and Simulated annealing[431] along with the individual pollutants(E_{SO2}, E_{NOX}, E_{CO2}) ,with a CEED value of 3642.80 \$/h which is 75.41 \$/h less than SA [431] and Lagrange method [428] in just 0.51sec with 12 iterations only. With the same demand, considering h_2 max-max price penalty factor the fuel cost 2593.10 \$/hr is lowest again with all the emissions and CEED value of 10110.0 \$/hr as compared to Langrage [428], MBO[432], SA[430,431],AEO[434],CAE04[435],QPSO[436] and SCA[437] with very low computational time of 0.48 seconds in just 6 iterations. Now another $h_6$new average price penalty factor results in lowest fuel cost 2555.20 \$/hr with lowest all pollutants and CEED value of 6191.0 \$/hr as compared with Langrage [428] and SA [431] with very small CT value, showing the efficiency of PCPSO. The results obtained in case of 175MW with $h_2$max- max price penalty factor is compared with the results of SA [430], MBO [432], GOA [433], AEO[434],CAE04[435],QPSO[436] and SCA[437] showing excellent values of fuel cost 2844.40 \$/hr, low individual pollutants with lowest remarkable CEED value of 11105.0\$/hr in very small time and iteration. For power demand 200MW, results from h_2 max-max price penalty factor with various latest optimization algorithms research papers SA[430],MBO[432], GOA [433], AEO[434],CAE04[435],QPSO[436] and SCA[437] are compared in all parameters listed above are still lowest and shows the efficient characteristic of the PCPSO algorithm. Further with load demand of 225MW in the same category of max-max price penalty factor h_2 fuel cost is 3723.10 \$/h which is 592.05 \$/h less than lowest value of SA[430],MBO[432], GOA [433], AEO[434],CAE04[435],QPSO[436] and SCA[437] with lowest CEED value of 10708.0\$/hr along with lower individual pollutants. In case of 250MW demand load the fuel cost is still less in min-maxh_1, max-max h_2 and new average h_6 class as compared to PSO [429] including emissions, power loss P_L and CEED value showing better convergence characteristics with excellent results. Similarly with 300MW demand load and with these three price penalty factors, this algorithm was able to make lowest CEED value, lower power losses P_L along with individual pollutants in very small time showing its computational efficiency in few iterations. The CEED results clearly shows the changes in power output, fuel cost, individual pollutants, and total emission with the change in price penalty factors. Finally it is concluded that CEED value depend upon the price penalty factor when load demand varies from 150MW to 300MW in the following sequence from lowest to highest CEED value.

$$h_1 < h_6 < h_7 < h_2 < h_3 < h_5 < h_4$$

Following table6.2 shows the comparison of CEED results with various algorithms Langrage method [428], PSO [429], SA [430, 431] and MBO [432], with seven price penalty factors for load demand from 150 MW to 300 MW. For demand load of 150MW to 300mw PCPSO has performed well with all seven price penalty factors as compared to Langrage method[428], PSO[429] SA[430,431] and MBO[432]. Moreover, mostly research papers had taken max-max h_2 /min-max h_1 price penalty factors and resulted in inferior optimal solutions as compared to PCPSO [401].

Following table6.3 shows the comparison with all the available research papers for fuel cost, individual pollutants, total emission and CEED value for max-max price penalty factor only. For all the 6 different load demands, intelligent PCPSO[401] has lowest fuel cost except at last load level but individual emissions were lowest which results in lowest total emissions and CEED value when compared to Lagrange method [428], PSO [429],SA[430,431],MBO [432], GOA algorithm [433], Artificial ecosystem based optimization (AEO)[434],Multi objective 4th chaotic function Artificial ecosystem based optimization (CAE04)[435] ,Quantum Particle swarm optimization (QPSO)[436] and Sine-cosine algorithm(SCA)[437]using $h_2$max-max price penalty factor only. Figure6.1 Shows the comparison of various algorithms for $F_{T\,CEED}$using max-max price penalty function.

Table6.2 Comparison of $F_{T\,CEED}$ results by various algorithms on different price penalty factors.

P_D MW	Algorithms	$h_{min-max}$ h_1	$h_{max-max}$ h_2	$h_{min-min}$ h_3	$h_{max-min}$ h_4	$h_{average}$ h_5	$h_{new\ average}$ h_6	h_{common} h_7
150	PCPSO	3642.80	10110.00	12269.00	65171.00	22315.00	6191.00	6023.70
	LAGR[428]	3768.28	10264.56	---	----	---	7500.75	8122.63
	PSO[429]	---	10385.00	---	----	----	---	-----
	SA[430,431]	3718.21	10261.49	----	--	--	7043.58	----
	MBO[432]	---	10255.21	---	---	----	----	----
175	PCPSO	4399.80	11105.00	12530.00	20580.00	10275.00	8340.20	6392.50
	LAGR[428]	4551.67	13251.51	----	-----	----	8987.32	9802.29
	PSO[429]	----	12425.00	----	----	---	---	----
	SA[430,431]	4474.78	12280.04	----	-----		8417.25	---
	MBO[432]	-----	12241.67	---	-----	----	----	----
200	PCPSO	4605.40	11316.00	20531.00	13911.00	16075.00	9911.20	11059.00
	LAGR[428]	5438.00	16077.40	----	------	------	11623.52	10137.40
	PSO[429]	-----	14642.00	-----	-----	-----	-----	-----
	SA[430,431]	5337.56	14421.30	-----	----	---	9923.07	----
	MBO[432]	----	14413.71	----	----	----	----	-----
225	PCPSO	6093.80	10708.00	23093.00	17623.00	31017.00	11653.00	8141.80
	LAGR[428]	6418.90	19661.32	----	----	------	13283.85	14936.26
	PSO[429]	6418.90	17126.00	----	----	-----	12583.00	----
	SA[430,431]	6283.04	16790.69	----	-----	----	11570.32	-----
	MBO[432]	----	16784.34	----	-----	-----	-----	-----
250	PCPSO	6921.60	18887.00	23424.00	94687.00	34145.00	13135.00	10667.00
	PSO[429]	7487.10	19832.00	-----	------	-----	14547.00	---
300	PCPSO	9337.70	17890.00	37458.00	68191.00	86380.00	17361.00	20717.00
	PSO[429]	9938.50	26525.00	----	-----	-----	19067.00	----

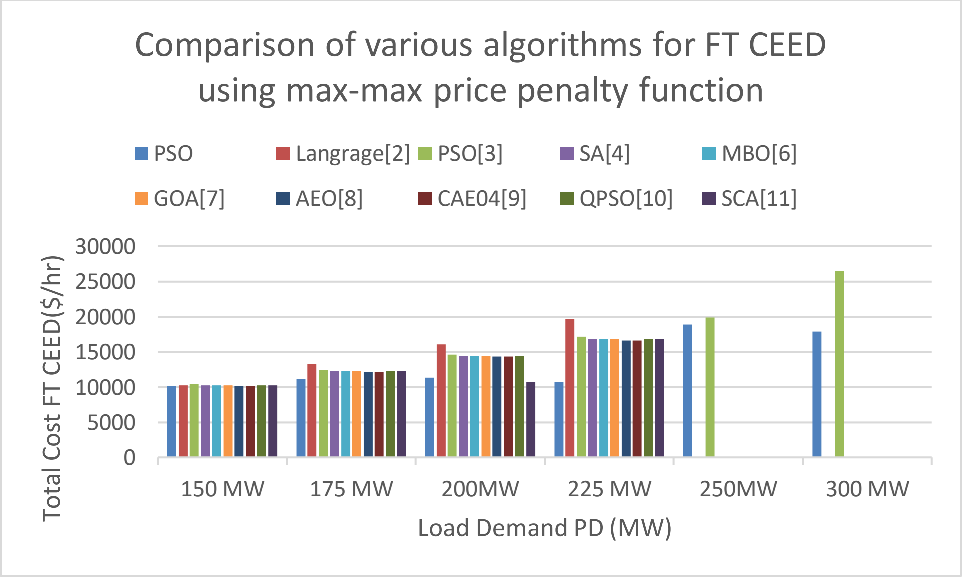


Fig6.1 Comparison of various algorithms for $F_{T\,CEED}$ using max-max price penalty function

Table 6.3 comparison of Fuel cost, individual pollutants(E_{SO2}, E_{NOX}, E_{CO2}), Total emission E_T and $F_{T\,CEED}$ results for load demand from 150MW to 300MW using h_2 max-max price penalty factors

	USING MAX-MAX PRICE PENALITYh_2						
P_D MW	Algorithms	F_C $/h	E_{SO2} Kg/h	E_{NOx} Kg/h	E_{CO2} Kg/h	E_T Kg/h	$F_{T\,CEED}$ $/h
150	PCPSO	2593.10	2810.60	2202.30	2411.80	7424.60	10110.00
	LAG[428]	2729.34	3091.64	2448.21	2537.12		10264.56
	PSO[429]	2734.20	3193.60	2424.60	2607.10	----	10385.00
	SA[430]	2705.21	3138.44	2379.35	2568.94	---	10261.49
	MBO[432]	2704.92	3146.83	2406.23	2564.57	---	10255.21
	GOA[433]	2704.87	3146.74	2406.18	2564.50	----	10254.98
	AEO[434]	2703.68	2978.03	2349.85	2480.41	--	10180.80
	CAE04[435]	2702.94	2885.21	2249.10	2448.86	--	10179.35
	QPSO[436]	2704.89	3146.86	2406.37	2564.82	--	10255.25
	SCA[437]	2704.92	3146.83	2406.23	2564.56	--	10255.20
175	PCPSO	2844.40	3131.60	2447.90	2670.70	8250.20	11105.00
	LAG[428]	3475.40	4146.17	2604.88	3613.53		13251.51
	PSO[429]	3236.30	3904.90	2879.70	3178.00	----	12425.00
	SA[430]	3220.51	3763.47	2789.92	3094.68	---	12280.04
	MBO[432]	3188.12	3859.48	2854.13	3129.20	---	12241.67
	GOA[433]	3188.08	3859.37	2854.08	3129.07	---	12241.41
	AEO[434]	3187.30	3829.72	2883.35	3136.05	--	12172.10
	CAE04[435]	3179.33	3891.63	2847.56	3154.93	--	12164.40
	QPSO[436]	3187.93	3859.89	2854.22	3129.91	--	12241.71
	SCA[437]	3188.15	3859.35	2854.00	3129.18	--	12241.66
200	PCPSO	2904.80	3205.10	2532.50	2722.90	8460.40	11316.00
	LAG[428]	4210.30	5053.58	3102.07	4473.36		16077.40
	PSO[429]	3784.90	4670.60	3373.20	3771.50	----	14642.00
	SA[430]	3735.73	4553.97	3285.64	3714.33	----	14421.30
	MBO[432]	3727.42	4592.63	3325.34	3715.68	----	14413.71
	GOA[433]	3727.40	4592.96	3325.31	3715.56	--	14413.52
	AEO[434]	3724.70	4618.24	3362.89	3719.47	--	14293.19
	CAE04[435]	3722.45	4589.86	3301.52	3725.20	--	14287.65
	QPSO[436]	3727.61	4591.38	3324.62	3715.45	--	14413.77
	SCA[437]	3727.44	4592.49	3325.33	3715.63	--	14413.70
225	PCPSO	3723.10	3065.00	2365.50	2527.50	7958.10	10708.00
	LAG[428]	5130.53	6106.49	3798.38	5502.52	----	19661.32
	SA[430]	4321.51	5287.30	3781.19	4324.30	---	16790.69
	PSO[429]	4402.30	5426.10	3877.60	4403.00	-----	17125.00
	MBO[432]	4315.63	5335.80	3811.17	4328.07	----	16784.34
	GOA[433]	4315.15	5338.42	3819.56	4322.26	----	16783.68
	AEO[434]	4315.00	5235.70	3758.10	4255.70	---	16617.10
	CAE04[435]	4314.38	5360.57	3832.06	4322.68	--	16603.90
	QPSO[436]	4315.03	5340.20	3820.14	4322.70	--	16783.86
	SCA[437]	4315.04	5339.96	3820.28	4322.52	--	16783.78
250	PCPSO	4552.30	5359.60	3879.50	4489.50	13729.00	18887.00
	PSO[429]	5081.10	6143.00	4376.40	5027.20	----	19832.00
300	PCPSO	6731.40	7348.20	5610.90	6591.70	19551.00	17890.00
	PSO[429]	6667.50	7665.70	5557.70	6591.40	---	26525.00

6.3.2 Bi-objective function CEED (BICEED)

In this section IEEE 30 Bus system 6 generating thermal units are considered as test system with load demand from 150MW to 300MW. Three cases are there which follow eq. 31,32, 33 independently.(A) Fuel cost with SO2 emission (B) Fuel cost with NO2 emission (C)Fuel cost with CO2emission.Following tables 4, 6, 7 shows the$F_{T\,so2}, F_{T\,NoX}, F_{T\,Co2}$ results considering each case independently with seven price penalty functions with different demand load levels.

Table6.4. $F_{T\,so2}$Results by PCPSO considering only fuel cost with SO2 emission and comparison with various algorithms on seven price penalty factors.

P_D (MW)	PPF	P_1 MW	P_2 MW	P_3 MW	P_4 MW	P_5 MW	P_6 MW	P_L MW	F_C $/hr	E_{SO2} Kg/hr	$F_{T\,SO2}$ $/hr	CT Sec	Iter
150	h_1	50.00	33.21	18.79	13.19	19.15	15.66	1.33	2726.90	2777.60	2254.20	0.51	12
	h_2	50.00	25.10	16.71	21.06	14.26	22.87	1.26	2593.10	2810.60	6134.20	0.48	14
	h_3	50.00	52.15	49.57	34.13	7.99	7.70	2.38	2857.30	2984.20	6463.70	0.63	16
	h_4	53.07	12.58	8.35	31.10	5.18	41.50	1.79	4791.70	5447.70	27522.00	3.62	19
	h_5	52.09	19.18	16.17	8.03	5.16	51.30	1.96	4804.00	5409.70	16275.00	2.58	09
	h_6	50.00	20.37	17.73	25.72	21.96	14.22	1.12	2555.20	2799.70	2970.80	0.60	17
	h_7	50.00	24.82	18.61	26.27	21.28	21.73	1.41	2987.30	3391.20	3391.20	1.59	05
175	h_1	54.48	22.46	24.34	31.30	21.62	22.36	1.57	3346.60	3781.30	3039.70	5.49	13
	h_2	50.00	31.04	16.02	30.76	17.18	28.45	1.52	2844.40	3131.60	7743.30	9.10	20
	h_3	50.00	22.58	19.80	31.00	20.69	30.93	1.27	2898.40	873.40	3942.80	0.73	18
	h_4	56.28	28.66	9.75	31.63	5.34	45.69	2.37	4384.70	4351.70	48350.00	0.53	13
	h_5	50.00	43.28	19.46	12.77	5.46	47.21	2.36	4051.90	4218.60	16122.00	0.61	20
	h_6	54.00	30.18	15.20	25.10	19.22	33.16	1.89	3295.00	3863.10	6932.70	0.53	08
	h_7	50.00	22.44	16.25	38.33	20.37	30.70	1.75	3252.90	3989.50	5307.60	1.06	06
200	h_1	52.02	32.29	23.31	32.10	28.42	33.88	2.06	3828.50	1723.30	4769.70	0.52	20
	h_2	50.00	27.38	19.51	19.06	10.49	30.02	1.50	2904.80	3205.10	8219.20	0.59	11
	h_2 **[429]**	54.23	30.09	21.06	33.33	30.53	30.97	-----	3739.99	4458.15	7481.40	--	--
	h_3	52.75	73.34	34.29	17.45	9.55	13.52	2.91	4667.10	4750.00	4318.30	0.71	10
	h_4	50.00	33.47	64.97	43.69	30.38	11.14	31.32	2934.10	3077.20	17588.00	0.48	11
	h_5	50.00	31.21	32.95	26.13	5.28	55.37	2.89	4004.80	4403.40	17517.00	0.52	12
	h_6	54.46	36.49	22.05	30.46	22.17	36.59	2.23	3870.40	4470.50	7761.50	0.45	05
	h_7	50.42	50.69	22.38	32.57	10.27	36.06	2.41	3962.00	4416.30	6424.90	0.80	11
225	h_1	61.98	22.37	29.25	41.27	49.17	23.38	2.44	4673.00	5507.70	3174.00	1.77	16
	h_2	50.00	36.87	16.94	45.81	36.01	39.37	2.25	3723.10	3065.00	5184.40	0.57	06
	h_3	58.47	65.38	43.85	37.08	9.71	13.52	3.03	3915.20	4236.60	9159.10	3.07	18
	h_4	50.00	67.27	36.06	32.27	13.20	26.07	3.13	3582.80	3738.10	24907.00	0.46	16
	h_5	50.00	21.56	59.03	41.98	27.63	25.24	2.40	3700.20	1337.80	8439.40	0.52	02
	h_6	50.38	50.13	27.25	34.81	25.71	37.35	2.62	4472.00	5067.50	7951.30	0.55	19
	h_7	50.00	30.30	37.84	41.16	48.29	30.09	2.33	4796.60	5491.70	4542.20	1.80	08

250	h_1	58.57	36.72	30.90	48.23	45.10	33.41	2.95	5159.90	6111.50	4575.20	0.90	15
	h_2	50.00	43.26	31.11	48.78	40.41	36.44	2.42	4552.30	5359.60	10321.00	1.87	14
	h_3	50.00	22.49	80.00	50.00	27.50	30.00	2.95	4974.00	2347.10	9435.10	0.53	13
	h_4	50.00	29.70	73.10	49.46	27.74	30.00	2.84	3791.80	1412.20	21964.00	0.56	05
	h_5	50.00	37.16	64.84	50.00	28.00	30.00	2.77	3793.20	1499.00	15817.00	0.52	02
	h_6	52.29	59.44	19.14	44.86	38.81	38.93	3.48	5080.10	6142.70	8299.30	0.78	03
	h_7	75.91	41.01	42.98	34.11	25.46	33.81	3.31	5543.60	5942.80	5300.70	1.66	06
300	h_1	67.80	74.79	43.03	48.41	34.95	35.52	4.53	6744.70	7611.90	4892.40	0.72	08
	h_2	62.47	59.90	37.62	50.00	50.00	40.00	4.56	6731.40	7348.20	8756.90	2.09	01
	h_3	89.08	62.51	73.33	40.55	9.81	30.30	5.60	5174.70	5394.90	9663.30	2.10	10
	h_4	98.18	66.02	50.92	31.32	12.13	47.36	5.96	16479.00	10721.00	28115.00	0.54	20
	h_5	98.31	65.41	53.17	49.63	12.05	26.95	5.55	10244.00	9937.20	14022.00	0.48	13
	h_6	61.38	61.06	47.30	48.21	49.45	36.56	3.99	6642.70	7475.80	7707.10	0.83	12
	h_7	64.45	81.84	60.42	43.91	14.89	39.38	4.92	10184.00	9929.60	2727.30	1.32	13

In table 6.4 for the load demand of 150MW and 175MW, new average h_6 and max-max h_2 price penalty factor has performed good results. In case of 200MW load demand with $h_2$max-max price penalty factor the results are compared with PSO [429] yielding low fuel cost with low SO2 emission with quite higher $F_{T\,SO2}$ value. But in the case of 225MW, 250MW and 300MW, the new average price penalty factor h_6 results in excellent fuel cost, SO2 emission and total cost. Hence for bi-objective function with SO2 emission, the price penalty factor sequence is as follows for optimal results with load demand varying from 150 MW to 300MW.

$$h_6 < h_1 < h_2 < h_7 < h_5 < h_3 < h_4$$

Table6.5 Comparison of best results considering fuel cost and SO2 emission with other algorithms.

		150 MW	**175 MW**	**200 MW**	**225 MW**	**250 MW**	**300 MW**
PCPSO	P1	50.00	50.00	54.46	50.38	52.29	61.38
	P2	20.37	20.12	36.49	52.13	59.44	61.06
	P3	17.73	22.27	22.05	27.25	19.14	47.30
	P4	25.72	33.03	30.46	34.81	44.86	48.21
	P5	10.52	13.58	22.17	25.71	38.81	49.45
	P6	14.22	15.65	36.59	37.35	38.93	36.56
	F_C $/hr	**2555.20**	**2883.50**	**3870.40**	**4472.00**	**5080.10**	**6642.70**
	E_{SO2} Kg/hr	**2799.70**	**3131.60**	**4470.50**	**5067.50**	**6142.70**	**7475.80**
Langrage[428]		3091.64	4146.17	5053.58	6106.49	---	----
PSO[429]		3193.60	3904.90	4670.60	5426.10	6143.00	7665.70
SA[430]		3138.44	3763.47	4553.97	5287.30	----	----

MBO[432]	3146.83	3859.48	4592.63	5335.80	----	----
GOA[433]	3146.74	3859.37	4592.96	5338.42	----	-----
AEO[434]	2978.03	3829.72	4618.24	5235.70	----	-----
CAE04[435]	2885.21	3891.63	4589.86	5360.57	----	-----
QPSO[436]	3146.86	3859.89	4591.38	5340.20	-----	-----
SCA[437]	3146.83	3859.35	4592.49	5339.96	----	-----

From table6.5 it is clear that in each load level from 150MW to 300MW the fuel cost and SO2 emission are minimum as compared to Lagrange method [428], PSO [429],SA[430,431],MBO [432], GOA algorithm [433], Artificial ecosystem based optimization (AEO)[434],Multi objective 4th chaotic function Artificial ecosystem based optimization (CAE04)[435] ,Quantum Particle swarm optimization (QPSO)[436] and Sine-cosine algorithm(SCA)[437].Following fig.2 shows the comparison of various algorithms for E_{SO2} emission at different load levels.

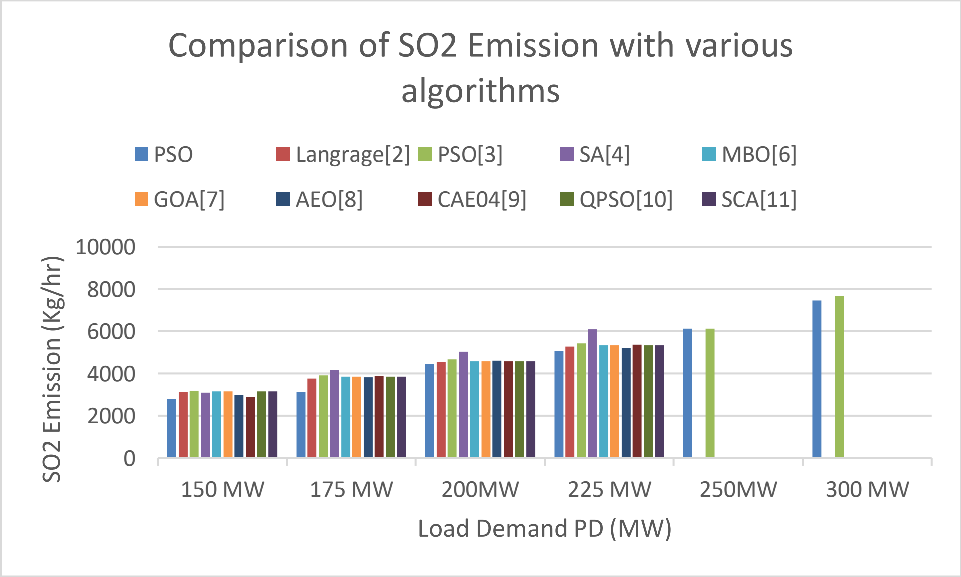


Fig6.2 Shows the comparison of various algorithms for SO2 emission at different load levels.

In the following table6 the results of bi objective function considering NOx emission using seven price penalty function with load demand in 7 levels from 150MW TO 300MW are given. For 150MW and 175MW, new average and max-max price penalty factors (PPF) results in optimal value of power lossP_L, Fuel cost, NOx emission and $F_{T\ NOx}$ value in a small computational time. In case of 200 MW load demand with $h_2$max-max price penalty factor is compared with recent research paper PSO [429], PCPSO results in 846.95 $/hr lower fuel cost and 735.33 Kg/hr lower NOx emission with 641.06 $/hr lower$F_{T\,NOx}$ cost value with better efficiency. Whereas with load demand of 225MW this algorithm shows better results with $h_2$max-max price penalty factor but 250MWand 300MW load demand , $h_6$new average price penalty factor shows excellent results in remarkable time period in few iterations.

Following price penalty factor sequence shows the impact on the power lossP_L, Fuel cost, NOx emission and $F_{T\,NOx}$ value.

$$h_6 < h_2 < h_1 < h_7 < h_4 < h_5 < h_3$$

Table6.6. $F_{T\,NOx}$ Results by PCPSO considering only fuel cost with NOx emission and comparison with various algorithms on seven price penalty factors.

P_D (MW)	PPF	P_1 MW	P_2 MW	P_3 MW	P_4 MW	P_5 MW	P_6 MW	P_L MW	F_C $/h	E_{NOx} Kg/h	$F_{T\,NOx}$ $/h	CT Sec	Iter
150	h_1	50.00	33.21	18.79	13.19	19.15	15.66	1.33	2726.90	2241.30	1805.80	0.51	12
	h_2	50.00	25.10	16.71	21.06	14.26	22.87	1.26	2593.10	2202.30	4917.50	0.48	14
	h_3	50.00	52.15	49.57	34.13	7.99	7.70	2.38	2857.30	2379.20	10658.00	0.63	16
	h_4	53.07	12.58	8.35	31.10	5.18	41.50	1.79	4791.70	4064.10	877970.00	3.62	19
	h_5	52.09	19.18	16.17	8.03	5.16	51.30	1.96	4804.00	4028.50	433020.00	2.58	09
	h_6	50.00	20.37	17.73	25.72	21.96	14.22	1.12	2555.20	2253.30	2079.60	0.60	17
	h_7	50.00	24.82	18.61	26.27	21.28	21.73	1.41	2987.30	2586.80	2975.10	1.59	05
175	h_1	54.48	22.46	24.34	31.30	21.62	22.36	1.57	3346.60	2941.30	2704.30	5.49	13
	h_2	50.00	31.04	16.02	30.76	17.18	28.45	1.52	2844.40	2447.90	6355.70	9.10	20
	h_3	50.00	22.58	19.80	31.00	20.69	30.93	1.27	2898.40	648.76	48251.00	0.73	18
	h_4	56.28	28.66	9.75	31.63	5.34	45.69	2.37	4384.70	3548.00	913420.00	0.53	13
	h_5	50.00	43.28	19.46	12.77	5.46	47.21	2.36	4051.90	3406.70	242550.00	0.61	20
	h_6	54.00	30.18	15.20	25.10	19.22	33.16	1.89	3295.00	2929.10	6257.90	0.53	08
	h_7	50.00	22.44	16.25	38.33	20.37	30.70	1.75	3252.90	2991.50	4612.80	1.06	06
200	h_1	52.02	32.29	23.31	32.10	28.42	33.88	2.06	3828.50	1228.10	4774.80	0.52	20
	h_2	50.00	27.38	19.51	19.06	10.49	30.02	1.50	2904.80	2532.50	6778.80	0.59	11
	h_2 **[429]**	52.39	28.71	21.76	33.67	29.32	34.52	---	3751.75	3267.83	7419.86	---	-----
	h_3	52.75	73.34	34.29	17.45	9.55	13.52	2.91	4667.10	3597.30	57015.00	0.71	10
	h_4	50.00	33.47	43.69	30.38	11.14	31.32	2.94	2934.10	2370.90	243490.00	0.48	11
	h_5	50.00	31.21	32.95	26.13	5.28	55.37	2.89	4004.80	3423.50	267480.00	0.52	12
	h_6	54.46	36.49	22.05	30.46	22.17	36.59	2.23	3870.40	3398.40	7127.80	0.45	05
	h_7	50.42	50.69	22.38	32.57	10.27	36.06	2.41	3962.00	3448.70	5369.20	0.80	11
	h_1	61.98	22.37	29.25	41.27	49.17	23.38	2.44	4673.00	3946.50	2859.50	1.77	16
	h_2	50.00	36.87	16.94	45.81	36.01	39.37	2.25	3723.10	3065.00	4062.30	0.57	06

225	h_3	58.47	65.38	43.85	37.08	9.71	13.52	3.03	3915.20	3254.00	168150.00	3.07	18
	h_4	50.00	67.27	36.06	32.27	13.20	26.07	3.13	3582.80	2953.20	40571.00	0.46	16
	h_5	50.00	21.56	59.03	41.98	27.63	25.24	2.40	3700.20	941.99	104110.00	0.52	02
	h_6	50.38	50.13	27.25	34.81	25.71	37.35	2.62	4472.00	5067.50	7327.00	0.55	19
	h_7	50.00	30.30	37.84	41.16	48.29	30.09	2.33	4796.60	3944.40	4172.60	1.80	08
250	h_1	58.57	36.72	30.90	48.23	45.10	33.41	2.95	5159.90	4438.10	4484.50	0.90	15
	h_2	50.00	43.26	31.11	48.78	40.41	36.44	2.42	4552.30	3879.50	8515.20	1.87	14
	h_3	50.00	22.49	80.00	50.00	27.50	30.00	2.95	4974.00	1559.40	174430.00	0.53	13
	h_4	50.00	29.70	73.10	49.46	27.74	30.00	2.84	3791.80	952.84	34118.00	0.56	05
	h_5	50.00	37.16	64.84	50.00	28.00	30.00	2.77	3793.20	1165.50	237100.00	0.52	02
	h_6	52.29	59.44	19.14	44.86	38.81	38.93	3.48	5080.10	4366.40	7572.60	0.78	03
	h_7	75.91	41.01	42.98	34.11	25.46	33.81	3.31	5543.60	4868.20	4909.40	1.66	06
300	h_1	67.80	74.79	43.03	48.41	34.95	35.52	4.53	6744.70	5799.80	4854.30	0.72	08
	h_2	62.47	59.90	37.62	50.00	50.00	40.00	4.56	6731.40	5610.90	7324.60	2.09	01
	h_3	89.08	62.51	73.33	40.55	9.81	30.30	5.60	5174.70	4305.00	179630.00	2.10	10
	h_4	98.18	66.02	50.92	31.32	12.13	47.36	5.96	16479.00	12504.00	47659.00	0.54	20
	h_5	98.31	65.41	53.17	49.63	12.05	26.95	5.55	10244.00	9436.40	204970.00	0.48	13
	h_6	61.38	61.06	47.30	48.21	49.45	36.56	3.99	6642.70	5545.90	6961.60	0.83	12
	h_7	64.45	81.84	60.42	43.91	14.89	39.38	4.92	10184.00	9279.20	16354.00	1.32	13

A comparison of the best results of NOx emission in table 7 has been done with Lagrange method [428], PSO [429],SA[430,431],MBO [432], GOA algorithm [433], Artificial ecosystem based optimization (AEO)[434],Multi objective 4th chaotic function Artificial ecosystem based optimization (CAE04)[435] ,Quantum Particle swarm optimization (QPSO)[436] and Sine-cosine algorithm(SCA)[437]from the load demand ranging from 150MW to 300MW shows the efficiency of this algorithm with optimal results. Fig6.3 Shows the comparison of various algorithms for NOx emission at different load levels.

Table 6.7 Comparison of best results considering fuel cost and NOx emission with other algorithms.

		150 MW	**175 MW**	**200 MW**	**225 MW**	**250 MW**	**300 MW**
PCPSO	P1	50.00	50.00	50.00	50.00	52.29	61.38
	P2	20.37	20.12	27.38	32.32	59.44	61.06
	P3	17.73	22.27	19.51	16.94	19.14	47.30
	P4	25.72	33.03	19.06	42.81	44.86	48.21
	P5	10.52	13.58	10.49	28.01	38.81	49.45
	P6	14.22	15.65	30.02	29.37	**38.93**	36.56
	F_C $/hr	**2555.20**	**2883.50**	**2904.80**	**3723.10**	**5080.10**	**6642.70**
	E_{NOx} Kg/hr	**2253.30**	**2561.20**	**2532.50**	**3065.00**	**4366.40**	**5545.90**
Langrage[428]		2448.21	2604.88	3102.07	3798.38	----	----
PSO[429]		2424.60	2879.70	3373.20	3877.60	4376.40	5557.70
SA[430]		2379.50	2789.92	3285.64	3781.19	----	**----**

MBO[432]	2854.13	3325.34	3325.34	3811.17	----	-----
GOA[433]	2854.08	3325.31	3325.31	3819.56	----	-----
AEO[434]	2883.35	3362.89	3362.89	3758.10	----	-----
CAE04[435]	2847.56	3301.52	3301.52	3832.06	---	-----
QPSO[436]	2854.22	3324.62	3324.62	3820.14	----	-----
SCA[437]	2854.00	3325.33	3325.33	3820.28	-----	-----

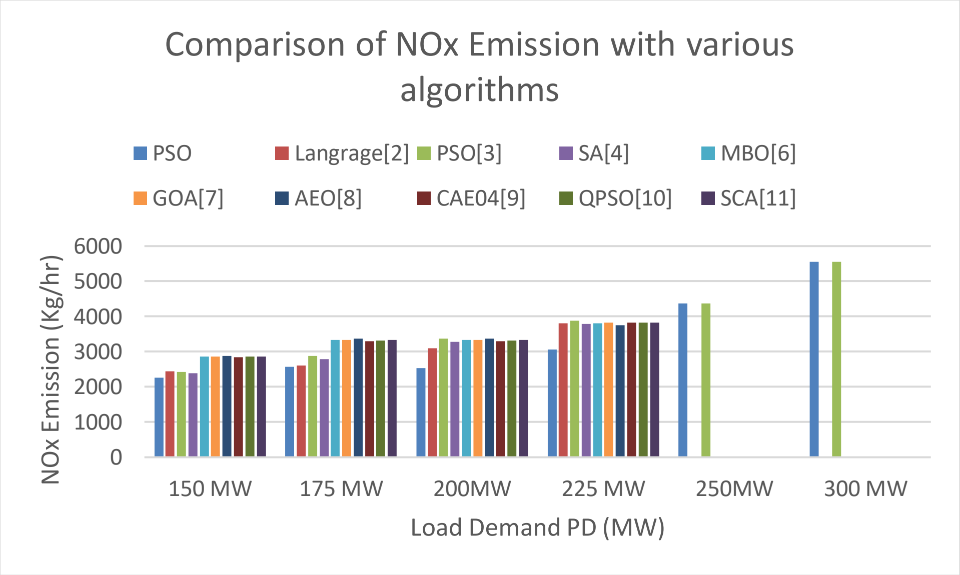


Fig6.3 Shows the comparison of various algorithms for NOx emission at different load levels. Following table6.8 Considers the bi objective function with CO2 using equation33 to calculate the $F_{T\,CO2}$ value from load demand 150 MW to 300 MW using seven price penalty factors. Among 150MW and 175 MW level, results from new average and common price penalty factors shows remarkable performance in calculating fuel cost, CO2 emission and the bi objective cost. At load demand of200MW using max-max price penalty h_2 it is compared with PSO [429] with a lowest fuel cost, lower CO2 pollutant and comparable bi objective value with PSO [429].But with a load level of 225MW and 250 MW, average and new average price penalty factor resulted in excellent results. When the load demand is 300MW, there is a steep increase in fuel cost and CO2 emission as compared to 250 MW using $h_6$new average price penalty factor. Among all the price penalty factors, $h_6$new price penalty is the best followed with the following sequence for optimal solutions.

$$h_6 < h_1 < h_2 < h_7 < h_3 < h_5 < h_4$$

Table6.8. $F_{T\,CO2}$ Results by PCPSO considering only fuel cost with CO2 emission and comparison with various algorithms on seven price penalty factors.

P_D (MW)	PPF	P_1 MW	P_2 MW	P_3 MW	P_4 MW	P_5 MW	P_6 MW	P_L MW	F_C $/h	E_{CO2} Kg/h	$F_{T\,CO2}$ $/h	CT Sec	Iter
150	h_1	50.00	33.21	18.79	13.19	19.15	15.66	1.33	2726.90	2377.10	1867.50	0.51	12
	h_2	50.00	25.10	16.71	21.06	14.26	22.87	1.26	2593.10	2411.80	5371.90	0.48	14
	h_3	50.00	52.15	49.57	34.13	7.99	7.70	2.38	2857.30	2559.70	5599.50	0.63	16
	h_4	53.07	12.58	8.35	31.10	5.18	41.50	1.79	4791.70	4401.30	23682.00	3.62	19
	h_5	52.09	19.18	16.17	8.03	5.16	51.30	1.96	4804.00	4573.50	15106.00	2.58	09
	h_6	50.00	20.37	17.73	25.72	21.96	14.22	1.12	2555.20	2247.60	2274.60	0.60	17
	h_7	50.00	24.82	18.61	26.27	21.28	21.73	1.41	2987.30	2798.30	3332.70	1.59	05
175	h_1	54.48	22.46	24.34	31.30	21.62	22.36	1.57	3346.60	3136.90	2805.20	5.49	13
	h_2	50.00	31.04	16.02	30.76	17.18	28.45	1.52	2844.40	2670.70	6897.30	9.10	20
	h_3	50.00	22.58	19.80	31.00	20.69	30.93	1.27	2898.40	2596.30	2853.00	0.73	18
	h_4	56.28	28.66	9.75	31.63	5.34	45.69	2.37	4384.70	4073.80	43062.00	0.53	13
	h_5	50.00	43.28	19.46	12.77	5.46	47.21	2.36	4051.90	3708.70	14724.00	0.61	20
	h_6	54.00	30.18	15.20	25.10	19.22	33.16	1.89	3295.00	3242.60	6613.90	0.53	08
	h_7	50.00	22.44	16.25	38.33	20.37	30.70	1.75	3252.90	3174.40	5137.70	1.06	06
200	h_1	52.02	32.29	23.31	32.10	28.42	33.88	2.06	3828.50	3740.10	4870.70	0.52	20
	h_2	50.00	27.38	19.51	19.06	10.49	30.02	1.50	2904.80	2722.90	7345.50	0.59	11
	h_2 **[429]**	53.20	29.64	21.91	33.62	30.49	31.31	---	3745.38	3697.11	7011.27	--	--
	h_3	52.75	73.34	34.29	17.45	9.55	13.52	2.91	4667.10	4247.70	3264.00	0.71	10
	h_4	50.00	33.47	43.69	30.38	11.14	31.32	2.94	2934.10	2678.70	12818.00	0.48	11
	h_5	50.00	31.21	32.95	26.13	5.28	55.37	2.89	4004.80	3787.30	16143.00	0.52	12
	h_6	54.46	36.49	22.05	30.46	22.17	36.59	2.23	3870.40	3774.40	7514.50	0.45	05
	h_7	50.42	50.69	22.38	32.57	10.27	36.06	2.41	3962.00	3689.80	6462.30	0.80	11
225	h_1	61.98	22.37	29.25	41.27	49.17	23.38	2.44	4673.00	4658.00	2964.50	1.77	16
	h_2	50.00	36.87	16.94	45.81	36.01	39.37	2.25	3723.10	2527.50	4464.30	0.57	06
	h_3	58.47	65.38	43.85	37.08	9.71	13.52	3.03	3915.20	3740.10	8501.90	3.07	18
	h_4	50.00	67.27	36.06	32.27	13.20	26.07	3.13	3582.80	3165.10	20155.00	0.46	16
	h_5	50.00	21.56	59.03	41.98	27.63	25.24	2.40	3700.20	3294.60	6787.10	0.52	02
	h_6	50.38	50.13	27.25	34.81	25.71	37.35	2.62	4472.00	5067.50	7720.40	0.55	19
	h_7	50.00	30.30	37.84	41.16	48.29	30.09	2.33	4796.60	4682.70	4519.80	1.80	08
250	h_1	58.57	36.72	30.90	48.23	45.10	33.41	2.95	5159.90	5093.70	4631.80	0.90	15
	h_2	50.00	43.26	31.11	48.78	40.41	36.44	2.42	4552.30	4489.50	9283.60	1.87	14
	h_3	50.00	22.49	80.00	50.00	27.50	30.00	2.95	4974.00	4713.00	8797.10	0.53	13
	h_4	50.00	29.70	73.10	49.46	27.74	30.00	2.84	3791.80	3421.10	17610.00	0.56	05
	h_5	50.00	37.16	64.84	50.00	28.00	30.00	2.77	3793.20	3394.40	14413.00	0.52	02
	h_6	52.29	59.44	19.14	44.86	38.81	38.93	3.48	5080.10	5069.70	8067.30	0.78	03
	h_7	75.91	41.01	42.98	34.11	25.46	33.81	3.31	5543.60	5369.80	5278.70	1.66	06

300	h_1	67.80	74.79	43.03	48.41	34.95	35.52	4.53	6744.70	6602.50	5010.90	0.72	08
	h_2	62.47	59.90	37.62	50.00	50.00	40.00	4.56	6731.40	6591.70	7998.30	2.09	01
	h_3	89.08	62.51	73.33	40.55	9.81	30.30	5.60	5174.70	4721.90	9040.90	2.10	10
	h_4	98.18	66.02	50.92	31.32	12.13	47.36	5.96	16479.00	20013.00	23855.00	0.54	20
	h_5	98.31	65.41	53.17	49.63	12.05	26.95	5.55	10244.00	10987.00	12576.00	0.49	13
	h_6	61.38	61.06	47.30	48.21	49.45	36.56	3.99	6642.70	6514.70	7427.60	0.83	12
	h_7	64.45	81.84	60.42	43.91	14.89	39.38	4.92	10184.00	10893.00	2218.10	1.32	13

Following table 6.7 Shows the comparison of the best results achieved using PCPSO with best price penalty factors for the fuel cost and CO2 emission with Lagrange method [428],PSO [429],SA[430,431],MBO [432],GOA algorithm [433], Artificial ecosystem based optimization (AEO)[434],Multi objective 4th chaotic function Artificial ecosystem based optimization (CAE04)[435] ,Quantum Particle swarm optimization (QPSO)[436] and Sine-cosine algorithm(SCA)[437].CO2 emission at all the load levels clearly shows it is lowest with better fuel cost make this algorithm fast, robust and efficient for finding the optimal solutions with low computational time.

Table6.9 Comparison of best results considering fuel cost and CO2 emission with other algorithms.

		150 MW	**175 MW**	**200 MW**	**225 MW**	**250 MW**	**300 MW**
PCPSO	P1	50.00	50.00	50.00	50.00	50.00	61.38
	P2	20.37	31.04	27.38	21.56	22.49	61.06
	P3	17.73	16.02	19.51	69.03	80.00	47.30
	P4	25.72	30.76	19.06	41.98	50.00	48.21
	P5	10.52	17.18	10.49	27.63	27.50	49.45
	P6	14.22	28.45	30.02	25.24	30.00	36.56
	F_C \$/hr	**2555.20**	**2844.40**	**2904.80**	**3700.20**	**4974.00**	**6642.70**
	E_{CO2} Kg/hr	**2247.60**	**2670.70**	**2722.90**	**3294.60**	**4713.00**	**6514.70**
Langrage[428]		2537.12	3613.53	4473.36	5502.52	----	----
PSO[429]		2607.10	3178.00	3771.50	4403.00	5027.20	6591.40
SA[430]		2568.94	3094.68	3714.33	4344.30	---	----
MBO[432]		2564.57	3129.20	3715.68	4328.07	----	----
GOA[433]		2564.50	3129.07	3715.56	4322.26	-----	----
AEO[434]		2480.41	3136.05	3719.47	4255.70	------	----
CAE04[435]		2448.86	3154.93	3725.20	4322.68	-----	----
QPSO[436]		2564.82	3129.91	3715.45	4322.70	----	-----
SCA[437]		2564.56	3129.18	3715.63	4322.52	----	----

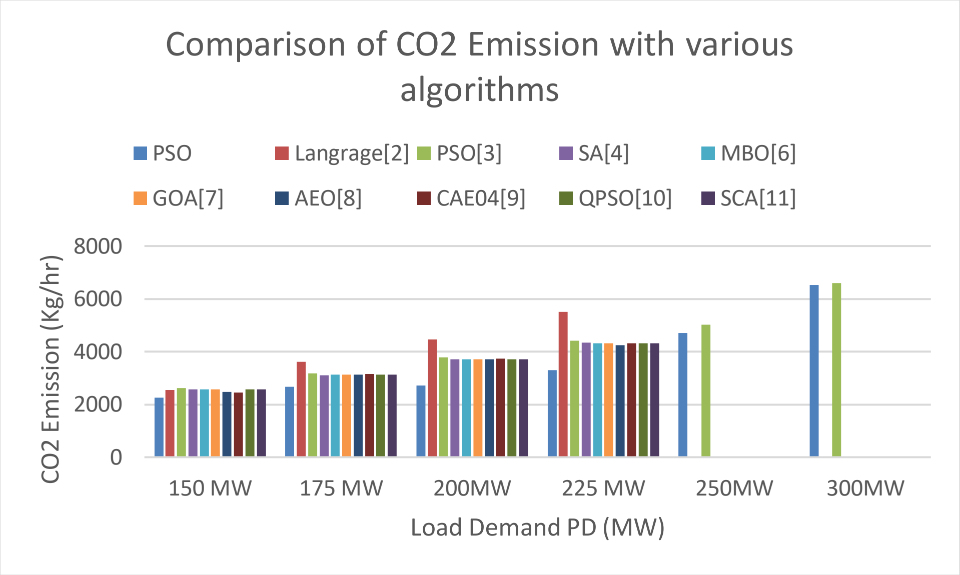


Fig6.4 Shows the comparison of CO2 emission for various algorithms at different load levels.

6.4 Conclusion

The combined economic and emission problem is formulated using cubic cost and cubic emission fuctions.PCPSO is developed to solve four-objective and bi objective optimization problem using seven price penalty factors for IEEE 30 bus,6 generator system for demand load from 150MW to 300MW.Following are the conclusions regarding the fuel cost,individual pollutants,total emission ,total cost and computation time.

(a) The total cost $F_{T\,CEED}$ for four objective is minimum by using the h_1 min-max price penalty fuction whereas in case of $F_{T\,SO2}, F_{T\,NOx}, F_{T\,CO2}$ is minimum with the help of h_6 new average price penalty function for bi objective function taking one emission at a time.

(b) The fuel cost F_C for CEED optimization problem is minimum by using h_2 max-max price penalty fuction wheras for bi objective with SO2 and NOx ,fuel cost is minimum with h_2 max-max price penalty fuction but with CO2 it is lowest by using h_6 new average price penalty function.

(c) Regarding emissions E_{SO2} and E_{NOx} , $h_1$min-max and h_2 max-max price penalty yields lowest whereas E_{CO2} gives minmum with h_7 common price penalty function.

(d) The total emission E_T is minimum by using $h_1$min-max price penalty factor.

(e) The computational time is minmum by using h_6 new average price penalty fuction which makes intelligent PSO very fast ,accurate and robust.

A conclusion can be made from the comparison with Langrage relaxation, PSO, SA, GOA algorithm, MBO, AEO, CAE04, QPSO and SCA the intelligent PSO outperforms in solving multi-objectives and bi objective power system optimization problems efficiently with optimal results in very small computational time with low iteration.

CHAPTER 7

MULTI AREA ECONOMIC EMISSION LOAD DISPATCH USING PERFECTLY CONVERGENT PARTICLE SWARM OPTIMIZATION

7.1 Introduction

Due to the effect of global environmental responsibility, the electrical power industry is moving toward development with energy savings and reduced emissions. The country's power facilities are dispersed across the country in order to compete for power reliability and best dispatch. With the right placement of producing stations around the region and the creation of small area zones in between, the concentration of harmful gaseous emissions was stabilized. Different parts of electrical power networks are integrated to increase operational efficiency, dependability, and lower total operating costs. Through tie lines, the places are connected to one another. The problem of multi-area economic dispatch (MAED) directly influences system, which includes diverse locations and tie lines. Each region in MAED has its own generator characteristics and load demand, which are connected via tie lines. If the load in any region varies, all of the generators cover it collectively with an altered power flow in the tie lines. Shoults et al. [446] solved the economic dispatch problem by taking into account inter-area import and export limits. This research presents a comprehensive definition of multi-area generation scheduling as well as an approach for multi-area research. The Dantzig–Wolfe decomposition principle was used for the restricted economic dispatch of multi-area systems by Romano et al. [447]. Multi-area economic dispatch with area control error was addressed by Helmick et al. [448]. Wang and Shahidehpour [449] suggested a decomposition strategy for leveraging expert systems to solve multi-area generation scheduling with tie-line restrictions. Streiffert [450] proposed network flow approach for handling the many-area economic dispatch problem with transmission restrictions. The Hopfield neural network technique was used by Yalcinoz and Short [451] to handle multi-area economic dispatch problems. Jayabarathi et al. [452] used evolutionary programming to tackle multi-area economic dispatch problems with tie line limitations. Sharma et al. [453] compared basic PSO and DE techniques, as well as their modifications, for handling the reserve restricted multi-area economic dispatch problem with power balancing restrictions, upper/lower generation limits, ramp rate limits, transmission constraints, and other practical constraints. To overcome the MAED problem, a Karush Kuhun Tucker (KKT) optimality criterion was used to guarantee optimal convergence in a covariance matrix adapted evolutionary strategy for MAED problems [454]. To tackle the reserve constrained ED problem with prohibited operation zones, Lee and Breipohl [455] used a decomposition technique (POZ). The reserve restricted dynamic dispatch problem was modeled using a hybrid technique combining particle swarm optimization (PSO) and sequential quadratic programming (SQP) in Ref. [456]. A PSO-based method for the reserve constrained multi-area environmental/economic dispatch problem was recently developed by Wang and Singh [457]. In the multi-area power market dispatch problem, a limited PSO technique is presented to cope with both energy and reserve allocation [458]. Due to their simplicity, lack of convexity assumptions, and great random parallel search capacity, evolutionary optimization approaches are rapidly being suggested for ED issues with quasi cost functions. Tabu search, simulating annealing, neural networks, genetic algorithm , particle swarm optimization , harmony search , ant colony optimization , bacterial foraging , artificial immune system, and differential evolution (DE) are some of the methods mentioned in various heuristic optimizing strategies for tackling ED problems can be found in ref. [459]. Because of their dependability, durability, speed of convergence, minimal information required, and simplicity of application, PSO and their versions have become more common. With the introduction of meta-heuristic methodologies, the focus has switched to the use of such innovation approaches to deal with the complexity of real-world scenarios. Many scholars have focused on meta-heuristic strategies because of its potential to generate a closer optimal solution. This optimization problem is tackled utilizing perfectly convergent particle swarm optimization to achieve a better overall Pareto-optimal solution. Krishnamurthy and Tzoneva [428] defined price penalty factor (PPF) is defined as the ratio of fuel cost to emission value with several strategies such as Min-Min, Max-Max, Max-Min, and Min Max. The proposed approach has been successfully tested on a single area, two area, three areas and four-area, forty generator system and twelve-generator system with and without tie lines.

7.2 Problem overview and its Formulation

7.2.1 Single area economic emission power dispatch (SAEEPD)

This section covers the problem formulation for two different sorts of ED problems. The single objective environmental/economic dispatch (EED) problem is an outgrowth of the ED problem that includes environmental aspects. The EED challenge is expanded to include power dispatching among multi-area environmental/economic dispatch (MAEED) by reducing operational expenses and pollutant emissions while dispatching power over several zones. When a large turbine generator is called upon to increase production, a number of fuel entry ports are typically opened one after the other. When a valve is opened, throttling losses increase quickly, causing the incremental heat rate to climb abruptly. The valve-point effects cause disturbances in the heat-rate curves, and the objective function becomes disjointed, non-convex, and has many minima. The operating cost of single area thermal power plants is described as follows:

7.2.2 Multi area economic emission power dispatch (MAEEPD) using PCPSO

The operating cost of multi area thermal power plants using PCPSO [401] is described as follows:

$$Min\ F_C = \sum_{m=1}^{M} \sum_{n=1}^{N} a_{mn} P_{mn}^2 + b_{mn} P_{mn} + c_{mn} \ + \left| \alpha_{mn} \sin\left(\beta_{mn}\left(P_{mn,min} - P_{mn}\right)\right) \right| \ \$/h \tag{7.1}$$

The emission of the multi area thermal power plant with valve point loading effect is as follows:

$$E_T = \sum_{m=1}^{M} \sum_{n=1}^{N} (d_{mn} P_{mn}^2 + e_{mn} P_{mn} + f_{mn}) + \gamma_{mn} exp(\delta_{mn} P_{mn}) \quad Kg/h \tag{7.2}$$

$$F_{T\,CEED} = \sum_{i=1}^{n} \left[\left[a_{mn} P_{mn}^2 + b_{mn} P_{mn} + c_{mn} + \left| \alpha_{mn} \sin\left(\beta_{mn} (P_{mn,min} - P_{mn}) \right) \right| \right] + h_{mn} \left[(d_{mn} P_{mn}^2 + e_{mn} P_{mn} + f_{mn}) + \gamma_{mn} exp(\delta_{mn} P_{mn}) \right] \right] \; \$/h \tag{7.3}$$

The operating cost for transmission of power through tie lines is as follows:

$$F_{TL}(P_{TL}) = \sum_{m=1}^{M} \sum_{\substack{i=1 \\ i \neq m}}^{M} (q_{mi} P_{TLmi} + q_{im} P_{TLim}) \quad \$/\mathrm{h} \tag{7.4}$$

Subject to constraints:

Generator limit constraint: The minimum and maximum real power produced for every committed generating unit should be within following limit.

$$P_{mn,min} \ll P_{mn} \ll P_{mn\,max} \tag{7.5}$$

Tie- line limit constraint: The real power send through the tie lines of the committed generating unit are valid for both directions and be within following limit.

$$P_{TLmi,min} \ll P_{TLmi} \ll P_{TLmi,max} \tag{7.6}$$

Power balance constraint: The total real power generation is equal to the sum of total power demand for the area $m^{th}$and there transmission losses.

$$\sum_{i=1}^{n} P_{mn} = P_{Dm} + P_{Lm} + \sum_{\substack{i=1 \\ i \neq m}}^{M} [P_{TLmi} - (1 - \rho_{im}) P_{TLim}] \tag{7.7}$$

Where

$$P_{Lm} = \sum_{n=1}^{N} \sum_{j=1}^{N} \left(P_{mn} B_{mnj} P_{mj} \right) + \sum_{n=1}^{N} B_{0n} P_{mn} + B_{m00} \tag{7.8}$$

7.3 **Implementation of PCPSO in MAED**

Step1. Set the lower and higher bounds for each unit's generation, as well as the area load demand and tie line transfer limits.

Step2. For a population size S in the j^{th}-dimensional space, generate particles at random between the maximum and minimum operational limits of the N units and M number of tie lines with $i^{th}\ particle$ as P_i = $[(P_{i1}^{n}, P_{i2}^{n} \ldots\ldots P_{iN}^{n}, T_{i1}^{n}, T_{i2}^{n} \ldots\ldots T_{iM}^{n})]$ where i=1, 2...S.

To satisfy the generation limit requirements given by (7.5) and tie-line limits given by (7.6), r is a uniformly distributed random number between 0 and 1 in this situation.

$$P_{ij}^{n} = P_{min} + r\left(P_{ij\,max} - P_{ij\,min}\right) \tag{7.9}$$

Step3.Constraints as a result of prohibited operating zones

If any element P_{ij} of the starting population (or updated population) is found to be within the kth forbidden operating zone, it is modified and allocated the generation value corresponding to the zone's lower (P_{ij}^{lower}) or higher (P_{ij}^{upper}) boundary, according to the specified logic. The midpoint of the kth forbidden zone is $P_{mid,k}$.

$$P_{ij} = \begin{cases} P_{ij}^{lower} & if\ P_{ij}^{lower} \leq P_{ij} < P_{mid,k} \\ P_{ij}^{upper} & if\ P_{mid,k} \leq P_{ij} < P_{ij}^{upper} \end{cases} \tag{7.10}$$

Step4. In N-dimensional space, generate particle velocity in the range $[v_i^{min} v_i^{max}]$.

Step5. Calculate the fitness of each individually using the equation (7.1-7.4).

Step6. To increase fitness, the parameters are adjusted iteratively. Equations (6.13-6.15) are used to update the parameters in PSO.

Step7. For the revised positions of the particles, the evaluation function values are determined. If the new value is better than the existing pbest, it is set to pbest in PSO. Similarly, the value of gbest is modified to reflect its status as the best vector among pbest.

Step8.In stopping criterion, if the equation (17) is less than stagnation threshold ε = $1\ X\ 10^{-6}$ the position of particles is represented as Gbest for the optimal solution and stop.

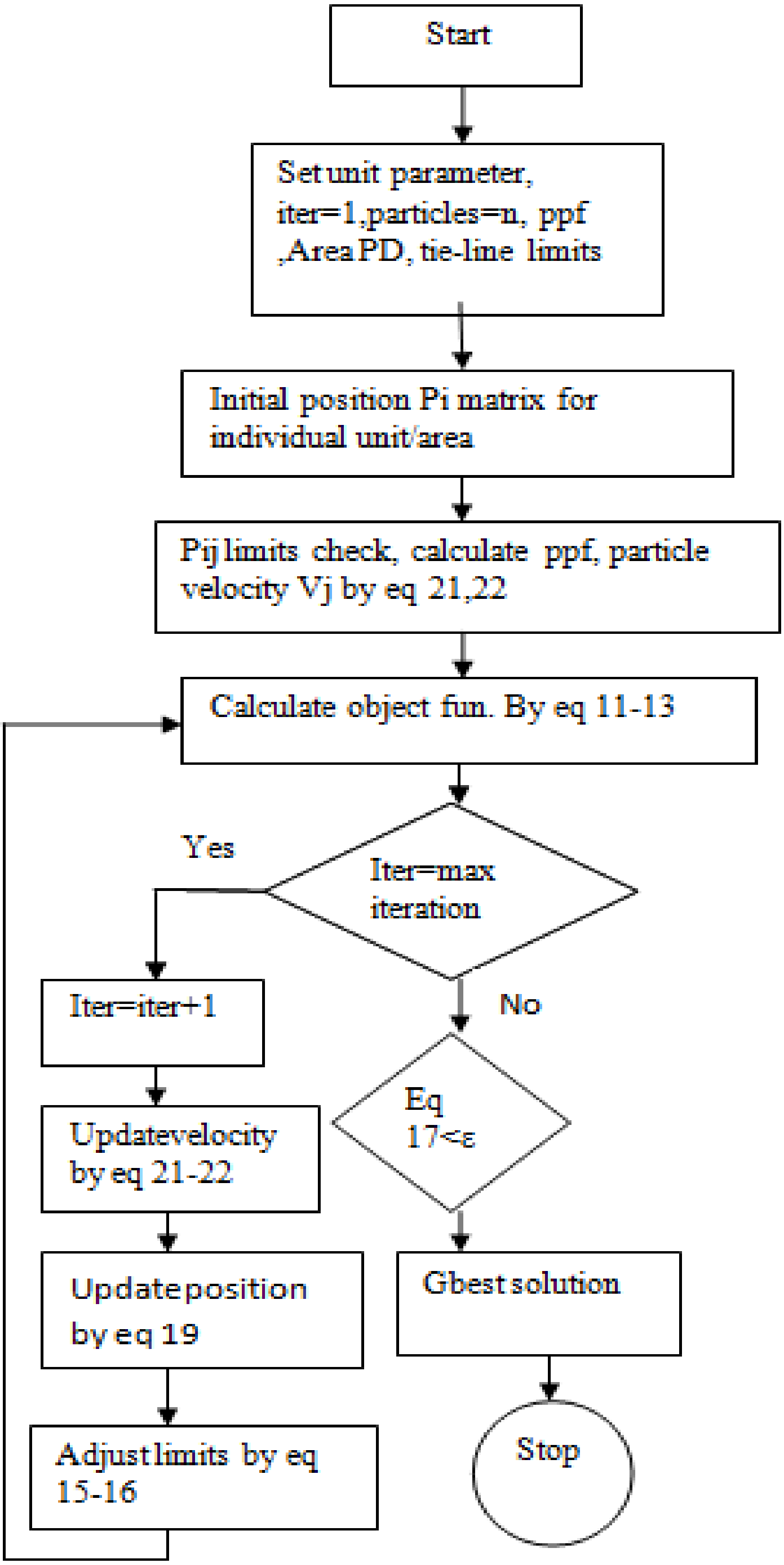

Start
Set unit parameter, iter=1,particles=n, ppf ,Area PD, tie-line limits
Initial position Pi matrix for individual unit/area
Pij limits check, calculate ppf, particle velocity Vj by eq 21,22
Calculate object fun. By eq 11-13
Yes
Iter=max iteration
Iter=iter+1
No
Updatevelocity by eq 21-22
Eq 17<ε
Update position by eq 19
Gbest solution
Adjust limits by eq 15-16
Stop

Fig7.1 Flow chart of PCPSO for Multi area economic emission dispatch

7.4 Results and Discussion

The MALD problem is substantially more complex and difficult to answer than the standard ED problem because of the extra tie-line constraints and area power balancing constraints. On three test systems with different sizes and nonlinearities, the PCPSO [401] and other techniques are tested for the suggested practical MALD problem. The results were compared to those reported before and found PCPSO to be superior.

The suggested algorithm in one case is tested on single, two and four areas with forty generators connected by six tie lines shown in test case 1 to 3 and another case with one ,two, three and four areas with twelve generators with six tie lines to see the effectiveness and robustness of PCPSO. Number of particles in swarm is 20, number of iterations are 250, number of trails is 5, linearly decreasing inertia weight with minimum and maximum inertia wmin=0.4, wmax = 0.9, acceleration constants c1 = c2 =2 are some of the parameters of proposed PCPSO.

7.4.1 Problem A: 40 generating unit test system

Test Case1: Single area, 40 unit test system

The realistic Tai power system, which is a large-scale and mixed generating system with coal-fired, oil-fired, gas-fired, diesel, and combined cycle all present, has 40 generating units [460] placed in a single area without tie lines and transmission losses as shown in appendix . The system's load demand is 10500MW with the valve point loading (VPL) effect, the ramp rate limit (RRL), and prohibited operating zones (POZ), non smooth cost function, emission function. At this load demand optimal simulation results of PCPSO [401] are compared with DE [408] MODE [408], and NSGA-II [461] respectively. Fuel cost and emission cost using PCPSO is 121370.00 $/h and 72403.00ton/h which is lower than DE[408] ,MODE [408] and NSGA-II[461] showing better convergence characteristics in obtaining optimal values in just 0.025 seconds as shown in table7.1.

Table7.1.Optimal results shown by PCPSO as compared with MODE, DE, NSGA-II algorithms

Power Output (MW)	**PCPSO**	**MODE [408]**	**DE [408]**	**NSGA-II [461]**	**Power Output (MW)**	**PCPSO**	**MODE [408]**	**DE [408]**	**NSGA-II [461]**
P_1	114	113.5295	110.9515	113.8685	P_{21}	550	434.6068	524.5336	434.6639
P_2	114	114	113.2997	113.6381	P_{22}	550	434.531	526.6981	434.15
P_3	94.2474	120	98.6155	120	P_{23}	550	444.6732	530.7467	445.8385
P_4	158.8569	179.8015	184.1487	180.7887	P_{24}	550	452.0332	526.327	450.7509
P_5	97	96.7716	86.4013	97	P_{25}	550	492.7831	525.6537	491.2745
P_6	118.9533	139.276	140	147	P_{26}	550	436.3347	522.9497	436.3418
P_7	300	300	300	300	P_{27}	15.9062	10	10	11.2457
P_8	300	298.9193	285.4556	299.0084	P_{28}	15.9062	10.3901	11.5522	10
P_9	300	290.7737	297.511	288.889	P_{29}	15.9062	12.3149	10	12.0714
P_{10}	130.00	130.9025	130	131.6132	P_{30}	97	96.905	89.9076	97
P_{11}	227.2476	244.7349	168.7482	246.5128	P_{31}	190	189.7727	190	189.4826
P_{12}	223.8508	317.8218	95.695	318.8748	P_{32}	190	174.2324	190	174.7971
P_{13}	308.7384	395.3846	125	395.7224	P_{33}	190	190	190	189.2845
P_{14}	327.2448	394.4692	394.3545	394.1369	P_{34}	90	199.6506	198.8403	200
P_{15}	326.7523	305.8104	305.5234	305.5781	P_{35}	90	199.8662	174.1783	199.9138
P_{16}	326.7523	394.8229	394.7147	394.6968	P_{36}	90	200	197.1598	199.5066
P_{17}	455.9423	487.9872	489.7972	489.4234	P_{37}	110	110	110	108.3061
P_{18}	457.6355	489.1751	489.362	488.2701	P_{38}	110	109.9454	109.3565	110
P_{19}	501.3522	500.5265	520.9024	500.8	P_{39}	110	108.1786	110	109.7899
P_{20}	501.3546	457.0072	510.6407	455.2006	P_{40}	501.3522	422.0682	510.9752	421.5609
Total Fuel cost($/h)						**121370.00**	125790.00	121840.00	125830.00
Total Emission (ton/h)						**72403.00**	211190.00	374790.00	210950.00
Computational time (s)						**0.025**	5.39	13.25	7.32

7.4.2 Test case2: two area, 40 unit test system.

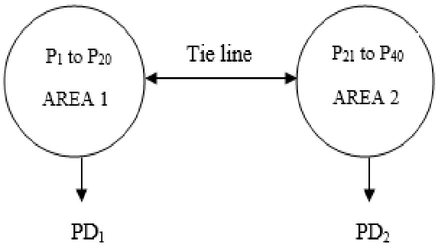


Fig 7.2. Shows the two area network topology with 20 units in each area.

It consists of two areas with forty [460] power producing units. The operating limitations, tie line limits, and fuel cost characteristics data are comparable to [408]. This test scenario includes all of the practical complexity, such as the valve point loading (VPL) effect, the ramp rate limit (RRL), and prohibited operating zones (POZ), making the system extremely complex and non linear. The entire system is comprised of two groups, each with 20 generating units and connected by a tie line. The power demand for areas 1 and 2 is set at 7500MW and 3000MW, respectively, while the total power demand is 10500MW. The transmission line flow limit is set at 1500MW for modeling purposes. Figure 2 depicts the two-area network topology. For this test instance, PSO simulations were used for five trails. Table7.2 shows the best result achieved using PCPSO[401] and is compared with DE algorithm with chaotic sequences based on logistic map (DEC2)[453],Hybridizing sum-local search optimizer(HSLSO)[463] and MFO[464]. Following Fig7.1shows the network topology of two area 40 unit tests system

Table7.2 shows the comparison of PSO, DEC2, HSLSO and MFO for total fuel cost, total emission and computational time.

Area1					**Area2**				
Power Output (MW)	**PCPSO**	**DEC2[453]**	**HSLSO[463]**	**MFO[464]**	**Power Output (MW)**	**PCPSO**	**DEC2[453]**	**HSLSO[463]**	**MFO[464]**
P_1	114	112.8292	110.8012	114	P_{21}	550	343.7598	523.2792	523.2794
P_2	114	114	113.9997	114	P_{22}	550	433.5196	523.2791	523.2794
P_3	120	97.3999	120	120	P_{23}	550	523.2794	523.2794	433.5196
P_4	190	179.7331	179.7331	179.7331	P_{24}	550	550	523.2794	523.2794
P_5	97	97	95.551	97	P_{25}	550	550	523.2795	433.5196
P_6	140	68.0001	140	140	P_{26}	245.161	254	254	10
P_7	275.1152	300	300	300	P_{27}	10	10	10.0001	10
P_8	300	284.5997	284.5997	284.9652	P_{28}	10	10.001	10	10
P_9	300	284.5997	284.5997	284.9475	P_{29}	10	10	10	
P_{10}	300	130	270	270	P_{30}	97	47	87.7997	87.8007
P_{11}	136.5160	360	94	168.7998	P_{31}	75.9463	159.7331	188.5959	165.8263
P_{12}	252.9017	94.0001	300	168.7998	P_{32}	75.9463	190	159.7331	166.2398
P_{13}	291.5930	304.5196	304.5195	394.2794	P_{33}	75.9463	163.7269	159.733	161.0328
P_{14}	395.3015	500	394.2797	394.2794	P_{34}	90	164.7998	164.8002	164.7998
P_{15}	436.7864	484.0392	484.0395	484.0392	P_{35}	90	200	164.7998	164.7999
P_{16}	436.7864	500	484.0391	484.0392	P_{36}	90	164.7998	164.7998	90
P_{17}	500	489.2794	489.2794	489.2794	P_{37}	110	110	89.1143	89.1143
P_{18}	500	500	489.2795	489.2794	P_{38}	110	57.0571	89.114	89.1151
P_{19}	550	550	549.9998	511.2794	P_{39}	110	25	89.1134	89.1147
P_{20}	550	550	511.2791	511.2794	P_{40}	550	511.2794	242.0001	331.7598
Tie line Power (MW) T_{12}						-1500	-1500	-1500	-1500
Total Fuel cost($/h)						**120200.00**	127344.8528	125100.2621	124746.1
Total Emission (ton/h)						**103400.00**	---	---	---
Computational time (s)						**10.2793**	---	---	---

For a tie line limit of 1500MW, PCPSO outstanding results in terms of operational costs is $/h120200.00 and emission of 103400.00 as compared to other recently reported methods DEC2[453], HSLSO[463] and GFO[464] in a very small computational time showing excellent converging behavior without getting trapped in local minima . PCPSO [401] outperforms other methods in finding the best solution for a fairly big complex enclosed power system optimization issue.

7.4.3 Test Case-3: Four area, 40 unit test system

There are forty generating units[460] in this test system, each with fuel and emission coefficients, real power limits, tie–line limits, ramp rate limit, prohibited operating zones (POZ), and without transmission loss. The overall demand for these areas is 10500MW, which is shared by the following areas: area1 demand is 1575MW (15%), area2 demand is 4200MW (40%), area3 demand is 3150MW (30%), area4 demand is 1575MW (15%), and the tie-line power flow limit between area 1 and area 2 is 200MW. The maximum tie-line power flow between areas 1 and 3 or from area 3 to area 1 is 200MW. The tie-line power flow limit is 200MW between area 3 and area 2, or from area 2 to area 3. A tie-line power flow limit of 100MW exists between area 4 and area 1 or from area 1 to area 4. A tie-line power flow limit of 100MW exists between area 4 and area 2 or from area 2 to area 4.Figure 7.3 depicts the network topology used in this test case scenario which has four area forty generating units.

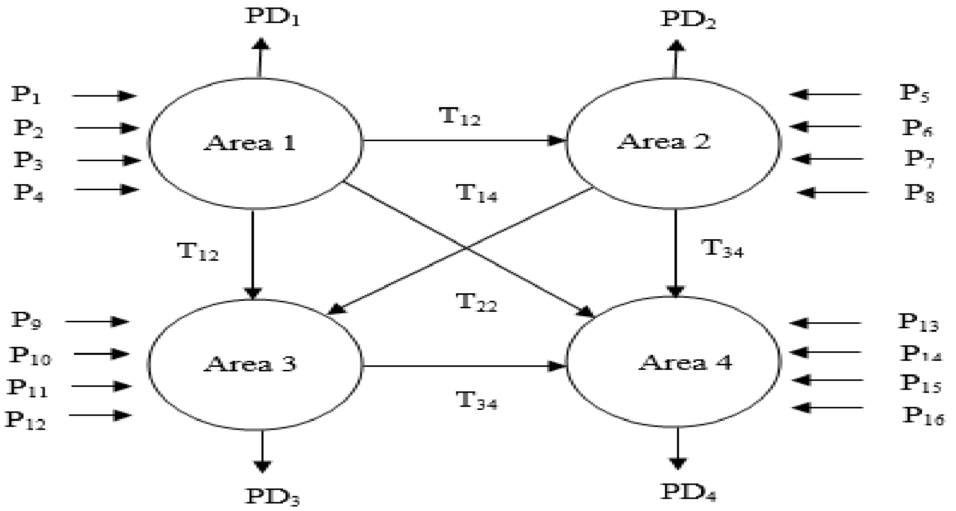


Fig 7.3 Shows the four area network topology with 10 units each area.

For load demand of 10500 MW, PCPSO[401] results in total fuel cost of 123220 $/hr and total emission 95171ton/h which is very low as compared to SSA-WSA[465]and

MOSSA[465] with reasonable computational time of 63.98 seconds showing the optimal performance of this algorithm. Following table7.3 shows the comparison of PCPSO[401] with other heuristic methods for total fuel cost and total emission.

Table7.3Shows the four area 40 unit test system comparison of PCPSO with other heuristic methods for total fuel cost, total emission and computational time.

Power Output (MW)	**PCPSO**	**SSA-WSA [465]**	**MOSSA [465]**	**Power Output (MW)**	**PCPSO**	**SSA-WSA [465]**	**MOSSA [465]**
Area1				**Area3**			
P_1	90.2165	111.8462	110.9538	P_{21}	555.4919	432.7510	434.5725
P_2	90.2165	111.5728	110.8816	P_{22}	507.0144	432.8681	434.4692
P_3	120	120	120	P_{23}	504.8487	467.9573	450.5827
P_4	134.7249	179.4692	179.7097	P_{24}	504.8487	432.9132	434.5625
P_5	97	96.5825	89.4135	P_{25}	470.3979	432.6851	434.3924
P_6	136.5832	139.9594	139.9382	P_{26}	475.1169	432.7357	434.3848
P_7	125.7200	298.9183	299.9273	P_{27}	10	10	10
P_8	292.1972	285.3418	284.6758	P_{28}	10	10	10
P_9	300	285.5432	284.6154	P_{29}	10	10	10
P_{10}	188.3417	130	130	P_{30}	97	89.5109	97
Area2				**Area4**			
P_{11}	179.2089	317.6829	318.3747	P_{31}	75.3283	150.4178	150.6829
P_{12}	195.8519	317.5074	318.5792	P_{32}	75.3283	190	189.8213
P_{13}	297.9388	394.8213	394.4819	P_{33}	75.3283	190	189.9506
P_{14}	490	394.4323	394.4273	P_{34}	115	193.3462	198.9723
P_{15}	490	394.7807	394.4847	P_{35}	90	200	200
P_{16}	490	394.4285	394.4019	P_{36}	90	200	200
P_{17}	490	487.5579	488.9985	P_{37}	110	110	108.6298
P_{18}	490	487.7379	488.9246	P_{38}	110	110	108.9851
P_{19}	538.5	420.8462	420.9319	P_{39}	110	110	108.4175
P_{20}	538.5	510.3568	512	P_{40}	550	415.4294	418.8564
T_{12}	50.0786	158.3422	145.5769	T_{41}	14.9774	100	100
T_{31}	-86.9819	-125.8912	-129.5384	T_{42}	13.7911	94.1934	99.3243
T_{32}	0.1470	-172.6875	-170.5059	T_{43}	16.7230	100	99.9916
Total Fuel cost($/h)					**123220.00**	125760.0557	125591.3223
Total Emission (ton/h)					**95171.00**	206705.9772	205965.4061
Computational time (s)					**63.98**	----	-----

7.4.4 Problem B: 12 units test system

The test system consist of 12 units with quadratic cost ,valve point loading, emission level function, unit data and loss coefficients[467] The proposed algorithm PCPSO[401] is applied to 12 generating unit system shown in appendix with a load demand of 2090MW in the following cases.

Single area load dispatch (SALD): The entire power system is viewed as a single area with twelve generators and no tie lines in between.

7.4.5 Two area load dispatch (TALD): The 12 unit test system is divided equally into two areas with six generators in each area having load demand of 1200MW and 890MW and one tie line of 1500MW in between them.

7.4.6 Three area load dispatch (THALD): The 12 unit test system is divided equally into three areas with four generators in each area having load demand of 627,627 and 836 MW with three tie-lines in between them.

7.4.7 Four area load dispatch (FALD): The test system has four areas with three generators in each area and six tie lines among them. The total load demand of all areas is 2090MW which is divided into 500MW, 410MW, 580MW, and 600 MW throughout the four regions with and without transmission losses taken into account. The tie-line power flow between areas 1 and 3 or from area 3 to area 1 is 5-60MW. The tie-line power flow limit is 5-60MW between area 3 and area 2, or from area 2 to area 3. A tie-line power flow limit of 5-60MW exists between area 4 and area 1 or from area 1 to area 4. A tie-line power flow limit of 5-60MW exists between area 4 and area 2 or from area 2 to area 4.

7.4.8 Test case5: 12 units test system with transmission losses

There are twelve generating units[467] in this test system, each with fuel and emission coefficients, real power limits, tie –line limits, ramp rate limit, prohibited operating zones (POZ), and with transmission loss. The overall demand for these areas is 2090MW, which is shared by the following areas: area1 demand is 500MW (24%), area2 demand is 410 MW (20%), area3 demand is 580MW (28%), area4 demand is 600MW (28%), and the tie-line power flow limit between area 1 and area 2 is 5-60MW. The tie-line power flow between areas 1 and 3 or from area 3 to area 1 is 5-60MW. The tie-line power flow limit is 5-60MW between area 3 and area 2, or from area 2 to area 3. A tie-line power flow limit of 5-60MW exists between area 4 and area 1 or from area 1 to area 4. A tie-line power flow limit of 5-60MW exists between area 4 and area 2 or from area 2 to area 4.

Following table7.4 shows PCPSO[401] is implemented in different types of areas,12 unit test system with the valve point loading (VPL) effect, the ramp rate limit (RRL), and prohibited operating zones (POZ), non smooth cost function, emission function. Transmission losses are neglected here. The load demand in single area is 2090MW, in two area system load in first area is 1200MW and in other area is 890MW.In third type of area it is 627,627 and 836 MW respectively. Results shows PCPSO[401] total fuel cost of 118310 $/h ,emission is 1653.40ton/h and CEED value using different penalty factors are very less as compared to SPSO[465] results in single area system. In two areas, three areas and four areas results are also remarkable in very small computational time.

Table7.4 Shows the comparison of single, two, three and four areas for12 unit test system using PCPSO with other heuristic methods for total fuel cost, total emission and computational time.

Power Output (MW)	SALD PCPSO	SALD SPSO[465]	TALD PCPSO	THALD PCPSO	FALD PCPSO
P_1	201.6515	210	210	210	117.3351
P_2	325	302.79	275.7630	150	188.4085
P_3	315	230.04	272.3978	175	182.2364
P_4	115.7411	150	150	161.7851	138.1334
P_5	110	110	110	325	110
P_6	131.8932	215	189.7268	110	155.8793
P_7	124.2892	120.04	175	215	165.6951
P_8	161.6722	135.04	215	154.9003	205.0949
P_9	146.8550	248.24	186.5159	315	195.8793
P_{10}	113.4169	120.04	175	125	149.8005
P_{11}	126.7259	135.04	183.3885	167.5345	166.8083
P_{12}	211.1506	331.12	184.4267	273.3862	283.8263
PT_{12}	---	---	176.3610	7.3170	-12.0347
PT_{13}	---	---	---	12.8168	-6.2398
PT_{14}	---	---	---	9.4675	-13.3574
PT_{23}	---	---	---	----	0.4060
PT_{24}	---	---	---	----	-6.0352
PT_{34}	---	---	---	----	3.4241
Fuel cost($/h)	**118310.00**	142545.03	140910.00	141210.00	127430.00

Emission cost(ton/h)		**1653.40**	2011.81	1962.50	2096.20	1499.60
CEED ($/h)	min-max	**144000.00**	172538.81	219960.00	170880.00	149650.00
	min-min	**232390.00**	377600.54	325080.00	116970.00	269490.00
	max-max	**211880.00**	264943.13	349890.00	117680.00	226760.00
	max-min	**353130.00**	1414109.9	291750.00	222460.00	672760.00
time (s)		**0.0741**	---	8.06	30.03	0.131

Results shown in following table7.5shows the comparison of four units in each four areas considering transmission losses in the network, the total fuel cost of PCPSO [401] is 133210.00 $/h which is less as compared to SPSO [465], PSO [465] and LDCM [466] with emission value, losses are comparable to SPSO but very less than PSO [465] and LDCM [466] showing the fast convergent characteristics of PCPSO in very small computational time. The CEED value with min-max, min-min, min-max price penalty is lowest as compared to other methods, except max-min price penalty factor.

Table7.5 Shows the comparison of four areas for 12 unit test system with transmission losses using PCPSO with other heuristic methods for total fuel cost, total emission and computational time.

Power Output (MW)	PCPSO	SPSO[465]	PSO[465]	LDCM[466]
P_1	148.88	165	160.9	131.45
P_2	150	260	163.3	209.49
P_3	175	265	289.4	202.25
P_4	175	150	109.6	150
P_5	182.74	75	117.1	110
P_6	110	175	184.4	191.63
P_7	215	145	171.5	175
P_8	164.49	160	197.5	215
P_9	176.0	280	198.3	236.38
P_{10}	156.30	145	230	164.09
P_{11}	197.10	160	144.8	180.2
P_{12}	287.36	280	211.5	306.18
PT_{12}	-3.6579	15.23870	77.94	9.9944
PT_{13}	0.5297	50	0	9.9944
PT_{14}	-5.4905	15.2380	18.35	9.9944
PT_{23}	7.9609	15.2380	46.99	9.9881
PT_{24}	0.5945	15.2380	8.45	9.9881
PT_{34}	11.7808	15.2380	12.25	9.9876

Power Loss P_L	**48.66**	46.66	48.22	61.82
Fuel cost($/h)	**133210.00**	139359.86	136598.3	144058.2
Emission cost	**1629.10**	1598.14	3713.93	1923.7
CEED Cost ($/h) for min-max	**154460.00**	232500.67	277258.2	266400.92
CEED Cost ($/h) for min-min	**279750.00**	393711.74	----	---
CEED Cost ($/h) for max-max	**241050.00**	494067.99	----	---
CEED Cost ($/h) for max-min	**675390.00**	263817.77	----	---

7.5 Conclusion:

PCPSO has been effectively implemented to tackle MAED problems in this article. Different types of test systems are used to demonstrate the usefulness of the suggested method, and the test results are compared to DE, MODE, NSGA-II, DEC2, HSLSO, MFO, SSA-WSA, MOSSA, SPSO, PSO and LDCM results. The suggested PCPSO has the ability to converge to a higher quality solution than other techniques, as evidenced by the comparison. The use of the perfect convergent particle swarm optimization (PCPSO) approach increases computational efficiency when compared to previous PSO variants or other heuristic techniques and achieves a speedy true global minimum, as per the results. By automatically generating a new particle mechanism, this programme assists and gives a chance to all particles who are experiencing premature convergence and stagnation. if the best particle changes its position each time for ρ(k) to readjust and stabilize. By assisting particles in premature convergence stages and allowing all particles to reach real global minima, PCPSO was able to provide balanced exploration and exploitation in the search space. This algorithm is a promising alternate technique to solve single area and multi area economic emission dispatch in real power system problems. The applications could be considered in the future as a way to improve system security, emission quota trading, transmission costs, and cross-area buying and selling laws, can be incorporated to reflect more realistic scenarios in MAED issues.

Chapter 8

Conclusion and future recommendations

8.1 Introduction

Traditional power dispatch issues overlook all errors as well as ambiguities present in real-world power system operations and assume all of the variables as deterministic ones. The CEED problem is presented in this thesis as a stochastic problem, where two components namely cost of fuel and pollution must be minimized concurrently with meeting the set constraints, like generating capabilities and power flow balance. The power balance constraint model accounts for transmission line losses. Instabilities such as the valve-point effect, POZ, and ramp rate limit are indeed taken into account. The multi-objective issue is more accurate when these nonlinearities are taken into account.

The CEED issue has multiple local extreme solutions due to such predominance of nonlinear and non-convex properties, making it difficult to find a global optimum. For CEED issues, the traditional multi-objective optimization techniques need not yield Pareto-optimal results.

8.2 Main conclusion

The electrical grid is a network of connected structures that transports electricity generated plants to loads. Several power stations exist geographically apart in the actual world, and thus the system is extremely complicated. Due to the numerous interconnections, restrictions, tie-lines, and loads that it is characterized by, the corresponding economic dispatch problem is likewise highly difficult. The complex network must be viewed as a collection of distinct but linked locations in order to comply with the standards of deregulation. Consequently, the challenge has a new framework for the optimal allocation, which in this case is exploring for two approaches: solutions for the individual areas and solutions for the entire system as a collection of the area's solutions. Following are the conclusions made chapter wise to show the performance of PCPSO for solving CEED optimization problems.

1. Literature survey of PSO suggests the developments like historical development, addition of new parameters, tuning or refinement of parameters and its variants for different optimization problems with constraints, multi-objectives PSO, parallel PSO, its hybrids, communication topology and for multi-objective problems strategy used for parallel computing is covered in detail. The new variant PCPSO [401] is developed after doing intensive detailed study on theoretical analysis of existing and new parameters, exploration, particle diversity, premature convergence and stagnation to avoid trapping in the local minima's.
2. Results were also examined to train the PSO variants (OPSO with Cauchy mutation & PCPSO) using Back Propagation algorithms (Lavenberg-Marquardt and Bayesian Regularization) on the bench mark functions in 30 dimensions. By changing number of neurons in hidden layer from 4,8,12 learning by Lavenberg-Marquardt algorithm showed remarkable accuracy to train. ANN is able to train uni modal, multi modal with many minima as well as noisy functions with very low (near to zero) using PSO. The simulation results shows that Lavenberg-Marquardt algorithm is more efficient for uni-modal functions and Bayesian Regularization out performs for multi modal /noisy functions. The training of ANN using PCPSO [401] is very fast and has efficient learning rate when high precision is required.
3. CEED problems are investigated and validated on the three test systems IEEE 30-bus, 6-generator system at a demand load of 283.4 MW, Ten generating unit test system with a demand of 2000MW and 40 unit actual generating Tai power station with demand of 10500 MW having quadratic cost and emission function with valve point loading effects, RRL, POZ and transmission losses. The results are compared with various meta-heuristic algorithms, PCPSO avoids to struck in the premature convergence in local minima which results in better economic and emission impact,

computational effectiveness, and its convergence feature. As a result, PCPSO [401] optimization is a viable method for addressing challenging issues in power systems.

4. Here CEED optimization problem is simulated on perfectly convergent Particle swarm optimization (PCPSO)[401] for solving combined economic and multiple emissions dispatch problems while taking into account the impacts of SO2,NOx,CO2 pollutants with seven price penalty factors using cubic cost and emission functions. Two types of objective are considered first is bi-objective and other is four objectives with all constraints. Cubic cost functions are more accurate and show the actual response of all thermal units. This algorithm has better search capabilities with strong convergence characteristics that minimizes the cubic cost and cubic multiple emissions functions at various load demands with minimum transmission losses for an IEEE 30 bus, 6 generators test system. The simulation test results were compared with Lagrange method, simulated annealing, PSO, Bio-geography based method, grasshopper optimization algorithm, Artificial ecosystem based optimization, Multi objective 4th chaotic function Artificial ecosystem based optimization, Quantum Particle swarm optimization and Sine-cosine algorithm. This algorithm is fast, robust, accurate and takes less computational time with better results for solving such non-convex problems.

5. Energy transfer between locations and fossil fuel emissions from generating units are key concerns in multi-area CEED problems. Developed the PCPSO [401] algorithm for single area and multi-area(two area and four area) for 40 generating test unit system with tie line limits, generator constraints, POZ, RRL having non smooth cost and emission function with max-max price penalty factor having a total load of 10500 MW. Second case is with 12 generating unit test system with all the above said constraints as in 40 generating unit case, with a total load of 2090 MW. Here single

area, two area, three areas and four areas are considered with min-max, max-max, max-min, min-min price penalty factors are used to calculate CEED and in both cases it is compared with meta heuristic algorithms published in latest papers. Results show PCPSO approach increases computational efficiency when compared to previous PSO variants or other heuristic techniques and achieves a speedy true global minimum, as per the results. By automatically generating a new particle mechanism, this programme assists and gives a chance to all particles who are experiencing premature convergence and stagnation. if the best particle changes its position each time for ρ(k) to readjust and stabilize. By assisting particles in premature convergence stages and allowing all particles to reach real global minima, PCPSO was able to provide balanced exploration and exploitation in the search space.

8.3 Future scope of work

The aim of the research work is obtained successfully and can be easily applied in the following areas

1) Testing of the developed algorithm can be applied to bigger IEEE power system models.
2) Integration of solar hydro thermal systems into the power grid can be applied by using the developed algorithm.
3) Integration of the developed algorithm can be made with wind thermal power plants.
4) More meta-heuristic algorithms like Bat algorithm, differential evolution, harmony search, etc. can be incorporated with the developed algorithm to form hybrid algorithms.
5) Developed algorithm can be used in parallel computing for faster results in real world problems.

Appendix

Six unit generator characteristics

Unit	$P_{i\,min}$	$P_{i\,max}$	a_i	b_i	c_i	d_i	e_i	f_i
	MW	**MW**	**$/h**	**$/MWh**	$(\$/MW^2)h$	**Ib/h**	**Ib/MWh**	$(Ib/MW^2)h$
1	50	200	0.0038	2	0	22.9830	-1.1000	0.0126
2	20	80	0.0175	1.750	0	22.3130	-0.1000	0.0200
3	15	50	0.0625	1.0	0	25.5050	-0.1000	0.0270
4	10	35	0.0083	3.250	0	24.9000	-0.0050	0.0291
5	10	30	0.0250	3.0	0	24.7000	-0.0400	0.0290
6	12	40	0.0250	3.0	0	25.3000	-0.0055	0.0271

Transmission loss formula Co-efficient of six unit system

B00=1.4000

B01=[-3.000 2.1000 -5.6000 3.4000 1.5000 7.8000]

$$B=\begin{bmatrix} 2.1800 & 1.0300 & 9.0000 & -1.0000 & 2.0000 & 2.70000 \\ 1.0300 & 1.8100 & 4.0000 & -1.5000 & 2.0000 & 3.0000 \\ 9.0000 & 4.0000 & 4.1700 & -1.3100 & -1.5300 & -1.0700 \\ 1.0000 & -1.5000 & -1.3100 & 2.2100 & 9.4000 & 5.0000 \\ 2.0000 & 2.0000 & -1.5300 & 9.4000 & 2.4300 & 0.0000 \\ 2.7000 & 3.0000 & -1.0700 & 5.0000 & 0.0000 & 3.5800 \end{bmatrix}$$

Six unit generator cubic characteristics

Bus number		**1**	**2**	**5**	**8**	**11**	**13**
Gen	$P_{i\,min}$ MW	50	20	15	10	10	12
Limits	$P_{i\,max}$ MW	200	80	50	50	50	40
Fuel cost	$a_i(\$/MW^3)h$	0.0010	0.0004	0.0006	0.0002	0.0013	0.0004
coefficients	$b_i(\$/MW^2)h$	0.092	0.025	0.075	0.10	0.12	0.084
	$c_i(\$/\text{MWh})$	14.5	22	23	13.5	11.5	12.5
	$d_{i(}\$/\text{h})$	-136	-3.5	-81	-14.5	-9.75	75.6
SO2	$a_{so2}(\$/t^3)h$	0.0005	0.0014	0.0010	0.0020	0.0013	0.0021
Emission	$so2(\$/t^2)h$	0.150	0.055	0.035	0.070	0.120	0.080
coefficients	$c_{so2}(\$/\text{t h})$	17.0	12.0	10.0	23.5	21.5	22.5
	$d_{so2(}\$/\text{h})$	-90.0	-30.5	-80.0	-34.5	-19.75	25.6
NOx	$a_{so2}(\$/t^3)h$	0.0012	0.0004	0.0016	0.0012	0.0003	0.0014
Emission	$so2(\$/t^2)h$	0.052	0.045	0.050	0.070	0.040	0.024
coefficients	$c_{so2}(\$/\text{t h})$	18.5	12.0	13.0	17.5	8.5	15.5
	$d_{so2(}\$/\text{h})$	-26.0	-35.0	-15.0	-74.0	-89.0	-75.0
CO2	$a_{so2}(\$/t^3)h$	0.0015	0.0014	0.0016	0.0012	0.0023	0.0014
Emission	$so2(\$/t^2)h$	0.092	0.025	0.055	0.010	0.040	0.080
coefficients	$c_{so2}(\$/\text{t h})$	14.0	12.5	13.5	13.5	21.0	22.0
	$d_{so2(}\$/\text{h})$	-16.0	-93.5	-85.0	-24.5	-59.0	-70.0

Ten unit generator characteristics

Unit	$P_{i\,min}$	$P_{i\,max}$	a_i	b_i	c_i	d_i	e_i	$\propto_i$	β_i	γ_i	η_i	δ_i
	MW	**MW**	$\$/h$	$\$/MWh$	$(\$/MW^2)h$	$\$/h$	**rad/MW**	**Ib/h**	**Ib/MWh**	$(Ib/MW^2)h$	**Ib/h**	**1/MW**
1	10	55	1000.403	40.5407	0.12951	35	0.0174	360.0012	-3.9864	0.04702	0.25475	0.01234
2	20	80	950.606	39.5804	0.10908	25	0.0178	350.0056	-3.9524	0.04652	0.25475	0.01234
3	47	120	900.705	36.5104	0.12511	32	0.0162	330.0056	-3.9023	0.04652	0.25163	0.01215
4	20	130	800.705	39.5104	0.12111	30	0.0168	330.0056	-3.9023	0.04652	0.25163	0.01215
5	50	160	756.799	38.5390	0.15247	30	0.0148	13.8593	0.3277	0.00420	0.24970	0.01200
6	70	240	451.325	46.1592	0.10587	20	0.0163	13.8593	0.3277	0.00420	0.24970	0.01200
7	60	300	1243.531	38.3055	0.03546	20	0.0152	40.2669	-0.5455	0.00680	0.24800	0.01290
8	70	340	1049.998	40.3965	0.02803	30	0.0128	40.2669	-0.5455	0.00680	0.24990	0.01203
9	135	470	1658.569	36.3278	0.02111	60	0.0136	42.8955	-0.5112	0.00460	0.25470	0.01234
10	150	470	1356.659	38.2704	0.01799	40	0.0141	42.8955	-0.5112	0.00460	0.25470	0.01234

Transmission loss formula Co-efficents of ten unit system

$$\begin{bmatrix}
0.000049 & 0.000014 & 0.000015 & 0.000015 & 0.000016 & 0.000017 & 0.000017 & 0.000018 & 0.000019 & 0.000020 \\
0.000014 & 0.000045 & 0.000016 & 0.000016 & 0.000017 & 0.000015 & 0.000015 & 0.000016 & 0.0000\ 18 & 0.000018 \\
0.000015 & 0.000016 & 0.000039 & 0.000010 & 0.000012 & 0.000012 & 0.000014 & 0.000014 & 0.000016 & 0.000016 \\
0.000015 & 0.000016 & 0.000010 & 0.0000040 & 0.000014 & 0.000010 & 0.000011 & 0.000012 & 0.000014 & 0.000015 \\
0.000016 & 0.000017 & 0.000012 & 0.000014 & 0.000035 & 00000.11 & 0.000013 & 0.000013 & 0.000015 & 0.000016 \\
0.000017 & 0.000015 & 0.000012 & 0.000010 & 0.000011 & 0.000036 & 0.000012 & 0.000012 & 0.000014 & 0.000015 \\
0.000017 & 0.000015 & 0.000014 & 0.000011 & 0.000013 & 0.000012 & 0.000038 & 0.000016 & 0.000016 & 0.000018 \\
0.000018 & 0.000016 & 0.0000140 & 0.000012 & 0.000013 & 0.000012 & 0.000016 & 0.000040 & 0.000015 & 0.000016 \\
0.000019 & 0.000018 & 0.000016 & 0.000014 & 0.000015 & 0.000014 & 0.000016 & 0.000015 & 0.000042 & 0.000019 \\
0.000020 & 0.000018 & 0.000016 & 0.000015 & 0.000016 & 0.000015 & 0.0000188 & 0.000016 & 0.000019 & 0.000044
\end{bmatrix}$$

Twelve unit generator characteristics

Unit	$P_{i\,min}$	$P_{i\,max}$	a_i	b_i	c_i	$\propto_i$	β_i	γ_i
	MW	**MW**	**$/h**	**$/MWh**	$(\$/MW^2)h$	**Ib/h**	**Ib/MWh**	$(Ib/MW^2)h$
1	35	210	1243.5311	38.30553	0.03546	40.26690	-0.54551	0.00683
2	130	325	1658.5696	36.32782	0.02111	42.89553	-0.51160	0.00461
3	125	315	1356.6592	38.27041	0.01799	42.89553	-0.51160	0.00461
4	10	150	756.7989	38.53973	0.15247	33.85932	-0.32767	0.00484
5	35	110	449.9977	40.39655	0.02803	50.639310	-0.54551	0.00754
6	125	215	558.5696	38.34001	0.14834	45.83267	-.0.63262	0.00661
7	15	175	451.3251	46.15916	0.10587	48.21560	-0.43211	0.00914
8	30	215	673.0267	43.83562	0.07505	52.45210	-0.61173	0.00533
9	50	335	530.7199	50.63211	0.11934	41.10420	-0.49731	0.00674
10	15	175	851.3251	46.15916	0.10587	30.36320	-0.6821	0.00728
11	30	215	1038.533	41.03782	0.13552	26.17650	-0.50660	0.00479
12	50	335	1285.907	33.56211	0.08963	27.75490	-0.49340	0.00387

Tie line min limit one side TLmin_jm	**Tie line min limit other side TLmin_mj**	**Tie line max limit one side TLmax_jm**	**Tie line max limit other side TLmax_mj**	**Tie line fractional loss rate values TLflr_jm**
5	5	60	50	0.11
5	5	50	60	0.21
5	5	60	60	0.14
5	5	60	60	0.16
5	5	60	50	0.22
5	5	50	60	0.11

Transmission loss formula Co-efficient of twelve unit system

$$\begin{bmatrix}
0.000071 & 0.00003 & 0.000025 \\
0.00003 & 0.000069 & 0.000032 \\
0.000025 & 0.000032 & 0.00008 \\
0.000056 & 0.000045 & 0.000015 \\
0.000023 & 0.000042 & 0.000047 \\
0.000032 & 0.000023 & 0.000027 \\
0.00002 & 0.000028 & 0.000053 \\
0.000086 & 0.000034 & 0.000016 \\
0.000053 & 0.000016 & 0.000028 \\
0.000074 & 0.00003 & 0.000025 \\
0.000049 & 0.000069 & 0.000037 \\
0.000022 & 0.000032 & 0.000083
\end{bmatrix}$$

Forty unit generator characteristics

Unit	$P_{i\,min}$	$P_{i\,max}$	a_i	b_i	c_i	d_i	e_i	$\propto_i$	β_i	γ_i	η_i	δ_i
	MW	MW	$/h	$/MWh	$(\$/MW^2)h$	$/h	rad/MW	ton/h	ton/MWh	$(ton/MW$	ton/h	1/MW
1	36	114	94.705	6.73	0.0069	100	0.084	60	−2.22	0.048	1.31	0.0569
2	36	114	94.705	6.73	0.0069	100	0.084	60	−2.22	0.048	1.31	0.0569
3	60	120	309.54	7.07	0.02028	100	0.084	100	-2.36	0.0762	1.31	0.0569
4	80	190	369.03	8.18	0.000942	150	0.063	120	-3.14	0.054	0.9142	0.0454
5	47	97	148.89	5.35	0.0114	120	0.077	50	-1.89	0.085	0.9936	0.0406
6	68	140	222.33	8.05	0.01142	100	0.084	80	-3.08	0.0854	1.31	0.0569
7	110	300	278.71	8.03	0.00357	200	0.042	100	-3.06	0.0242	0.655	0.02846
8	135	300	391.98	6.99	0.00492	200	0.042	130	-2.32	0.031	0.655	0.02846
9	135	300	455.76	6.6	0.00573	200	0.042	150	-2.11	0.0335	0.655	0.02846
10	130	300	722.82	12.9	0.00605	200	0.042	280	-4.34	0.425	0.655	0.02846
11	94	375	635.2	12.9	0.00515	200	0.042	220	-4.34	0.0322	0.655	0.02846
12	94	375	654.69	12.8	0.00569	200	0.042	225	-4.28	0.0338	0.655	0.02846
13	125	500	913.4	12.5	0.00421	300	0.035	300	-4.18	0.0296	0.5035	0.02075
14	125	500	1760.4	8.84	0.00752	300	0.035	520	-3.34	0.0512	0.5035	0.02075
15	125	500	1728.3	9.15	0.00708	300	0.035	510	-3.55	0.0496	0.5035	0.02075
16	125	500	1728.3	9.15	0.00708	300	0.035	510	-3.55	0.0496	0.5035	0.02075
17	220	500	647.85	7.97	0.00313	300	0.035	220	-2.68	0.0151	0.5035	0.02075
18	220	500	649.69	7.95	0.00313	300	0.035	222	-2.66	0.0151	0.5035	0.02075
19	242	550	647.83	7.97	0.00313	300	0.035	220	-2.68	0.0151	0.5035	0.02075
20	242	550	647.81	7.97	0.00313	300	0.035	220	-2.68	0.0151	0.5035	0.02075
21	254	550	785.96	6.63	0.00298	300	0.035	285	-2.22	0.0145	0.5035	0.02075
22	254	550	785.96	6.63	0.00298	300	0.035	285	-2.22	0.0145	0.5035	0.02075
23	254	550	794.53	6.66	0.00284	300	0.035	295	-2.26	0.0138	0.5035	0.02075
24	254	550	794.53	6.66	0.00284	300	0.035	295	-2.26	0.0138	0.5035	0.02075
25	254	550	801.32	7.1	0.00277	300	0.035	310	-2.42	0.0132	0.5035	0.02075
26	254	550	801.32	7.1	0.00277	300	0.035	310	-2.42	0.0132	0.5035	0.02075
27	101	50	1055.1	3.33	0.52124	120	0.077	360	-1.11	1.842	0.9936	0.0406
28	101	50	1055.1	3.33	0.52124	120	0.077	360	-1.11	1.842	0.9936	0.0406
29	101	50	1055.1	3.33	0.52124	120	0.077	360	-1.11	1.842	0.9936	0.0406
30	47	97	148.89	5.35	0.0114	120	0.077	50	-1.89	0.085	0.9936	0.0406
31	60	190	222.92	6.43	0.0016	150	0.063	80	-2.08	0.0121	0.9142	0.0454
32	60	190	222.92	6.43	0.0016	150	0.063	80	-2.08	0.0121	0.9142	0.0454
33	60	190	222.92	6.43	0.0016	150	0.063	80	-2.08	0.0121	0.9142	0.0454
34	90	200	10787	8.95	0.0001	200	0.042	65	-3.48	0.0012	0.655	0.02846
35	90	200	116.58	8.62	0.0001	200	0.042	270	-3.24	0.0012	0.655	0.02846
36	90	200	116.58	8.62	0.0001	200	0.042	270	-3.24	0.0012	0.655	0.02846
37	25	110	307.45	5.88	0.0161	80	0.098	100	-1.98	0.095	1.42	0.0677
38	25	110	307.45	5.88	0.0161	80	0.098	100	-1.98	0.095	1.42	0.0677
39	25	110	307.45	5.88	0.0161	80	0.098	100	-1.98	0.095	1.42	0.0677
40	242	550	647.83	7.97	0.00313	300	0.035	220	-2.68	0.0151	0.5035	0.02075

List of Publications

[1] Devinder kumar, N.k.Jain, Uma Nangia, ”Perfectly convergent Particle swarm optimization in multi-dimensional space”, International journal of Bio-inspired computation, Inderscience Publications, vol 18,no 4,pp.221-228,2021

[2] N.K. Jain, Uma Nangia , Devinder Kumar, “Machine learning through back propagation networks using OPSO with Cauchy mutation and GCPSO in higher dimensions“ in proceedings of the IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES) Electrical Engineering Department, Delhi Technological Univeristy,Delhi,pp.1-6,2018

[3] Devinder kumar, N.k.Jain, Uma Nangia, Combined Economic Emission Dispatch using Perfectly convergent Particle swarm optimization”, in proceedings of the IEEE International Conference on on Electrical, Electronics and Computer Engineering, Delcon, February 11-13, 2022.

[4] Devinder kumar, N.k.Jain, Uma Nangia, Multi area Economic Emission load dispatch using perfectly convergent Particle swarm optimization”, , in proceedings of the IEEE International Conference on on Electrical, Electronics and Computer Engineering, Delcon, February 11-13, 2022.

REFERENCES:

[1] Reeves WT "Particle systems-a technique for modelling a class of fuzzy objects", ACM Trancs Graphics 2(2):91-108, 1983.

[2]Reyonlds CW "Flocks, heards and schools" A distributed behavioural model, Computer Graphics, volume 2, number 4,25-34(Proc of SINGRAPH" 1987).

[3]Reyonlds CW: Home page on Boids, http://www.red3d.com/cwr/boids/

[4] Heppner F,Grenander U" A Stochastic nonlinear model for coordinated bird flocks", In: Krasner's (Eds) The ubiquity of chaos, AAAS Publications, Washington DC, 1990.

[5]Hoffmeyer J "The Swarming body", in Semiotics around the world, proceeding of fifth congress on International associationfor Semiotics studies, Berkley, 1994.

[6]J. Kennedy, and R. Eberhart. "Particle swarm optimization", Proceedings of the IEEE International Conference on Neural Networks .IEEE Service Centre, Piscataway, NJ, IV, pp.1941-1948, 1995.

[7] Eberhart, R. C., and Kennedy, J. A new optimizer using particle swarm theory. Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Centre, Piscataway, NJ, 39-43, 1995.

[8] Eberhart RC, Simpson P, Dobbins R, Computational intelligence PC tools, AP Professional, San Diego, CA, chapter 6, pp. 212-226, 1996.

[9]Kennedy, J, "The Particle Swarm: Social Adaptation of Knowledge", Proceedings of the 1997 International Conference on Evolutionary Computation, pp. 303-308, IEEE Press, 1997.

[10]Kennedy, J and Eberhart, R.C, "A discrete binary version of the particle swarm algorithm", Int. IEEE Conf. on Systems, Man, and Cyber, Vol.5, pp.4104 – 4108, 1997.

[11]Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in Proc. IEEE Int. Conf. Evolutionary Computation, pp. 69-73, '1998.

[12] Ozcan E, Mohan CK Analysis of a simple particle swarm optimization system. Intelligent Engineering, Systems through Artificial Neural Networks 253–258,1998.

[13] Ozcan E, Mohan CK Particle swarm optimization: surfing the waves. Proceedings of IEEE congress On evolutionary computation, Washington, DC, 1999.

[14] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in Proc. IEEE Int. Conf. Evolutionary Computation, Washington, DC, pp. 1945-50, '1999.

[15] Clerc, M. the swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings, ICEC, Washington, DC, pp. 1951-1957, 1999.

[16] Kennedy, J Small worlds and Mega-Minds: Effects of neighbourhood topology on Particle swarm Performance, congress on Evolutionary Computation, Vol-Ill, pp. 1931-38, 1999.

[17] Suganthan, P.N., “Particle Swarm Optimiser with Neighbourhood Operator”, Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, pp. 1958-1962, 1999.

[18] Eberhart, R. C., and Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. Proc. Congress on Evolutionary Computation 2000, San Di,ego, CA, pp. 84-88, 2000.

[19] Carlisle A, Dozier G , Adapting particle swarm optimization to dynamic environments. Proceedings of International conference on artificial intelligence, vol 1, pp 429–434, Las Vegas, NV, 2000.

[20] Kennedy J, Stereotyping: improving particle swarm performance with cluster analysis. Proceedings of IEEE congress on evolutionary computation, pp. 1507–1512, San Diego, CA,2000.

[21] Van den Bergh F and Engelbrecht A. P, Effect of swarm size on cooperative particle swarm optimizers”, Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 892–899, 2001.

[22] Carlisle A, Dozier G Tracking changing extrema with particle swarm optimizer. Auburn University Technical Report CSSEOl-08, 2001.

[23] Carlisle A, Dozier G,”An off-The-Shelf PSO”, Proceedings of workshop on particle swarm optimization. Indianapolis, IN, 2001.

[24] Carlisle A, Dozier G,Tracking changing extrema with adaptive particle swarm optimizer”, Proceeding Of WAC 2002, Orlando, Florida,2002.

[25] Clerc M, Kennedy .J, The particle swarm-explosion stability, and convergence in a multidimensional complex space”, IEEE Transactions on Evolutionary Computation, 2002, 6(1):58-73,2002.

[26] Ratnaweera, A, Halgamuge, S., Watson, H., “Particle Swarm Optimization with Time Varying Acceleration Coefficients”, Proceedings of the International Conference on Soft Computing and Intelligent Systems, pp. 240-255, 2002.

[27] Kennedy, J., Mendes, R., “Population Structure and Particle Swarm”, Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1671-1676, IEEE, 2002.

[28] van den Bergh F, Engelbrecht AP , A new locally convergent particle swarm optimizer. Proceedings of IEEE conference on systems, man and cybernetics, Hammamet, Tunisia,2002.

[29] Fan HY,A modification to particle swarm optimization algorithm? Eng Comput 19(7–8):970–989, 2002.

[30] Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," Information Processing Letters, vol. 85, no. 6, pp. 317-325, 2003.

[31] Zheng, Y.-L., Ma, L.-H., Zhang, L.-Y., & Qian, J.-X On the convergence analysis and parameter selection in particle swarm optimization In Proceedings of the IEEE international conference on machine learning and cybernetics (pp. 1802–1807). Piscataway: 2003.

[32] Ratnaweera A, Halgamuge SK, Watson HC , Self-organising hierarchical particle swarm optimizer with time-varying acceleration coefficients”, IEEE Trans Evol Comput 8(3):240–255,2004.

[33] Chatterjee, A., and Siarry, P, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization”, *Computers and Operations Research*, 33(3):859–871, 2006.

[34] Bratton, D., and Kennedy, J, Defining a standard for particle swarm optimization. In IEEE Swarm Intelligence Symposium, pp. 120–127, 2007.

[35] Arumugam, M. S., Rao, M., and Chandramohan, A, A new and improved version of particle swarm optimization algorithm with global–local best parameters. Knowledge and Information Systems, 16(3):331–357, 2008.

[36] Chen, D. B., and Zhao, C. X, Particle swarm optimization with adaptive population size and its application, Applied Soft Computing, 9(1):39–48, 2009.

[37] Nakagawa, N, Ishigame, A,Yasuda, K. Particle swarm optimization withapproximate gradient. IEEJ Transactions on Electrical and Electronic Engineering, Wiley Subscription Services, Inc., A Wiley Company, v. 3, n. 5, p. 590–592, 2009. ISSN1931-4981.

[38] Van den Bergh, F,and Engelbrecht, A, A convergence proof for the particle swarm optimiser. FundamentaInformaticae, 105(4):341–374, 2010.

[39] Nickabadi, A., Ebadzadeh, M. M., and Safabakhsh, R, A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing, 11(4):3658–3670, 2011.

[40] Bansal, J.C, SinghP.K, Saraswat Mukesh, VermaAbhishek, Jadon Shimpi Singh, Abraham Ajith, Inertia weight strategies in particle swarm optimization ,Third World Congress on Nature and Biologically Inspired Computing, p.p633–640.2011.

[41] Engelbrecht, A, Particle swarm optimization: Velocity initialization, in Proceedings of the IEEE Congress on Evolutionary Computation (pp. 1–8). doi:10.1109/CEC.2012.6256112, 2012.

[42] Schmitt M, and Wanka, R, Particle swarm optimization almost surely finds local optima. In Genetic and Evolutionary Computation Conference, pp. 1629–1636, 2013.

[43] Cleghorn, C. W., and Engelbrecht, A. P, A generalized theoretical deterministic particleswarm model. Swarm intelligence, 8(1):35–59, 2014.

[44] van Zyl, E., & Engelbrecht, A, A subspace-based method for pso initialization.", In Proceedings of the IEEE Symposium Series on Computational Intelligence pp. 226–233,doi:10.1109/SSCI.2015.42, 2015.

[45] Bonyadi, M., and Michalewicz, Z, Analysis of stability, local convergence, and transformation sensitivity of a variant of particle swarm optimization algorithm. IEEE Transactions on Evolutionary Computation, 20(3):370–385, 2016.

[46] Liu, Q., Wei, W., Yuan, H., Zhan, Z.H., Li, Y., "Topology selection for particle swarm optimization", Inf. Sci. 363 , 154–173, 2016.

[47] M. R. Bonyadi and Z. Michalewicz. Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. IEEE Transactions on Evolutionary Computation, 21(3):378–390, June 2017.

[48] Oldewage, E. T, The Perils of Particle Swarm Optimisation in High Dimensional Problem Spaces. Master's thesis University of Pretoria,Pretoria, South Africa, 2018.

[49] HaiyanShi, ShilongLiu, HongkunWu. RuoweiLi, Sanchi Liu, Ngaiming Kwok, Oscillatory Particle swarm optimizer,Applied soft computing journal,73,316-327, 2018.

[50] Oldewage, E.T. Engelbrecht A.P,Cleghorn C.W, Movements patterns of a Particle Swarm in High dimensions spaces, Journal of Information Science, 2019.

[51] Lijun Sun, Xiaodong, Tianfei Chen, An improved convergence Particle swarm optimization algorithm with Random sampling of control parameters, Journal of Control science and engineering,Article ID 7478498,vol 2019,Hindawi.

[52]Kennedy J, Eberhart RC, A discrete binary version of the particle swarm algorithm. Proceedings of the conference on systems, man and cybernetics, Piscataway, New Jersey, pp 4104-4109, 1997.

[53]P. J. Angeline, "Using selection to improve particle swarm optimization," in Proc. IEEE Congr. Evol.Comput., pp. 84-89,1998.

[54]Shi, Y. & Eberhart, R. C, Fuzzy adaptive particle swarm optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, South Korea, (101-106), 2001.

[55]M. Lovberg et al., "Hybrid Particle Swarm Optimiser with Breeding and Sub-populations", Proceedings of GECC02001- Genetic and Evolutionary Computation Conference 2001, pp. 409, July 7-11, 2001

(56] Xie, X.-F., Zhang, W.-J., and Yang, Z.-L, "A Dissipative Particle Swarm Optimization", Cong. on Evolutionary Comput. (CEC), pp.1456-1461, 2002

[57] Al-kazemi B, Mohan CK (2002) Multi-phase Discrete Particle Swarm Optimization. Proceedings of fourth international workshop on frontiers in evolutionary algorithms (FEA 2002)

[58] V. Miranda, N. Fonseca, "New Evolutionary Particle Swarm Algorithm (EPSO) Applied to Voltage/Var Control", Proceedings of PSCC'02 -Power System Computation Conference, June 24-28, 2002

[59]Riget, J. and Vesterstroem, J. S,"A diversity-guided particle swarms optimizer –the ARPSO" Technical Report No. 2002-02. Dept. of Computer Science, University of Aarhus, EVALife, 2002.

[60] Brits, R., Engelbrecht, A.P., and van den Bergh, F., "Niche Particle Swarm Optimization". Technical report, Department of Computer Science, University of Pretoria, 2002.

[61] Krink T, Vesterstrom JS, Riget J Particle swarm optimization with spatial particle extension. Proc Cong EvolComput (CEC'02) 2: 1474-1479, 2002

[62] K. Parsopoulos and M. Vrahatis, “Recent approaches to global optimization problems through particle swarm optimization,” Natural Computing, vol. 1, pp. 235–306, May 2002.

[63]Brits, R, Engelbrecht A.P, van den Bergh, F. Solving systems of unconstrained equations using particle Swarm optimization. In Proceedings of the IEEE 2002 Conference on Systems, Man, and Cybernetics, Yasmine Hammamet, Tunisia, 6–9 October 2002.

[64] J. Kennedy, “Bare bones particle swarms,” in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS ’03)*, pp. 80–87, Indianapolis, Ind, USA, 2003.

[65] Secrest, B.R. and Lamont, G.B, "Visualizing particle swarm optimization -Gaussian particle swarm Optimization", Proc. Of Swarm lntell. Symp. (IEEE/SIS), pp. 198-204, 2003

[66] Peram, T., Veeramachaneni, K., and Mohan, C. K, "Fitness-distance-ratio based particle swarm optimization". Proc. of the IEEE/SIS, pp. 174-181, 2003.

[67] Cui, Zh. Zeng, J. And Cai, X., "A guaranteed convergence dynamic double particle swarm optimizer", Fifth World Cong. on Intel!., Control and Automation, vol. 3, pp. 2184 - 2188,2004.

[68] Mendes, R., Kennedy, J., and Neves, J, "The fully informed particle swarm: simpler, maybe better," IEEE Trans. on Evolutionary Comput., Vol. 8 (3), pp. 204—210, 2004.

[69] Noel, M. M., & Jannett, T. C. Simulation of a new hybrid particle swarm optimization algorithm. Proc. Of the IEEE Symposium on System Theory (pp. 150 -153).2004

[70] Sun, J., Feng, B. and Xu,W, "Particle swarm optimization with particles having quantum behaviour", Proc. IEEE/CEC, Vol. 1, pp. 325 -331,2004.

[71] K. E. Parsopoulos and M. N. Vrahatis, "UPSO-a unified particle swarm optimization scheme," in Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE '04), vol. 1 of Lecture Series on Computational Sciences, pp. 868-873, Zeist, The Netherlands, 2004.

[72] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation. vol.8, no.3, pp.225-239, 2004.

[73] X. Li, “Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization,” in Proc Genetic Evol. Computing. Conf., Jun. 2004, pp. 105–116,2004.

[74]Monson, C. K., &Sepp, I. K. D, The Kalman Swarm: A New Approach to Particle Motion in Swarm Optimization”Proceedings of the Genetic and Evolutionary Computation Conference. Springer, 140-150, 2004.

[75] Chang, J.-F., Chu, S.-C., Roddick, J.F and Pan, J.-S,"A parallel particle swarm optimization algorithm with communication strategies," J. of Information Science and Eng., Vol. 21, No. 4, pp. 809-818, 2005.

[76] Pampara, G., Franken, N. and Engelbrecht, A, "Combining particle swarm optimization with angle modulation to solve binary problems," The IEEE Cong. on Evolutionary Computing., Vol. 1, pp. 89-96, 2005.

[77] Poli R, Langdon WB, Holland O (2005) Extending particle swarm optimization via genetic programming. In: Keijzer M, Tettamanzi A, Collet P, van Hemert J, Tomassini M (eds) Proceedings of eighth European conference, EuroGP 2005. Lausanne, Switzerland, March 30–April 1, 2005

[78] Stefan Janson and Martin Mid□endorf, "A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant", IEEE Transactions on Systems, Man, and Cybernetics-PART B: CYBERNETICS, VOL. 35, NO. 6, December 2005

[79] Yang, C. and Simon, D, "A New Particle Swarm Optimization Technique", Proc. Of the Int. Conf. on systems Eng., (IEEE/ISEng'05), 2005

[80] Madar, J., Abonyi, J. and Szeifert, F, "Interactive particle swarm optimization", Proc. Int. IEEE conf. On Intelligent Systems Design and Applications (IEEE/ ISDA), Vol.8, No.10, pp.314 -- 319., 2005.

[81] Dou Q, Zhou, C, Pan, G, Luo, H. and Liu, Q," Neural particle swarm optimization for casing damage prediction", Int. Sympos. On neural networks, vol. 3498, 2005

[82] Yuan, Z., Jin, R., Geng, J., Fan, Y., Lao, J., Li, J., Rui, X., Fang, Z. and Sun, J, "A perturbation particle swarm optimization for the synthesis of the radiation pattern of antenna array", Proc. IEEE Conf. on Asia- Pacific, Vol.3, No, 4-7, 2005.

[83] Voss, M.S, "Principal component particle swarm optimization (PCPSO)", Proc., Of the IEEE Symp. On swarm Intel!., pp. 401 – 404,2005.

[84] H. R. Tizhoosh, Opposition-based learning,: A new scheme for machine intelligence, International Conference on Computational Intelligence for Modeling Control and Automation - CIMCA 2005, Vienna, Austria, val. I, 695-701, 2005

[85] Hongbo, L., Abraham, A., & Zhang, W, Fuzzy adaptive turbulent particle swarm optimization. Proceedings of The Fifth international Conference on Hybrid Intelligent System, pp.6, 2005.

[86] Zeng, J. and Hu, J. and Jie, J, "Adaptive Particle Swarm Optimization Guided by Acceleration Information", Proc. IEEE/ ICCIAS, Vol.1, pp.351-355, 2006.

[87] Liang, J., J., Qin, A., K., and Baskar, S,"Comprehensive Learning Particle Swarm Optimizer for Global Optimization of multimodal Functions" IEEE Trans. Evolutionary Computation, Vol. 10, No. 3, 2006.

[88] Wang, X.,Wang, Y., Zeng, H., and Zhou, H," Particle Swarm Optimization with Escape Velocity" Int. Conf. on Computational lntell.and Security, Vol. 1, pp. 457-460, 2006.

[89] Yin, P. Y, "Genetic particle swarm optimization for polygonal.approximation of digital curves" J. of Pattern Recognition and lmage Analysis., Vol. 16, No. 2, pp. 223-233, 2006.

[90] Sadri, J. and Suen, C.Y,"A Genetic Binary Particle Swarm Optimization Model", Proc. IEEE/CEC, pp.656 – 663, 2006.

[91] Pasupuleti, S. and Battiti, R, "The Gregarious Particle Swarm Optimizer (GPSO)", GECC0'06, 2006.

[92] Chandrasekaran, 5.,Ponnambalam, S.G., Suresh, R.K. and Vijayakumar, N, "A Hybrid Discrete Particle Swarm Optimization Algorithm to solve Flow Shop Scheduling Problems", Proc. IEEE/ ICCIS, pp.1- 6, 2006.

[93] Roy, R. and Ghoshal, S.P, "Evolutionary computation based optimization in fuzzy automatic generation control", IEEE/ POWER, pp.7, 2006.

[94] Zhao B," An Improved Particle Swarm Optimization Algorithm for Global Numerical Optimization", Int. Conf. on Comput. Science N6, Reading, (Royaume-uni), Vol. 3994, pp. 657-664, 2006.

[95] Sedlaczek, K., Eberhard, P, "Using Augmented Lagrangian Particle Swarm Optimization for Constrained Problems in Engineering", J. of Structural and Multidisciplinary Optimization, Vol. 32, No. 4, pp. 277-286, 2006.

[96] Meissner, M., Schmuker, M. and Schneider, G, "Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training", BMC Bioinformatics, 7:125, 2006.

[97] Koh, B-11, Fregly, B.-J., George, A.D and Haftka, R.-T," Parallel asynchronous particles swarm for global biomechanical", Int J Number Methods Eng., 67(4): 578-595, 2005.
[98] Ozcan, E. and Yilmaz, M, "Particle Swarms for Multimodal Optimization" Lecture Notes In Computer Science; Vol. 4431, Proc. Of the 8th int. cont. on Adaptive and Natural Computing Algorithms, Part I. pp. 366 – 375, 2006.
[99] Lu, H., and Chen, W (2006)," Dynamic-objective particle swarm optimization for constrained optimization problems", J. of Combinatorial Optimization, Vol. 12, No., 4, PP. 409-419.
[100] Jie, J., Zeng, j. and Han, C., (2006), "Self-Organization Particle Swarm Optimization Based on Information Feedback", Advances in natural comput. (Part 1-11: Second Int. conf., ICNC 2006, Xi'an, China.
[101] Li, T., Lai, X. and Wu, M., (2006), "An Improved Two-Swarm Based Particle Swarm Optimization Algorithm", Proc. IEEE/ WCICA, Vol. 1, pp.3129-3133.
[102] Moore, P.W. Venayagamoorthy, G.K., (2006), 'Empirical Study of an Unconstrained Modified Particle Swarm Optimization', IEEE/CEC, pl). 1477:1482.
[103] Xu, F., and Chen, W., {2006), "Stochastic Portfolio Selection Based on Velocity Limited Particle Swarm Optimization", Proc. IEEE/ WCICA, Vol. 1, pp.3599-3603.
[104] Shen, X., Wei, K., Wu, D., Tong, Y. and Li, Y., (2007), "A Dynamic Adaptive Dissipative Particle Swarm Optimization with Mutation Operation", Proc. IEEE/ ICCA, pp. 586-589.
[105] Atyabi, A. and Phon-Amnuaisuk, S. (2007), "Particle swarm optimization with area extension (AEPSO)", IEEE/CEC, pp. 1970-1976
[106] Hui,W. and Feng, Q., (2008), "Improved particle swarm optimizer with behaviour of distance models and its applications in soft-sensor", J. of Computer Eng. and Applications, 43 (30): 30-32.
[107] Alviar, J.-B., Pena, J., and Hincapie, R., (2007), "Subpopulation best rotation: a modification on PSO", RevistaFacultad de lngenieria. No. 40, pp.118-122.
[108] Wilke, D., Kok, S., and Groenwold, A. {2007). Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance. International Journal for Numerical Methods in Engineering, 70(8):985-1008.
[109] Jarbouia, B., Cheikha, M., Siarryb, P. and Rebaic, A., (2007), "Combinatorial particle swarm optimization (CPSO) for partitioned clustering problem", J. Applied Mathematics and Computation, Vol. 192, Issue 2, 15 pp. 337-345.
[110] Felix T. S. Chan 1, Kumar, V. and Mishra, N., (2007), "A CMPSO algorithm based approach to solve the multi plant supply chain problem", Swarm lntell. : Focus on Ant and Particle Swarm Optimization, pp. 532.
[111] Subrarnanyam, V. Srinivasan, D. and Oniganti, R. (2007) "Dual layered PSO Algorithm for evolving an Artificial Neural Network controller", IEEE/CEC, pp. 2350-2357.
[112] Liao, c.-y., Lee, W.-P., Chen, X., (2007), "Dynamic and adjustable particle swarm optimization, Proc. of the 8th Conf. on 8th WSEAS Int. Conf. on Evolutionary Computing, Vol. 8.
[113] Kulkarni, R.V. and Venayagamoorthy, G.K., {2007), "An Estimation of Distribution Improved Particle Swarm Optimization Algorithm", Proc. IEEE/ ISSNIP, pp. 539-544.
[114]Lee, T.-Y., (2007b), "Optimal Spinning Reserve for a Wind-Thermal Power System Using EIPSO", IEEE/ TPWRS, Vol. 22, No.4, pp. 1612 -1621.
[115] Wei, C., He, Z., Zhang, Y., and Pei, W. (2002). "Swarm directions embedded in fast evolutionary programming". In Proc. of the IEEE/CEC, pp. 1278-1283.
[116] Lam, H., T., Nikolaevna, P., N. and Quan N., T., M. (2007)” The Heuristic Particle SwarmOptimization” Proc. of annual Conf. on Genetic and evolutionary computation in Ant

colony optimization, swarm Intell., and artificial immune systems,”GECCO’07”, pp.174 – 174.

[117] Lam, H., T., Nikolaevna, P., N. and Quan N., T., M. (2007)" The Heuristic Particle Swarm Optimization" Proc. of annual Conf. on Genetic and evolutionary computation in Ant colony optimization, swarm Intell, and artificial immune systems,"GECC0'07", pp.174-174.

[118] McNabb, A., W., Monson, C., K. and Seppi, K., D., (2007), "MRPSO: Map Reduce particle swarm optimization", Proc. of the 9th annual Conf. on Genetic and evolutionary Com put., pp. 177 -177.

[119] Yuan, L. and Zhao, Z.-0., (2007), "A Modified Binary Particle Swarm Optimization Algorithm for Permutation. Flow Shop Probiem", IEEE/ICMLC, Vol.2, pp.902-907.

[120] Li, H.-Q., and Li, L., (2007), "A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems", Proc. IEEE/IPC, pp. 94-97.

[121] Jang, W.-S., Kang, H.-1., Lee, B.-H., Kim K.-1., Shin, D.-1. and Kim, S.-C., (2007) "Optimized fuzzy clustering by predator prey particle swarm optimization" In IEEE/CEC, pp. 3232-3238.

[122] Pant, M. Radha, T., and Singh, V.P. (2007), "A New Particle Swarm Optimization with Quadratic
Interpolation", Int. IEEE Conf. on Computational lntell. and Multimedia Applications, Vol.1, pp.55-60.

[123] Wang, H., and Qian, F., {2007), "An improved particle swarm optimizer with shuffled sub-swarms and its application in soft-sensor of gasoline endpoint", Proc. Advances in Intelligent Systems Research.

[124] Gheitanchi, S., Ali, F.H. and Stipidis, E. (2008) "Trained Particle Swarm Optimization for Ad-hoc Collaborative Computing Networks" Swarm lntell. Algorithms and Applications Symp., ASIB 2008, UK.

[125] Wei, K., Zhang, T., Shen, X. and Liu, J., (2007), "An Improved Threshold Selection Algorithm Based on Particle Swarm Optimization for Image Segmentation", Proc. IEEE/ICNC.

[126] Yang, W.-P., (2007), "Vertical Particle Swarm Optimization Algorithm and its Application in Soft-Sensor Modeling", Int. Conf. on Machine Learning and Cyber. (IEEE/ ICMLC), Vol.4, pp.1985-1988.

[127] Hui Wang, Yong Liu, Sanyou Zeng , Hui Li and Changhe Li, "Opposition based Particle swarm Algorithm with Cauchy Mutation", IEEE 2007.

[128] Tan, Y., Xiao, Z.M.: Clonal particle swarm optimization and its applications. In: Proceedings of IEEE Co11gress on Evolutionary Computation, pp. 2303-2309 {2007)

[129] Yang, B., Chen, Y., Zhao, Z., “A hybrid evolutionary algorithm by combination of PSO and GA for unconstrained and constrained optimization problems”, Proceedings of the IEEE International Conference on Control and Automation, 2007, pp. 166–170.

[130] Zhang, Y.-N., Hu, Q.-N. andTeng, H.-F., {2008), "Active target particle swarm optimization: Research Articles", J. of Concurrency and Computation: Practice & Experience, Vol.20, No.1, pp.29 -40.

[131] Pant, M., Thangaraj, R. and Abraham, A., {2008), "Particle Swarm Optimization Using Adaptive Mutation", IEEE/DEXA'08, pp. 519-523.

[132] Yao, X., (2008), "Cooperatively Coevolving Particle Swarms for Large Scale Optimization", Conf. of EPSRC, Artificial Intel!. Technologies New and Emerging Computer Paradigms.

[133] Moraglio, A., Di ·Chio, C., Togelius, J. and Poli, R. {2008), "Geometric Particle Swarm Optimization", J. of Artificial Evolution and Applications.

[134] Lin, C., Liu Y., and Lee, C., (2008) "An efficient neural fuzzy network based on immune particle swarm optimization for prediction and control applications", J. of Innovative Computing, Information and Control Vol.4, No.7, pp.1711-1722:

[135] Zhiming, L., Cheng, W. and Jian L., {2008), "Solving constrained optimization via a modified genetic particle swarm optimization", Proc. Of lnt. Conf.On Forensic applications and techniques in telecommunications, information, and multimedia and workshop, No. 49

[136 I Ho, S.-Y., Lin, H.-S., Liauh, W.-H., and Ho, S.-J., (2008), "OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems" IEEE Trans. on Systems, Man and Cyber., Part A, Vol. 38, No., 2, pp.288-298

[137] Higashitani, M., lshigame, A., and Yasuda, K., {2008) "Pursuit-Escape Particle Swarm Optimization", Trans. On Electrical and Electronic Eng., {IEEJ), Vol.3, No.1, pp.136 -142.

[138] Lu, H. and Chen, W. {2008), "Self-adaptive velocity particle swarm optimization for solving constrained optimization problems", J. of Global Optimization, Vol.41, No.3, pp. 427-445.

[139] Xu, F., and Chen, W., {2006), "Stochastic Portfolio Selection Based on Velocity Limited Particle Swarm Optimization", Proc. IEEE/ WCICA, Vol. 1, pp.3599-3603.

[140] Montes deOca,M. A., St·· utzle, T., Birattari,M., and Dorigo, M. (2009). Frankenstein's PSO: A composite particle swarm optimization algorithm. IEEE Transactions on Evolutionary Computation, 13(5):1120-1132.

[141] Zhan, Z., Zhang, J., Li, Y., and Chung, H. (2009). Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39(6):1362-1381.

[142] George I.Evers and Mounir Ben Gallia, “Regrouping Particle Swarm Optimization: A new Global Optimization Algorithm with Improved Performance Consistency Across Benchmark” M.S. Thesis, Electrical Engineering Department, The University of Texas-Pan American,Edinburg,TX,2009.

[143] Premalatha, K., Natarajan, A.M., “Discrete PSO with GA operators for document clustering”, International Journal of Recent Trends in Engineering 1 (2009) 20–24.

[144] Z. C. Li, D.J. Zheng, and H.J. Hou, "A hybrid particle swarm optimization algorithm based on nonlinear simplex method and tabu search," in Advances in Neural Networks-ISNN 2010, L. Q. Zhang, B. L. Lu, and J. Kwok, Eds., vol. 6063 of Lecture Notes in Computer Science, pp. 126-135, Springer, Berlin, Germany, 2010. [145] S. Nakano, A. lshigame, and K. Yasuda, "Consideration of particle swarm optimization combined with tabu search," Electrical Engineering in Japan, vol. 172, no. 4, pp. 31-37, 2010.

[146] Daneshyari, M., & Yen, G. G. {2010). Cultural based Multiobjective Particle Swarm Optimization. IEEE Transactions on Systems, Man and Cybernetics, 41(2).

[147] Abdel-Kader, R.F., “Genetically improved PSO algorithm for efficient data clustering”, Proceedings of International Conference on Machine Learning and Computing, pp. 71–75,2010.

[148] Xinchao, Z. A perturbed particle swarm algorithm for numerical optimization. Applied Soft Computing, 10(1):119–124, 2010.

[149] H. Z. Tang, Y. Xiao, H. Huang, and X. Guo (2010) A novel dynamic particle swarm optimization algorithm based on improved artificial immune network, in Proceedings of the 10th IEEE International Conference on Signal Processing (ICSP ’10), B. Z. Yuan, Q. Q. Ruan, and X. F. Tang, Eds., pp. 103–106,New York, NY, USA.

[150] Tang, Y.,Wang, z., and Fang, J.-a Feedback learning particle swarm optimization. Applied Soft Computing, 11{8):4713-4725, 2011.

[151] Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., and Tian, Q. (2011). Self-adaptive learning based particle swarm optimization. lnformation Sciences, 181{20):4515-4538.

[152] T. Zhang, Y. J. Zhang, Q. P. Zheng, and P.M.Pardalos, “A hybrid particle swarm optimization and tabuseach algorithm for order planning problems of steel factories based on the make-to stock and make-to-order management architecture,” Journal of Industrial and Management Optimization, vol. 7, no. 1, pp. 31–51, 2011.

[153] Changhe, L., Shengxiang,Y., and TrungThanh, N. (2012).A self-learning particle swarm optimizer for global optimization problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 42(3):627- 646.

[154] Z.H. Zhan, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (6) {2011) 832-847.

[155] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu, “Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions,” *Information Sciences*, vol. 181, no. 20, pp. 4539–4549, 2011.

[156] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, “Enhancing particle swarm optimization using generalized opposition-based learning,” *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.

[157] S.-M. Chen and C.-Y. Chien, “Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques,” Expert Systems with Applications, vol. 38, no. 12, pp. 14439–14450, 2011.

[158] Hu,M., Wu, T., andWeir, J. D. (2012).An intelligent augmentation of particle swarm optimization with multiple adaptive methods. Information Sciences, 213:68-83.

[159]C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man Cybern. B: Cybern. 42 (3) (2012)627-646.

[160]C. S. Li, J. Zhou, P. Kou, and J. Xiao, “A novel chaotic particle swarm optimization based fuzzy clustering algorithm,”Neuro computing, vol. 83, pp. 98–109, 2012.

[161]N. Dong, C.-H. Wu, W.-H. Ip, Z.-Q. Chen, C.-Y. Chan, and K.-L. Yung, “An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection,” *Computers & Mathematics with Applications*, vol. 64, no. 6, pp. 1886–1902, 2012.

[162] N. K. Khan, A. R. Baig, and M. A. Iqbal, “Opposition-based discrete PSO using natural encoding for classification rule discovery,” *International Journal of Advanced Robotic Systems*, vol. 9, 2012.

[163] Leu MS, Yeh MF. Grey particle swarm optimization. Applied Soft Computing. 2012; 12(9):2985–96. doi: 10.1016/j.asoc.2012.04.030

[164]Bonyadi, M. R., Li, X., and Michalewicz, Z. (2013).A hybrid particle swarm with velocity mutation for constraint optimization problems. In Genetic and Evolutionary Computation Conference (GECCO), pp. 1-8.

[165] Elsayed, S. M., Sarker, R. A., and Mezura-Montes, E. (2013). Particle swarm optimizer for constrained optimization. In IEEE Congress on Evolutionary Computation, pp. 2703-2711.

[166] R.Ktari and H.Chabchoub, "Essential particle swarm optimization queen with tabu search for MKP
Resolution," Computing, vol. 95, no. 9, pp. 897-921, 2013.

[167] X. Jin, Y.Q. Liang, D.P. Tian, F.Z. Zhuang, Particle swarm optimization using dimension selection methods, Appl. Math. Comput. 219 (10) (2013)5185-5197.

[168] M.S. Kiran, M. Gndz, A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems, Appl. Soft Com put. 13 (4) (2013) 2188-2203.

[169] E. Aminian and M. Teshnehlab, “A novel fuzzy particle swarm optimization,” in Proceedings of the 13th Iranian Conference on Fuzzy Systems, pp. 1–6, IEEE, Qazvin, Iran, 2013.

[170] Z.-H. Liu, J. Zhang, S.-W. Zhou, X.-H. Li, and K. Liu, “Co evolutionary particle swarm optimization using AIS and its application in multi parameter estimation of PMSM,” IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1921–1935, 2013.

[171] C.-L. Huang, W.-C. Huang, H.-Y. Chang, Y.-C. Yeh, and C.- Y. Tsai, “Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering,” Applied Soft Computing Journal, vol. 13, no. 9, pp. 3864–3872, 2013.

[172] H. Jiang and L. Zou, “A hybrid PSO-SA optimizing approach for SVM models in classification,” International Journal of Biomathematics, vol. 6, no. 5,Article ID 1350036, 18 pages, 2013.

[173] M. El-Abd, “Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’13), pp. 2215–2220, IEEE, Canc´un, Mexico, June 2013.

[174] T. K. Sharma, M. Pant, and A. Abraham, “Blend of local and global variant of PSO in ABC,” in Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC’13), pp. 113–119, August 2013.

[175] G. Maione and A. Punzi, “Combining differential evolution and particle swarm optimization to tune and realize fractional order controllers,” Mathematical and Computer Modelling of Dynamical Systems, vol. 19, no. 3, pp. 277–299, 2013.

[176] E. Davoodi, M. T. Hagh, and S. G. Zadeh, "A hybrid improved quantum-behaved particle swarm optimization simplex method (IQPSOS) to solve power system load flow problems," Applied Soft Computing Journal, vol. 21, pp. 171-179, 2014.

[177] Zahra Beheshti ,SitiMariyamHj. Shamsuddin, CAPSO: Centripetal accelerated particle swarm optimization, Information Sciences: an International Journal, 258, p.54-79, February, 2014

[178] P. Li and H. Xiao, “An improved quantum-behaved particle swarm optimization algorithm,” *Applied Intelligence*, vol. 40,no 3, pp. 479–496, 2014.

[179] S. Darzi, T. S. Kiong, M. T. Islam, M. Ismail, S. Kibria, linear constraint minimum variance assisted by particle swarm optimization,” The Scientific World Journal, vol. 2014, Article ID 724639, 10 pages, 2014.

[180] Y. Zhang, S.Wang, Y. Sun, G. Ji, P. Phillips, and Z.Dong, “Binary structuring elements decomposition based on an improved recursive dilation-union model and RSAPSO method,” Mathematical Problems in Engineering, vol. 2014, Article ID 272496, 12 pages, 2014.

[181] X. Yu, J. Cao, H. Shan, L. Zhu, and J. Guo, “An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization,” The Scientific World Journal, vol. 2014,Article ID 215472, 16 pages, 2014.

[182]G.-G. Wang, A. HosseinGandomi, X.-S. Yang, and A. HosseinAlavi, “A novel improved accelerated particle swarm optimization algorithm for global numerical optimization,” Engineering Computations, vol. 31, no. 7, pp. 1198–1220, 2014.

[183] A. Yadav and K. Deep, “An efficient co-swarm particle swarm optimization for non-linear constrained optimization,” Journal of Computational Science, vol. 5, no. 2, pp. 258–268, 2014.

[184] W. Guo, W. Li, Q. Zhang, L. Wang, Q. Wu, and H. Ren, “Biogeography-based particle swarm optimization with fuzzy elitism and its applications to constrained engineering problems,” Engineering Optimization, vol. 46, no. 11, pp. 1465–1484, 2014.

[185] X. Yu, X. Zhang, Enhanced comprehensive learning particle swarm optimization, Appl. Math. Comput., 242 (2014) 265-276.

[186] HusevinHakli,HarunUguz,A novel particle swarm optimization with levy flight ,Applied soft computing,23,p.333-345,October 2014.

[187] Wei Hong Lim,NorAshidi Mat Isa ,Teaching and peer-learning particle swarm optimization ,Applied soft computing.18,n.c,p.39-58,May 2014

[188] Jun Sun, Xiaojun Wu, Vasile Pa lade, Wei Fang, YuhuiShi. Random drift particle swarm optimization algorithm: convergence analysis and parameter selection. Mach Learn (2015) 101:345-376.

[189] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (6) (2015) 43-60.

[190] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Computing. 24 (2015) 11-24.

[191] L.M. Zhang, Y.G. Tang, C.C. Hua, X.P. Guan, A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques, Appl. Soft Computing. 28 (2015) 138-149.

[192] M.R. Tanweer, S. Suresh, N. Sundararajan, Self- regulating particle swarm optimization algorithm, Inf. Sci. 294(10)(2015) 182-202.

[193] A. Rezaee Jordehi, J. Jasni, N. AbdWahab, M.Z. Kadir, M.S. Javadi, Enhanced leader PSO (ELPSO): a new Algorithm for allocating distributed TCSC's in power systems, Int. J. Electric. Power Energy Syst., 64 (2015) 771- 784.

[194] Wang Ll,ZhangJianfeng ,Li Xin,SunGuoqiang,"Particle Swarm Optimization Algorithm Based on Two Swarm Evolution, IEEE 27th Chinese Control and Decision Conference (CCDC),1200-1205,2015.

[195] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, “Feature selection algorithm based on bare bones particle swarm optimization,”Neurocomputing, vol. 148, pp. 150–157, 2015.

[196] J. Geng, M.-W. Li, Z.-H. Dong, and Y.-S. Liao, “Port throughput forecasting by MARS-RSVR with chaotic simulated annealing particle swarm optimization algorithm,” Neurocomputing, vol. 147, pp. 239–250, 2015.

[197] L. N. Vitorino, S. F. Ribeiro, and C. J. A. Bastos-Filho, “A mechanism based on artificial bee colony to generate diversity in particle swarm optimization,” Neurocomputing, vol. 148, pp. 39–45, 2015.

[198] C. Pornsing, M.S. Sodhi, B.F. Lamond, Novel self-adaptive particle swarm optimization methods, Soft Computing. 20 (2016) 3579-3593.

[199] Y.J. Gong, J.J. Li, Y.C. Zhou, Y. Li, H.S.H. Chung, Y.H. Shi, J. Zhang, Genetic learning particle swarm Optimization, IEEE Trans. Cyber. 46 (10) (2016)2277-2290.

[200] QuandeQin,ShiCheng,QingyuZhang,YuhuiShi, Particle Swarm Optimization with lnter-swarm Interactive Learning Strategy IEEE TRANSACTIONS ON CYBERNETICS, VOL. 46, NO. 10, 2016, PP 2238-2251.

[201] C. Zeng and _Y.Shen," Distribution-Guided bare-bones particle swarm optimization," I 2th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016.

[202] ZhenlunYang,AngusWu, Haihualiao,Jianxin Xu," An elitist promotion quantum-behaved particle swarm Optimization algorithm, 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, 348-30, 2016.

[203] M.S. Hossain, M. Moniruzzaman, G. Muhammad, A. Ghoneim, A. Alamri, Big data-driven service composition using parallel clustered particle swarm optimization in mobile environment, IEEE Trans. Serv. Comput. 9 (5) (2016) 806–817.

[204] X.W. Xia, C.W. Xie, B. Wei, Z.B. Hu, B.J. Wang, C. Jin, Particle swarm optimization using multi-level adaptation and purposeful detection operators, Inf. Sci. 385 (2017) 174-195.

[205]J. Guo and Yuji Sato, A Hierarchical Bare Bones Particle Swarm Optimization Algorithm, IEEE International Conference on Systems, Man, and Cybernetics (SMC) Banff Centre, Banff, Canada, October 5-8,pp.1936- 1941, 2017.

[206] Munlin and Anantathanavit New Social-Based Radius Particle Swarm Optimization in the Proceedings of the12th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2017.PP 838-843.

[207] Dada Emmanuel Gbenga, Joseph Stephen Bassi, OkunadeOluwasogoAdekunle ,Primal-dual Asynchronous Particle swarm optimization (pdAPSO) hybrid metaheuristic algorithm for solving global optimization problems American journal of Engineering research (AJER) Vol-6,issue-4,pp 60-71,2017.

[208] Amjad A. Hudaib1 & Ahmad K. Al Hwaitat2,Movement Particle Swarm Optimization, Modern Applied Science; Vol. 12, No. 1; 2018, Published by Canadian Centre of Science and Education.

[209] Anwar Ali Yahya, Centroid Particle Swarm Optimization for high dimension data classification, Journal of Experimental &TheoreticalArtificialIntelligence,Vol 30(6),pp 857-886,2019

[210] Hassan Koyuncu and RahimeCeylan, A PSO based approach: Scout particle swarm optimization for continuous global optimization problems, Journal of computational Design and Engineering, vol 6,issue 2,p.129-142.

[211] ShakhnazAkhmedova , Vladimir Stanovov, and Eugene Semenkin," Success-History Based Position Adaptation in Co-operation of Biology Related Algorithms, Springer Nature, Switzerland AG, pp 39-49,2019.

[212] DongpingTian, XiaofeiZhao,Z Shi," DGMPSO: Diversity-Guided Multi-Mutation PSO"IEEE Access, Vol 7,124008-124025, 2019.

[213] Xuewen Xia, Ling Gui,FeiYu,HongrunWu,Bowei,YING-Long Zhang and Zhi-Hui Zhan, Triple Archives Particle Swarm Optimization, IEEE Trans. On cybernetics, 2019.DOI 10.1109/TCYB.2019.2943928

[214] Xu L, Muhammad A, Pu Y, Zhou J, Zhang Y, Fractional-order quantum Particle swarm optimisation, PLoSone,Jun 20,14(6),2019.

[215] Zhang J,Zhu X,Wang Y,Zhou M,Dual-Environnemental PSO in noisy and noise free environments, IEEE Trans. Of Cybernatics, 49(6),2019.

[216] Hu X, Eberhart RC (2002b) Solving constrained nonlinear optimization problems with particle swarm optimization. In: Proceedings of the sixth world multiconference on systemics, cybernetics and informatics 2002 (SCI 2002), Orlando, USA

[217] Parsopoulos KE, Vrahatis MN (2002c) Particle swarm method for constrained optimization problems. In: Proceedings of the Euro-international symposium on computational intelligence 2002

[218] Coath, G., and Halgamuge, S. K. (2003). A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems. In IEEE Congress on Evolutionary Computation, pp. 2419–2425.

[219] Pulido, G. T., and Coello, C. A. C. (2004). A constraint-handling mechanism for particle swarm optimization. In IEEE Congress on Evolutionary Computation, pp. 1396–1403.

[220] Zavala AEM, Diharce ERV, Aguirre AH (2005) Particle evolutionary swarm for design reliability optimization. In: Coello Coello CA, Aguirre AH, Zitzler E (eds) Evolutionary multi-criterion optimization. Third international conference, EMO 2005. Lecture notes in computer science, vol 3410. Springer, Guanajuato, Me´xico, pp 856–869, March 2005.

[221] Liang, J.J. and Suganthan, P.N. (2006) Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism, In IEEE Congress on Evolutionary Computation.

[222] Bochenek, B.; Forys, P. Structural optimization for post-buckling behavior using particle swarms. *Structural and Multidisciplinary Optimization*, v. 32, n. 6, p. 521–531, Dec 2006. ISSN 1615-1488.

[223]Aguirre, A. H., Munoz Zavala, A. E. and Diharce, E. V., and Botello Rionda, S. (2007), "COPSO: Constraints Optimization via PSO algorithm", Comunicacion Tecnic, (CC/Cl MAT).

[224] Juan C.FuentesCabrera,CarlosA.CoelloCoello,Handling constraints in Particle swarm optimization using a small population size,MICAI,2007,Advances in Artificial intelligence ,pp 41-51,2007

[225] Flores-Mendoza J, Mezura-Montes E (2008) Looking inside particle swarm optimization in constrained search spaces. In: MICAI 2008: advances in artificial intelligence, pp 451–461

[226] C. Sun, J. Zeng, J. Pan, an New Vector Particle Swarm Optimization for Constrained Optimization Problems, in: Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization, 2009, pp. 485–488

[227] Liang, J., Zhigang, S., and Zhihui, L. {2010). Co-evolutionary comprehensive learning particle swarm optimizer. In IEEE Congress on Evolutionary Computation, pp. 1-8.

[228] Sun, C.-l., Zeng, J.-c., and Pan, J.-s. (2011).an improved vector particle swarm optimization for constrained optimization problems. Information Sciences, 181(6):1153–1163..

[229] M. Daneshyari and G. G. Yen, “Constrained multiple-swarm particle swarm optimization within a cultural framework,” IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, vol. 42, no. 2, pp. 475–490, 2012.

[230] M. H. Afshar, “Extension of the constrained particle swarm optimization algorithm to optimal operation of multireservoirs system,” International Journal of Electrical Power & Energy Systems, vol. 51, pp. 71–81, 2013.

[231] G. Koulinas, L. Kotsikas, and K. Anagnostopoulos, “A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem,” Information Sciences, vol. 277, pp. 680–693, 2014.

[232] N. Singh, R. Arya, and R. K. Agrawal, “A novel approach to combine features for salient object detection using constrained particle swarmoptimization,” Pattern Recognition, vol. 47, no. 4, pp. 1731–1739, 2014.

[233] L. Z. Cui, Z. Ling, J.Poonet al., “Aparallelmodel of independent component analysis constrained by a 5-parameter reference curve and its solution by multi-target particle swarm optimization,” Analytical Methods, vol. 6, no. 8, pp. 2679–2686, 2014.

[234] Y. Y. Shou,Y.Li, andC.T.Lai, “Hybrid particle swarm optimization for preemptive resource-constrained project scheduling,”Neuro -computing, vol. 148, pp. 122–128, 2015.

[235] Biwei Tang , Zhanxia Zhu,JianjunLuo ,A framework for constrained optimization problems based on a modified Particle Swarm Optimization, Mathematical Problems in Engineering, vol 2016,Article ID 8627083.,Hindawi,2016

[236] Shaotian Lu, Jingdong Zhao , Li Jiang, Hong Liu, Solving the time jerk optimal Trajectory Planning problem of a Robot using Augmented Lagrange Constrained Particle Swarm Optimization,Mathematical Problems in Engineering, vol 2017,Article ID 1921479.,Hindawi,2017

[237] Lejun Ma, Huan Wang ,Baohong Lu, Changjun Qi, Application of strongly constrained space Particle swarm optimization to optimal operation of a Reservoir system, Sustainability 2018,10,4445.

[238] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," i� Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, 2002, pp. 1051-1056.

[239] Parsopoulos, K.E. and Vrahatis, M.N (2002) Particle Swam Optimization Method in Multi-objective Problems, In Proceedings of the ACM Symposium on Applied Computing. pp.: 603 – 607.

[240] T. Ray and K. Liew, "A swarm metaphor for multi-objective design optimization," *Engineering Optimization*, vol. 34, no. 2, pp. 141–153, Mar. 2002.

[241] Omkar, S. N.,Mudigere,D., NarayanaNaik, G., and Gopalakrishnan, S., (2008), "Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures", J. Computers and Structures, Vol. 86, No.1-2. pp.1-14.

[242] Hu, X. and Eberhart, R. C. (2002) "Multi objective optimization using dynamic neighbourhood particle swarm optimization". Proc. of the IEEE/ dynamic neighbourhood particle swarm optimization". Proc. of the IEEE/ CEC, pp. 1677-1681.

[243] Hu, X., Eberhart, R.C. and Shi, Y. Particle Swarm with Extended Memory for Multi objective Optimization, In Proceedings of the IEEE Swarm Intelligence Symposium. pp.: 193-197, 2003.

[244]Ji, C., Zhang, Y., Gao, SH., Yuan, P. and Li, Zh, "Particle swarm optimization. for mobile ad hoc networks clustering", IEEE Int. Conf. on Networking, Sensing and Control, Vol 1, pp. 372 -375, 2004.

[245] He, S.,Wu, Q.H., Wen, J.Y., Saunders, J.R. and Paton, R.C., (2004), "A particle swarm optimizer with passive congregation", J. of Biosystems, Vol.78, No.1-3, pp. 135-147.

[246] Baumgar.tner, U, Magele, Ch. and Ren hart, W. (2004) Pareto Optimality and Particle Swarm Optimization, In IEEE Transactions on Magnetics, 40(2) pp.: 1172-1175

[247] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm Optimization," lEEE Transactions on Evolutionary Computation, vol. 8, pp. 256-279, Jun 2004.

[248] M. R. Sierra and C. A. C. Coello,-"Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance," in international Conference on Evolutionary Multi-Criterion Optimization, 2005, pp. 505- 519.

[249] Liu, H., Ajith, A., Okkyung, C. and Hwan Moon, S., (2006), "Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-Shop Scheduling Problems", Int. Conf. on Simulated evolution and learning, Seal 2006, Vol. 4247, pp. 197-204.

[250] Abido, M.A (2007) Two-Level of Non- dominated Solutions Approach to Multi objective Particle Swarm Optimization, In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. pp.: 726 -733.

[251] X. W. Zheng and H. Liu, "A scalable co-evolutionary multi-objective particle swarm optimizer," International Journal of Computational Intelligence Systems, vol. 3, pp. 590-600, Oct 2010.

[252] Z. Mart, S. nez, and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on Decomposition," in Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July, 2011, pp. 69-76.

[253] A. A. Ibrahim, A. Mohamed, H. Shareef, and S. P. Ghoshal, "Optimal power quality monitor placement in power systems based on particle swarm optimization and artificial immune system," in Proceedings of the 3rd Conference on Data Mining and Optimization (DMO '11), pp. 141–145, June 2011.

[254] A.A. Mousa, M.A. EI-Shorbagy and W.F. Abd-EI-Wahed, "Local search based hybrid particle swarm Optimization algorithm for Mu!ti -objective optimization," Swarm Evolutionary Computation, vol. 3, pp. 1-14, 2012.

[255] A.A. EI-Sawy, Z.M. Hendawy, M.A. EI-Shorbagy, "Trust-Region Algorithm based local search for Multi- objective Optimization", IEEE 1st International Conference on Innovative Engineering Systems {IEEE-RAS ICIES2012), Alexandria, Egypt, December, 7-9, 2012.

[256]Y. Zhang, D.-W. Gong, and Z. Ding, “A bare-bones Multiobjective particle swarm optimization algorithm for environmental/ economic dispatch,” Information Sciences, vol. 192, pp. 213–227, 2012.

[257] S. A. Khan and A. P. Engelbrecht, “A fuzzy particle swarm optimization algorithm for computer communication network topology design,” Applied Intelligence, vol. 36, no. 1, pp. 161–177, 2012.

[258] C. Y. Qiu, C. L. Wang, and X. Q. Zuo, “A novel multi-objective particle swarm optimization with K-means based global best selection strategy,” International Journal of Computational Intelligence Systems, vol. 6, no. 5, pp. 822–835, 2013.

[259] Z. H. Zhan et al., “Multiple populations for multiple objectives: A coevolutionary technique for solving multi objective optimization problems,” IEEE Trans. Cybern., vol. 43, no. 2, pp. 445–463, Apr. 2013.

[260] W. Elloumi, N. Baklouti, A. Abraham, and A. M. Alimi, “The multi-objective hybridization of particle swarm optimization and fuzzy ant colony optimization,” Journal of Intelligent and Fuzzy Systems, vol. 27, no. 1, pp. 515–525, 2014.

[261]W.-J. Chen, W.-C. Su, and Y.-L. Yang, “Application of constrained multi-objective hybrid quantum particle swarm optimization for improving performance of an ironless permanent magnet linear motor,” Applied Mathematics & Information Sciences, vol. 8, no. 6, pp. 3111–3120, 2014.

[262] S. Ganguly, “Multi-objective planning for reactive power compensation of radial distribution networks with unified power quality conditioner allocation using particle swarm optimization,” IEEE Transactions on Power Systems, vol. 29, no. 4, pp. 1801–1810, 2014.

[263]E. Z. Zhang, Y. F.Wu, andQ.W. Chen, “Apractical approach for solving multi-objective reliability redundancy allocation problems using extended bare-bones particle swarm optimization,” Reliability Engineering and System Safety, vol. 127, pp. 65–76, 2014.

[264] S. Cheng, M.-Y. Chen, and P. J. Fleming, “Improved Multi objective particle swarm optimization with preference strategy for optimal DG integration into the distribution system,”Neuro computing, vol. 148, pp. 23–29, 2015.

[265] Shakhnaz Akhmedova and Eugene Semenkin,” Co-operation of Biology related Algorithms for Constrained Multi-Objective optimization,” Advances in swarm and computational intelligence, pp 487-494, 2015.

[266] Xiang Yu and Zhang Xueqing,”Multi swarm Comprehensive learning particle swarm optimization for solving multi objective optimization problems”, Plos one, 1-21, 2017.

[267] Zhu Q, Lin Q, Chen W, Wong KC, Coello Coello CA, Li J, Chen J, Zhang J,” An External Archive-Guided Multi objective PSO algorithm”, IEEE Trans Cybern,47(9),2794-2808,2017.

[268] Yue C, Qu B, Liang J (2018) A multi-objective particle swarm optimizer using ring topology for solving multimodal multi objective problems. IEEE Trans Evolut. Comput. (in press)

[269] Han H,LuW,ZhangL,Qiao J,” Adaptive gradient multi objective PSO “,IEEE Trans Cybern,48(11),3067-3079,2018.

[270] Han H, Wu X, Zhang L, Tian Y, Qiao J,” Self-Organising RBF Neural network using an Adaptive Gradient Multi-objective Particle swarm optimization “IEEE Trans Cybern,49(1),69-82,2019.

[271] Zhiming Lv, Linqing Wang, Zhongyang Han, Jun Zhao, Wei Wang, Surrogate-Assisted PSO algorithm with Parento active learning for expensive multi-objective optimization, IEEE/CAA Journal of Automatica, SINICA, Vol 6,no 3,2019.

[272]Hui Song, A.H Qin, Pei-Wei Tsai, J.J Ling, Multitasking Multi-swarm optimization,IEEE,P.1937-1944,2019.

[273] Selvi, S.; Manimegalai, D.: Task scheduling using two-phase variable neighborhood search algorithm on heterogeneous computing and grid environments. Arab. J. Sci. Eng. **40**(3), 817–844 (2015).

[274] The Apache Software Foundation. Apache Hadoop. http:// hadoop.apache.org/ (2018)

[275] Wickham, H.:Advanced R. Chapman and Hall/CRC TheRSeries. Taylor and Francis, Milton Park (2014)

[276] MATLAB and Simulink. https://in.mathworks.com/ (2018)

[277] The OpenMP API specification for parallel programming. http:// www.openmp.org/ (2018)

[278] Gorelick, M.; Ozsvald, I.: High Performance Python: Practical Performant Programming for Humans. O'Reilly Media, Sebastopol (2014)

[279] The Julia Language. https://docs.julialang.org/en/stable/manual/ parallel-computing. Julia Parallel Computing (2018)

[280] The OpenMP API specification for parallel programming. http:// www.openmp.org/ (2018)

[281] Gies, D. and Y. Rahmat-Samii. "Reconfigurable array design using parallel particle swarm optimization", Antennas and Propagation Society International Symposium, IEEE, 2003.

[282] Schutte, J.F, Reinbolt, J.A, Fregly, B.J, Haftka, R.T, George, A.D, "Parallel Global Optimization with the Particle Swarm Algorithm", Int J Numer Meth Eng 61(13):2296–2315, John Wiley and Sons Ltd, Great Britain, 2004.

[283] Venter, G., Sobieszczanski-Sobieski, J., "A parallel particle swarm optimization algorithm accelerated by asynchronous evaluations", 6th world congresses of structural and multidisciplinary optimization, 30 May–03 June, 2005.

[284] Chang, J-F., Chu, S-C., Roddick, J.F., Pan, J.S., "A parallel particle swarm optimization algorithm with communication strategies", J Information SciEng 21:809–818, 2005.

[285] Chusanapiputt, S.; Nualhong, D.; Jantarang, S.; Phoomvuthisarn, S.: Relative velocity updating in parallel particle swarm optimization based lagrangian relaxation for large-scale unit commitment problem. In: IEEE Region 10 Conference, Melbourne, Qld., Australia, pp. 1–6 (2005)

[286] B.-I. Koh, A. D. George, R. T. Haftka, and B. J. Fregly, "Parallel asynchronous particle swarm optimization," International Journal for Numerical Methods in Engineering, vol. 67, no. 4, pp. 578–595, 2006.

[287] McNabb, A.W.; Monson, C.K.; Seppi, K.D.: Parallel PSO using MapReduce. In: IEEE Congress on Evolutionary Computation, pp. 7–14 (2007)

[288] the Apache Software Foundation. Apache Hadoop. http:// hadoop.apache.org/ (2018)

[289] Liu, Q.;Li,T.;Liu,Q.; Zhu, J.;Ding, X.;Wu, J.:Two phase parallel particle swarm algorithm based on regional and social study of object optimization. In: Third IEEE International Conference on Natural Computation, vol. 3, pp. 827–831 (2007)

[290] Wang, D.; Wu, C.H.; Ip, A.; Wang, D.; Yan, Y.: Parallel multi population particle swarm optimization algorithm for the uncapacitated facility location problem using openMP. In: IEEEWorld Congress on Computational Intelligence Evolutionary Computation, pp. 1214–1218 (2008)

[291] Waintraub, M., Schirru, R., Pereira, C. M. N. A., "Multiprocessor modeling of parallel Particle Swarm Optimization applied to nuclear engineering problems," Progress in Nuclear Energy, vol. 51, no. 6, pp. 680–688, 2009.

[292] Jeong, H.M.; Lee, H.S.; Park, J.H.: Application of parallel particle swarm optimization on power system state estimation. In: Transmission and Distribution Conference and Exposition: Asia and Pacific, pp. 1–4 (2009)

[293] Singhal, G.; Jain, A.; Patnaik,A.: Parallelization of particle swarm optimization using message passing interfaces (MPIs). In: IEEE World Congress on Nature and Biologically Inspired Computing, pp. 67-71 (2009)

[294] Kalivarapu, V.; Foo, J.L.; Winer, E.: Synchronous parallelization of particle swarm optimization with digital pheromones. Adv. Eng. Softw. **40**(10), 975–985 (2009)

[295] Lorion, Y.; Bogon, T.; Timm, I.J.; Drobnik, O.: An agent based parallel particle swarm optimization-APPSO. In: IEEE Swarm Intelligence Symposium, pp. 52–59 (2009)

[296] Farmahini-Farahani, A.; Vakili, S.; Fakhraie, S.M.; Safari, S.; Lucas, C.: Parallel scalable hardware implementation of asynchronous discrete particle swarm optimization. Eng. Appl. ArtifIntell. **23**(2), 177–187 (2010)

[297] Li, B.; Wada, K.: Communication latency tolerant parallel algorithm for particle swarm optimization. Parallel Comput. **37**(1), 1–10 (2011)

[298]Tu K, Liang Z (2011) Parallel computation models of particle swarm optimization implemented by multiple threads. Expert SystAppl 38(5):5858–5866

[299] Aljarah, I.; Ludwig, S.A.: Parallel particle swarm optimization clustering algorithm based on MapReduce methodology. In: Fourth IEEEWorld Congress onNature and Biologically Inspired Computing, pp. 104–111 (2012)

[300] Parsopoulos, K.E.: Parallel cooperative micro-particle swarm optimization: a master slave model. Appl. Soft Comput. **12**(11), 3552–3579 (2012)

[301] S.-C. Yu, “Elucidating multiprocessors flow shop scheduling with dependent setup times using a twin particle swarm optimization,” Applied Soft Computing Journal, vol. 21, pp. 578– 589, 2014.

[302] Gulcu, S.; Kodaz, H.: A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization. Eng. Appl. Artif. Intell. **45**, 33–45 (2015)

[303] Zhang, G.W.; Zhan, Z.H.; Du, K.J.; Lin, Y.; Chen, W.N.; Li, J.J.; Zhang, J.: Parallel particle swarm optimization using message passing interface. In: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, vol. 1, pp. 55–64 (2015)

[304] Cao, J.; Cui, H.; Shi, H.; Jiao, L.: Big Data: a parallel particle swarm optimization-back-propagation neural network algorithm based on MapReduce. PLoS ONE **11**(6), e0157551 (2016)

[305] Tian, N.; Wang, Y.; Ji, Z.: Parallel coevolution of quantum behaved particle swarm optimization for high-dimensional problems In: Asian Simulation Conference, pp. 367–376 (2016)

[306] Nedjah, N.; Rogerio, M.C.; Luiza, M.M.: A fine-grained parallel particle swarm optimization on many core and multi-core architectures. In: International Conference on Parallel Computing Technologies, pp. 215–224 (2017)

[307] Eman Abdulaziz Abdullah, Ibrahim Ahmed Saleh, Omar Ibrahim Al Saif, Performance Evaluation of parallel particle swarm optimization for multicore environment, International conference on Advanced science and engineering(ICOASE)Kurdistan region ,Iraq ,p.81-86,2018

[308] Lai, X.; Zhou, Y.: An adaptive parallel particle swarm optimization for numerical optimization problems. Neural Comput. Appl. 1–19 (2018)

[309**]** nVIDIA.:nVIDIA CUDA Programming Guide v.2.3. nVIDIA Corporation, Santa Clara (2009)

[310**]** Nvidia Corp Website (2012) NVIDIA CUDA C Programming Guide, version 4.2. https ://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide .pdf

[311] Farber, R.: Parallel Programming with OpenACC. Morgan Kaufmann, Burlington (2017)

[312] Kaz, S.: An in-depth look at Google'first Tensor Processing Unit. https://cloud.google.com/blog/bigdata/2017/05/an-indepth-look-at-googles-first-tensor-processing-unit-TPU (2018)

[313] Zou, X.; Wang, L.; Tang, Y.; Liu, Y.; Zhan, S.; Tao, F.: Parallel design of intelligent optimization algorithm based on FPGA. Int. J. Adv. Manuf. Technol. **94**(9), 3399–3412 (2018)

[314] J. Li, D. Wan, Z. Chi, and X. Hu, "An Efficient Fine-Grained Parallel Particle Swarm Optimization Method Based on GPU-Acceleration," International Journal of Innovative Computing, Information and Control, vol.3, no. 6(B), p.1707-1714, December 2007.

[315] Zhou and Y. Tan, "GPU-based Parallel Particle Swarm Optimization," IEEE Congress on Evolutionary Computation, 2009. CEC '09., pp.1493–1500, May 2009.

[316] L. De P. Veronese and R. A. Krohling, "Swarm's flight: Accelerating the particles using C-CUDA," in Proceedings of the 2009 IEEE Congress on Evolutionary Computation, CEC 2009, pp. 3264–3270, nor, May 2009.

[317] Rymut, B., Kwolek, B., "GPU-supported object tracking using adaptive appearance models and Particle Swarm Optimization", Proceedings of the 2010 international conference on Computer vision and graphics: Part II, ICCVG'10, Springer-Verlag, Berlin, Heidelberg, 227–234, 2010.

[318] Mussi, L., Daolio, F., Cagnoni, S., "Evaluation of parallel particle swarm optimization algorithms within the CUDA™ architecture", In Information Sciences, Volume 181, Issue 20, 2011, Pages 4642-4657, ISSN 0020-0255.

[319] S. Solomon, P. Thulasiraman, and R. K. Thulasiram, "Collaborative multi-swarm PSO for task matching using graphics processing units," in Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO '11), pp. 1563–1570, July 2011.

[320] Zhu, H.; Guo, Y.;Wu, J.; Gu, J.; Eguchi, K.: Paralleling Euclidean particle swarm optimization in CUDA. In: 4th IEEE International Conference on Intelligent Networks and Intelligent Systems, pp. 93–96 (2011)

[321] Y. K. Hung and W. C. Wang, "Accelerating parallel particle swarm optimization via GPU," Optimization Methods and Software, vol. 27, no. 1, pp. 33–51, 2012.

[322] J. Kumar, L. Singh, and S.Paul, "GPU based parallel cooperative particle swarm optimization using C-CUDA: a case study," in Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '13), July 2013.

[323] O. Awwad, A. Al-Fuqaha, G. Ben Brahim, B. Khan, and A. Rayes, "Distributed topology control in large-scale hybrid RF/FSO networks: SIMT GPU-based particle swarm optimization approach," International Journal of Communication Systems, vol. 26, no. 7, pp. 888–911, 2013.

[324] D. L. Souza, O. N. Teixeira, D. C. Monteiro, and R. C. L. de Oliveira, "A new cooperative evolutionary multi-swarm optimizer algorithm based on CUDA architecture applied to engineering optimization," in Proceedings of the Combinations of Intelligent Methods and Applications, pp. 95–115, Springer, Berlin, Germany, 2013.

[325] Calazan, R.M.; Nedjah, N.; Luiza, M.M.: Parallel GPU-based implementation of high dimension particle swarm optimizations. In IEEE Fourth Latin American Symposium on Circuits and Systems, pp. 1–4 (2013)

[326] Chen, R.-B., Hsu, Y.-W., Hung, Y., Wang, W., "Discrete particle swarm optimization for constructing uniform design on irregular regions," Computational Statistics & Data Analysis, vol. 72, pp. 282–297, 2014.

[327] Shenghui, L.; Shuli, Z.: Research on FJSP based on CUDA parallel cellular particle swarm optimization algorithm. In: International IET Conference on Software Intelligence Technologies and Applications, pp. 325–329 (2014)

[328] Silva E H M,FilhoC.J.A.Bastos ,"PSO Efficient Implementation on GPUs Using Low Latency Memory," IEEE Latin America Transactions, Vol 13,No.5,,May,2015.

[329] Li, J.; Wang, W.; Hu, X.: Parallel particle swarm optimization algorithm based on CUDA in the AWS cloud. In: Ninth International Conference on Frontier of Computer Science and Technology, pp. 8–12 (2015)

[330] Dali Narjess and BouamamaSadok,"GPU-PSO Parallel PSO approaches on GPU for constraint reasoning:case of Max-CSPs,19th International conference of knowledge based and intelligent information and Engineering systems,Elsevier,pp 1070-1080,2015.

[331] Hussain,M.; Hattori, H.; Fujimoto, N.: A CUDA implementation of the standard particle swarm optimization. In: 18th IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Romania, pp. 219–226 (2016)

[332] Wu, Q.; Xiong, F.; Wang, F.; Xiong, Y.: Parallel particle swarm optimization on a graphics processing unit with application to trajectory optimization. Eng. Optim. **48**(10), 1679–1692 (2016)

[333] Qu, Jianhua., Liu, Xiyu., Sun, Minghe., Qi, Feng., "GPU-Based Parallel Particle Swarm Optimization Methods for Graph Drawing," Discrete Dynamics in Nature and Society, vol. 2017, Article ID 2013673, 15 pages, 2017.

[334] Wachowiak, M.P.; Timson, M.C.; DuVal, D.J.: Adaptive particle swarm optimization with heterogeneous multicore parallelism and GPU acceleration. IEEE Trans. Parallel Distrib. Syst. **28**(10), 2784–2793 (2017)

[335] Bo Van, Lin Yang,ShunpingZhou,RunWang,Dezhi Wang and Wenjie Zhen, "A Parallel-computing Approach for Vector Road-Network Matching using GPU archtiture" ISPRS Int.journal.Geo-Inf,7(12),472,2018

[336] Andrea Tangherloni,LeonardoRundo,SimoneSpolaor,PaoloCazzaniga,Marco s. Nobile,GPU-Powered Multi-swarm Parameter estimation of biological systems:A Master Slave approach,26th Euromicro international conference on parallel distributed and network based processing,2018.

[337] YunkioKawano,FevrierValdez,Oscar Castillo, Performance evaluation of optimization algorithms based on GPU using CUDA Architecture, IEEE Latin American conference on Computational I ntelligence,Mexio,2018.

[338] Vincent Roberge,Mohammad Tarbouchi,Aboelmagd,Noureldin, Integrated motor optimization and Route planning for EV using embedded GPU system,IEEE,2019.

[339] K. Amin, G. von Laszewski, M. Hategan, N. J. Zaluzec, S. Hampton, and A. Rossi. Gridant: A client-controllable grid work.ow system. In HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7, 2004.

[340] M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, University of California at Berkeley, February 2009.

[341] http://aws.amazon.com

[342] http://aws.amazon.com/s3/

[343] S. Pandey, L. Wu, S.M. Guru, R Buyya, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments". 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 400-407, 2010.

[344] Z. Wu, Z. Ni, L. Gu, "A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling". Computational Intelligence and Security (CIS), pp. 184-188, 2010.

[345] W. N. Chen and J. Zhang. 2012. A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 773–778.

[346] L. Z. Guo, S. G. P. Zhao, S. G. Shen, and C. Y. Jiang. 2012. A particle swarm optimization for data placement strategy in cloud computing. Information Engineering and Applications, Lecture Notes in Electrical Engineering, 323–330.

[347] S. Wang, Z. Liu, Z. Zheng, Q. Sun, F. Yang, Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers, in IEEE International Conference on Parallel and Distributed Systems (ICPADS) (2013), pp. 102–109.

[348] M. A. Rodriguez and R. Buyya. 2014. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing2, 2 (2014), 222–235.

[349] A.P. Xiong, C.-X. Xu, Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center. Math. Probl. Eng. (2014)

[350] F. Ramezani, J. Lu, F.K. Hussain, Task-based system load balancing in cloud computing using particle swarm optimization. Int. J. Parallel Progm. Int. J. Parallel Program. **42**(5), 739–754, (2014)

[351] H. H. Li, Y. W. Fu, Z. H. Zhan, and J. J. Li. 2015. Renumber strategy enhanced particle swarm optimization for cloud computing resource scheduling. In Proceedings of the IEEE Congress on Evolution *Computation* in press.

[352] Al-Olimat, Hussein S., et al. "Cloudlet scheduling with particle swarm optimization." Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on. IEEE, 2015.

[353] Li-Der Chou, Hui-Fen Chen, Fan-HsunTseng, Han-ChienChao, Yao-Jen Chang, DPRA :Dynamic power saving resource allocation for cloud data centre using PSO,IEEE Systems journal,Vol 12,No 2,2018.

[354]P.Shivkumar, S.Kalaiyarasi, S.Shilpa, D.Swetha,QPSO based Load balancing mechanism in Cloud environment,Proceeding of International conference on systems computation Automation and networking,2019.

[355] Zhang, Y.; Gallipoli, D.; Augarde, C.E.: Simulation based calibration of geotechnical parameters using parallel hybrid moving boundary particle swarm optimization. Comput. Geotech. **36**(4), 604–615 (2009)

[356] Jin, M.; Lu, H.: Parallel particle swarm optimization with genetic communication strategy and its implementation onGPU. In: IEEE 2nd InternationalConference on Cloud Computing and Intelligent Systems, vol. 1, pp. 99–104 (2012)

[357] Roberge, V.; Tarbouchi, M.; Gilles, L.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. Ind. Inform. **9**(1), 132–141, (2013)

[358] Chen, Y.Y.; Cheng, C.Y.; Wang, L.C.; Chen, T.L.: A hybrid approach based on the variable neighbourhood search and particle swarm optimization for parallel machine scheduling problems: a case study for solar cell industry. Int. J. Prod. Econ., 141(1), 66–78 (2013)

[359] Yuan, S.; Zhao, L.; Mu, B.: Parallel cooperative co-evolution based particle swarm optimization algorithm for solving conditional nonlinear optimal perturbation. In: International Conference on Neural Information Processing, pp. 87–95 (2015)

[360] Cao, B.; Li, W.; Zhao, J.; Yang, S.; Kang, X.; Ling, Y.; Lv, Z.: Spark-based parallel cooperative co-evolution particle swarm optimization algorithm. In: IEEE International Conference on Web Services, pp. 570–577 (2016)

[361] Long, H.X.; Li, M.Z.; Fu, H.Y.: Parallel quantum-behaved particle swarm optimization algorithm with neighbourhood search. In: International Conference on Oriental Thinking and Fuzzy Logic, pp. 479–489 (2016)

[362] Franz, W.; Thulasiraman, P.: A dynamic cooperative hybrid MPSO+GA on hybrid CPU+GPU fused multi core. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2016)

[363] Ge, X.; Wang, H.; Fan, Y.; Cao, Y.; Chen, H.; Huang, R.: Joint inversion of T1–T2 spectrum combining the iterative truncated singular value decomposition and the parallel particle swarm optimization algorithms. Comput. Phys. Commun. 198, 59–70 (2016)

[364] Peng, Y.; Peng, A.; Zhang, X.; Zhou, H.; Zhang, L.; Wang, W.; Zhang, Z.: Multi-core parallel particle swarm optimization for the operation of inter-basin water transfer-supply systems. Water Resour. Manag. **31**(1), 27–41 (2017)

[365] Ahmad M.Manasrah and Hanan Ba Ali,Work flow Schdueling using hybrid GA-PSO algorithm in Cloud computing, Wireless communication and Mobile computing, Hindawi, vol 2018,Article ID 1934784

[366] A.M.Senthilkumar,K.Parthiban,SivaSankar S,An efficient task scheduling in cloud computing environment using hybrid GA-PSO algorithm, International conference on Intelligent sustainable systems (ICISS 2019).

[367] Vlachogiannis, J.G.; Lee, K.Y.: Determining generator contributions to transmission system using parallel vector evaluated particle swarm optimization. IEEE Trans. Power Syst. **20**(4), 1765–1774 (2005)

[368] Fan, S.K.; Chang, J.M.: A parallel particle swarm optimization algorithm for multi-objective optimization problems. Eng. Optim**41**(7), 673–697 (2009)

[369] You Zhou and Ying Tan. GPU-Based Parallel Multi-objective Particle Swarm Optimization, *International Journal of Artificial Intelligence*, Vol. 7, No. A11, pp. 125--141, October 2011.

[370] Arun, J.P.; Mishra, M.; Subramaniam, S.V.: Parallel implementation of MOPSO on GPU using OpenCL and CUDA. In: 18th IEEE International Conference on High Performance Computing, pp. 1–10 (2011)

[371] Soares, J.; Vale, Z.; Canizes, B.; Morais, H.: Multi-objective parallel particle swarm optimization for day-ahead vehicle-to-grid scheduling. In: IEEE Symposium on Computational Intelligence Applications in Smart Grid, pp. 138–145 (2013)

[372] F. Khoshahval, A. Zolfaghari, H. Minuchehr, and M. R. Abbasi, “A new hybrid method for multi-objective fuel management optimization using parallel PSO-SA,” Progress in Nuclear Energy, vol. 76, pp. 112–121, 2014.

[373] Li, J-Z.; Chen,W-N.; Zhang, J.; Zhan, Z-H.:A parallel implementation of multi objective particle swarm optimization algorithm based on decomposition. In: IEEE Symposium Series on Computational Intelligence, pp. 1310–1317 (2015)

[374] Borges, N.; Soares, J.; Vale, Z.; Canizes, B.: Weighted sum approach using parallel particle swarm optimization to solve multi-objective energy scheduling. In: IEEE/PES Transmission and Distribution Conference and Exposition, pp. 1–5 (2016)

[375] ShouqAlsubaihi ,Jean-Luc Gaudiot,A Runtime Workload Distribution with Resourse Allocation for CPU-GPU HetrogeneousSystems,IEEE International Parallel and Distributed Processing Symposium workshops,2017.

[376] Arashpender, A.; Bernabe,D.; Gregoire, D.; Pascal, B.: Ascalable parallel cooperative coevolutionary PSO algorithm for multi-objective optimization. J. Parallel Distribution. Comput. **112**, 111–125 (2018)

[377] Hussian Md. Maruf and Fujimoto Noriyuki,”Parallel Multi-Objective PSO for large swarm and High dimension problems,IEEE,Congress on Evolutionary Computation (CEC),Brasil.pp 1-10,2018.

[378] Shewata Varshney, Sarvpalsingh, An Optimal Bi-objective PSO algorithm for task schdueling in cloud computing, Proceedings in 2nd international conference on Trends in Electronics and Informatics, 2018.

[379] Samuel Dustin Stanley,Khalid Kattan, Robert Reynolds, CAPSO:A Parallelized Multi-objective Cultural algorithm Particle swarm optimizer,IEEE,978-1-7281-2153—6/19

[380] Arion de camposjr, AuroraT. R.Pozo, Elias P. Duartejr, “Parallel multi-swarm PSO strategies for solving many objective optimization problems" Elsevier, vol 126,pp13-33,2019.

[381]Conradt, L., Krause, J., Couzin, I. D. & Roper, T. J. Leading according to need” in self-organizing groups. Am. Nat. 173, 304–312, 2009.

[382]Couzin, I. D, Krause, J, Franks, N. R. & Levin, S. A. Effective leadership an decision making in animal groups on the move. Nature 433, 513–516, 2005.

[383]Engelbrecht A.P, Fundamentals of Computational Swarm Intelligence: John Wiley & Sons,2006.

[384]Jain N.K., Nangia Uma, Jain Ashwary “PSO for Multi objective Economic Load Dispatch (MELD) for minimizing Generation Cost and Transmission Losses”, journal of the Institution of Engineers (India): series B,2015.

[385]Jain N.K., Nangia Uma, Jain Jyoti , “An improved PSO based on initial selection of particles, “in proceedings of the IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES) Electrical Engineering Department, Delhi Technological Univeristy,Delhi,pp.1-6,2016.

[386]Koyuncu,H and Ceylan R ,”A PSO based approach : Scout particle swarm optimization for continuous global optimization problems”, Journal 0f Computational Design and Engineering, Elsevier,2018.https://doi.org/10.1016/j.jcde.2018.08.003

[387]Nagy, M., A´ Kos, Z., Biro, D. & Vicsek, T. Hierarchical group dynamics in pigeon flocks. Nature 464, 890–893,2010.

[388]Solis, F. J., and Wets, R. J.-B. Minimization by random search techniques. Mathematics of Operations Research, 6(1):19–30,1981.

[389]Van den Bergh F, "An analysis of particle swarm optimizers," Ph.D Thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.

[390]Wang H,Liu Yong, Zeng Sanyou, Li Hui, Li Changhe “Opposition-based particle swarm algorithm with Cauchy mutation,” in proceedings of the IEEE Congress on Evolutionary Computation, pp. 4750-4756, 2007.

[391]Zhihua Cui, Chunmei Zhang, Yaru Zhao and Zhentao Shi , Adaptive Bat Algorithm Optimization Strategy for Observation Matrix, Applied Sciences, 9(15), 3008, 2019.

[392]Zoltan Toth, Beniamino Tuliozi, Davide Baldman, Herbert Hoi &Matteo Griggio “The effect of social connections on the discovery of multiple hidden food patches in a bird species” Scientific Reports 7,816,2017.

[393] .Jain N K, Uma Nangia, Uttam Kumar, “Load Flow studies based on a new particle swarm optimization,” in proceedings of the IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES) Electrical Engineering Department, Delhi Technological Univeristy,Delhi,2016,pp.1-5.

[394]Y. Shi and R. Eberhart, “Parameter selection in particle swarm optimization, ”Evolutionary Programming VII Lecture Notes in Computer Science, vol. 1447, pp.591-600 and 1998.

[395]J.Kennedy, and R. Eberhart.“ Swarm Intelligence”, Morgan Kaufmann Publishers, 2001.

[396] Hassoun MH, Fundamentals of Artificial Neural Networks.MIT Press, Cambridge, 1995.

[397] J. Salerno, Using the particle swarm optimization technique to train a recurrent neural model, in: Proc. of Ninth IEEE Int. Conf. on Tools with Artificial Intelligence pp.45–49, 1997.

[398]Hagan, M.T, Menhaj, M, "Training feed-forward networks with the Marquardt algorithm", IEEE Trans. Neural Networks, Vol. 5, No. 6, pp989-993, 1994.

[399]Foresee, F.D. & Hagan, M.T, "Gauss-Newton approximation to Bayesian regularization", International Joint Conference on Neural Networks, 1997.

[400]https://www.epa.gov/clean-air-act-overview/air-pollution-current-andfuture-challenges.

[401] Devinder kumar, N. k. Jain, Uma Nangia, "Perfectly convergent Particle swarm optimization in multi-dimensional space", International journal of Bio-inspired computation, Indersciene Publications, vol 18,no 4,p.221228,2021.

[402] Devinder kumar, N.k. Jain, Uma Nangia, " Combined Economic Emission Dispatch using Perfectly convergent Particle swarm optimization", DELCON, IEEE Delhi section Conference,Delhi,2022.

[403] Devinder kumar, N.k. Jain, Uma Nangia, " Multi area Economic Emission load dispatch using perfectly convergent Particle swarm optimization", DELCON, IEEE Delhi section Conference, Delhi, 2022.

[404] N.K. Jain, Uma Nangia , Devinder Kumar, "Machine learning through back propagation networks using OPSO with cauchy mutation and GCPSO in higher dimensions" in proceedings of the IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems(ICPEICES) Electrical Engineering Department, Delhi Technological Univeristy,Delhi,pp.1-6,2018.

[405] Venkatesh P, Gnanadass R and Padhy Narayana Prasad, "Comparison and application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints", IEEE Trans Power Syst, Vol. 18, Issue 2, pp. 688–697, 2003.

[406] Sunil Kumar Goyal, Neeraj Kanwar ,Jitendra Singh ,Manish Shrivastava ,Amit Saraswat, O. P. Mahela, Economic Load Dispatch with Emission and Line Constraints using Biogeography Based Optimization Technique", International Conference on Intelligent Engineering and Management (ICIEM),PP.470-76,2020.

[407] I. Jacob Raglend, Sowjanya Veeravalli, Kasanur Sailaja, B. Sudheera and D.P. Kothari, "Comparison of AI techniques to solve combined economic emission dispatch problem with line flow constraints", Electrical Power and Energy Systems, Vol. 32, pp. 592–598, 2010.

[408] M. Basu, ''Economic environmental dispatch using multi-objective differential evolution,'' Appl. Soft Comput., vol. 11, no. 2, pp. 2845–2853, Mar. 2011.

[409] D. C. Secui, A new modified artificial bee colony algorithm for the economic dispatch problem, Int. J. Energy Convers. Manag, Vol 89, pp. 43-62, 2015.

[410] A. Abdelaziz, E.Ali, and S.A. Elazim, ''Combined economic and emission dispatch solution using flower pollination algorithm,'' Int. J. Elect. Power Energy Syst., vol. 80, pp. 264–274, Sep. 2016.

[411] Ruyi Dong, Shengsheng Wang," New Optimization algorithm inspired by Kernel tricks for the Economic emission dispatch problem with valve point," Special section on AI technologies for electric power system, 2020. doi 10.1109/ACCESS.2020.2965725.

[412] P.K. Roy and S. Bhui, ''Multi-objective quasi oppositional teaching learning based optimization for economic emission load dispatch problem,'' Int. J. Elect. Power Energy Syst., vol. 53, pp. 937–948, Dec. 2013.

[413] L.D.S. Coelho, ''Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems,'' Expert Syst. Appl., vol. 37, no. 2, pp. 1676–1683, Mar. 2010.

[414] M. Taherkhani and R. Safabakhsh, ''A novel stability-based adaptive inertia weight for particle swarm optimization, '' Appl. Soft Comput. , vol. 38, pp. 281–295, Jan. 2016.

[415] S. Jiang, Z. Ji, and Y. Shen, ''A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with

various practical constraints,'' Int.J.Electr.Power Energy Syst., vol. 55, pp. 628–644, Feb. 2014.

[416]T. Niknam and H. Doagou-Mojarrad, ''Multi objective economic/emission dispatch by multi objective thetas-particle swarm optimisation,'' IET Gener., Transmiss. Distrib., vol. 6, no. 5, pp. 363–377, May 2012.

[417] D. Aydin, S. Özyön, C. Yaşar, and T. Liao, ''Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem, ''Int. J. Electr. Power Energy Syst.,vol.54,pp.144–153, Jan. 2014.

[418] M. A. Ajzerman, E. M. Braverman, and L. I. Rozonoehr, ''Theoretical foundations of the potential function method in pattern recognition learning,'' Automat. Remote control, vol. 25, no. 6, pp. 821–837, Jan. 1964.

[419] B. Qu, J. Liang, Y. Zhu, Z. Wang, and P. Suganthan, ''Economic emission dispatch problems with stochastic wind power using summation based multi-objective evolutionary algorithm,'' Inf. Sci., vol. 351, pp. 48–66, Jul. 2016.

[420] E. Afzalan and M. Joorabian, ''Emission, reserve and economic load dispatch problem with non-smooth and non-convex cost functions using epsilon-multi-objective genetic algorithm variable,'' Int. J. Electr. Power Energy Syst., vol. 52, pp. 55–67, Nov. 2013.

[421] P. Hota, A. Barisal and R. Chakrabarti, ''Economic emission load dispatch through fuzzy based bacterial foraging algorithm,'' Int. J. Elect. Power Energy Syst., vol. 32, no. 7, pp. 794–803, Sep. 2010.

[422] S. Sayah, A. Hamouda, and A. Bekrar, ''Efficient hybrid optimization approach for emission constrained economic dispatch with non smooth cost curves, ''Int. J. Electr. Power Energy Syst., vol.56,pp.127–139,Mar.2014.

[423] Sharmila Deve Venkatachalam, Keerthivasan Krishnamoorthy, Ahmed Said Ahmed Al-Shahri, Balachander Kalappan, Modified Teaching Learning Based Optimization Technique to Achieve the Best Compromise Solution of Economic Emission Dispatch Problem", Journal of Green Engineering (JGE) ,Vol-10, Issue-5,pp. 2458–2482, 2020.

[424] Kuntal Bhattacharjee , Aniruddha Bhattacharya , Sunita Halder nee Dey , " Solution of Economic Emission Load Dispatch problems of power systems by Real Coded Chemical Reaction algorithm", Electrical Power and Energy Systems, Vol. 59 , pp.176–187, 2014.

[425] Rajasomashekar S, Aravindhababu S, "Biogeography based optimisation technique for best compromise solution of economic emission dispatch", Swarm and Evolutionary Computation, Vol. 7, pp.47-57, 2012.

[426] Mostafa Modiri-Delshad, Nasrudin Abd Rahim, Multi-objective backtracking search algorithm for economic emission dispatch problem", Applied Soft Computing, Vol 40,pp. 479-494,2016.

[427] Güvenc, Ugur,Y.U.S.U F. Sönmez, Serhat Duman, and Nuran Yörükeren. "Combined economic and emission dispatch solution using gravitational search algorithm." *Scientia Iranica* 19, no. 6,pp. 1754-1762.2012.

[428] Dexuan Zou, Steven Li, Zongyan Li, Xiangyong Kong, Ä new global particle swarm optimization for the economic emission dispatch with or without transmission losses", Energy conversion and management, vol 139,pp.49-70,2017.

[428]S. Krishnamurthy, R. Tzoneva, Impact of Price Penalty Factors on the Solution of the Combined Economic Emission Dispatch Problem using Cubic Criterion Functions. IEEE Power and Energy Society General Meeting, pp. 1-9, 2012.

[429] Pitono J, Soepriyanto A, Purnomo MH Advance optimization of economic emission dispatch by particle swarm optimization (PSO) using cubic criterion functions and various price penalty factors. Kursor Sci J Inf Technol 7(3):153–164, 2014.

[430]Ziane Ismail, Farid Benhamida, and Graa Amel."Simulated annealing algorithm for combined economic and emission power dispatch using max/max price penalty factor." Neural Computing and Applications 28.1 197-205, 2017.

[431]Ziane Ismail, Benhamida Farid and Graa Amel, “Combined economic emission dispatch with cubic criterion functions and various price penalty factors to reduce SO2, NOX and CO2 gas emissions,” Advances in energy environmental science and engineering, pp.159-165, 2015.

[432]Ramesh Ramachandran and P. Avirajamanjula, “Modified biogeography based optimization algorithm for power dispatch using max/max price penalty factor”, International Journal of Pure and Applied Mathematics, Vol. 118 No. 18, pp. 3813-3818, 2018.

[433] Karthikeyan. R, Subramanian. S, Elanchezhian. E B, Combined Economic and multiple emissions optimization considering third order polynomials using Grasshopper optimization algorithm” International journal of Innovative technology and Exploring Engineering, vol.8, issue-8 June,2019.

[434] W. Zhao, L. Wang, and Z. Zhang, ``Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm," Neural Computing. Appl., vol. 32, no. 13, pp. 9383_9425, Jul. 2020, doi: 10.1007/s00521-019-04452-x

[435] Mohamed H. Hassan, Salah Kamel, Sinan Q Salih, Tahir Khurshaid, Mohamed Ebeed, ”Developing chaotic Artificial Ecosystem-based optimization algorithm for combined economic emission dispatch, IEEE Access vol 9,2021 doi:10.1109/ACCESS.2021.3066914

[436]Mahadi F Parvez, VasantP, Abdullah-Al-WadudM, WatadaJ, Kallimani V,”Ä Quantum-inspired particle swarm optimization approach for environmental/economic power dispatch problem using cubic criterion function” Int Trans Electr Energ Syst.2017,doi:10.1002/etep.2497

[437]D. Gonidakis and A. Vlachos, ``A new sine cosine algorithm for economic and emission dispatch problems with price penalty factors," J. Inf. Optim. Sci., vol. 40, no. 3, pp. 679-697, Apr. 2019, doi:10.1080/02522667.2018.1453667.

[438]D. Mishra, A. Shukla, and P.K. Kalra, “OR-Neuron Based Hopfield Neural Network for Solving Economic Load Dispatch Problem”, Neural Information Processing – Letters and Reviews Vol. 10, No. 11, pp. 249-259, November 2006

[439]T. Adhinarayanan , and M. Sydulu, “Particle Swarm Optimisation for Economic Dispatch with Cubic Fuel Cost Function”, TENCON 2006, IEEE Region 10 Conference, 10th April 2007, pp 1-4.

[440] T. Adhinarayanan , and M. Sydulu, “Fast and Effective Algorithm for Economic Dispatch of Cubic Fuel Cost Based Thermal Units”, First International Conference on Industrial and Information Systems, 29th May 2007.

[441] J. S. Al-Sumait , J. K. Sykulski ,and A. K. Al-Othman, “Solution of Different Types of Economic Load Dispatch Problems Using a Pattern Search Method”, Electric Power Components and Systems, Vol. 36, pp. 250– 265,2006.

[442] Balamurugan, R., and Subramanian, S. A Simplified Re-cursive Approach to Combined Economic Emission Dispatch. Electric Power Components and Systems, Vol. 36, No. 1, pp. 17-27, 2007.

[443] Z.Ming, L. Xiaotong, Y.Fan, and T. Kuo, “The multi-objective optimization model of energy-efficient scheduling based on PSO algorithm” Power and Energy Engineering Conference (APPEEC), 28th – 31st March 2010 , Chendgu, china.

[444] Krishnamurthy, S., and Tzoneva, R,” Comparative Analyses of Min-Max and Max-Max Price Penalty Factor Approaches for Multi Criteria Power System Dispatch Problem Using Lagrange‘s Method”, International conference on recent advancements in Electrical, Electronics and Control Engineering, Sivakasi, India, pp 36-43,2011.

[445] Park JB, Jeong YW, Sin JR, Lee KY,” An Improved particle swarm optimization for non-convex Economic load dispatch problems”, IEEE Trans Power Syst 25(1):156–166,2010.

[446] R R Shoults, S K Chang, S Helmick , WM Grady. “A practical approach to unit commitment, economic dispatch and savings allocation for multiple area pool operation with import/export constraints”, IEEE Trans Power Appar Syst; 99(2):625e35, 1980

[447] R Romano , VH Quintana , R Lopez , V Valadez, “Constrained economic dispatch of multi-area systems using the Dantzige Wolfe decomposition principle”. IEEE Trans Power Appar Syst; 100(4):2127e37, 1981.

[448] S D Helmick , R R Shoults,”A practical approach to an interim multi-area economic dispatch using limited computer resources”, IEEE Trans Power Appar Syst; 104(6):1400e4, 1985

[449] C Wang,S M Shahidehpour ,”A decomposition approach to non-linear multi area generation scheduling with tie-line constraints using expert systems”. IEEE Trans Power System, 7(4):1409e18, 1992.

[450] D Streiffer,” Multi-area economic dispatch with tie line constraints”, IEEE Trans Power Syst; 10(4):1946e51, 1995.

[451] T Yalcinoz , “Short MJ. Neural networks approach for solving economic dispatch problem with transmission Capacity constraints”, IEEE Trans Power System, 13(2):307e13, 1998.

[452] T Jayabarathi , G Sadasivam , V Ramachandran, “Evolutionary programming based multi-area economic dispatch with tie line constraints”, Electr Mach Power Syst 28:1165e76,2000.

[453] M Sharma, M Pandit , L Srivastava , ‘Reserve constrained Multi-area economic dispatch employing differential Evolution with time-varying mutation”, Int J Electr Power Energy Syst, 33(3): 753-66,2011.

[454] P S Manoharan , P S Kannan , S Baskar , Iruthayarajan M Willjuice, ”Evolutionary algorithm solution and KKT based optimality verification to multi-area economic dispatch”, Electr Power Energy Syst , 31:365–73,2009.

[455]F N Lee, A M Breipohl," Reserve constrained economic load dispatch with prohibited operating zones" IEEE Trans Power Syst , 8(1):246– 54,1993.

[456] T Aruldoss , Albert Victoire, A Ebenezer Jeyakumar, "Reserve constrained dynamic dispatch of units with valve point effects", IEEE Trans Power Syst,20(3):1273–82,2005.

[457]Lingfeng Wang, Chanan Singh," Reserve constrained multi-area environmental/ economic dispatch based on particle swarm optimization with local search", Eng Appl Artif Intel, 22:298–307, 2009.

[458] E Nasr Azadani, S H Hosseinian, B Moradzadeh, "Generation and reserve dispatch in a competitive market using constrained particle swarm optimization", Electr Power Energy Syst, 32:79–86,2010.

[459] John G Vlachogiannis, Y Lee Kwang ,"Economic load dispatch – a comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO", IEEE Trans Power Syst, 24(2):991– 1001, 2009.

[460] N Sinha, R Chakrabarti, P K Chattopadhyay, "Evolutionary programming techniques for economic load dispatch", IEEE Transactions on Evolutionary Computation, 7(1):83e94, Feb 2003.

[461] T.F. Robert, A.H. King, C.S. Harry, Rughooputh, K. Deb, "Evolutionary multi-objective environmental/economic dispatch: stochastic versus deterministic approaches", KanGAL, Rep. 2004019, pp. 1–15, 2003.

[462] Ghasemia Mojtaba, , Aghaeia Jamshid, Akbarib Ebrahim, Ghavidelc Sahand, Li Li , "A differential evolution particle swarm optimizer for various types of multi-area economic dispatch problems" ,Energy,vol107, Pages 182-195,2016,doi.org/10.1016/j.energy.2016.04.

[463] Md. Asgar Ali, Hari Mohan Dubey, Manjaree Pandit, "Moth-Flame Optimization for Multi Area Economic Dispatch: A Novel Heuristic Paradigm", International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS-2017.

[464] V. P. Sakthivel , H. H. Goh , S. Srikrishna, P. D. Sathya , and S. K. A. Rahim, "Multi-Objective Squirrel Search Algorithm for Multi-Area Economic Environmental Dispatch with Multiple Fuels and Valve Point Effects", IEEE Access, 2021,doi:10.1109/ACCESS.2020.3046257.

[465] Namarta Chopra, Y.S. Brar and J. S. Dhillon, Multi area Economic Emission Dispatch Using Simplex Based Particle Swarm Optimization Method" Asian Journal of Electrical Sciences ISSN: 2249 – 6297, Vol. 7, No. 1, pp. 29-36, 2018.

[466] S. Krishnamurthy and R. Tzoneva, "Decomposition-Coordinating Method for Parallel Solution of a Multi Area Combined Economic Emission Dispatch Problem" International

Journal of Electrical and Computer Engineering (IJECE), Vol. 6, No. 5, pp. 2048-2063, 2016.

[467] Chen, Y.M., and WS Wang, “Economic Dispatch with Environmental Considerations Using Marginal Rate of Substitution Decision Approach. Journal of Quality Vol. 16, No. 2, pp.109-118, 2009.