# MULTIOBJECTIVE ECONOMIC LOAD DISPATCH USING INTELLIGENT TECHNIQUES

A Thesis Submitted to

Delhi Technological University

For the Award of

## DOCTOR OF PHILOSOPHY
## in
## Electrical Engineering

Submitted by

**JYOTI JAIN**

(Enrollment No. 2K11/PhD/EE/01)

Under the Supervision of

**Prof. N. K. Jain**
**EED, DTU**

**Prof. Uma Nangia**
**Head EED, DTU**

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**DELHI TECHNOLOGICAL UNIVERSITY**
**Bawana Road, Delhi-110042, INDIA**
**OCTOBER, 2019**

# CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this thesis entitled **"Multiobjective Economic Load Dispatch using Intelligent Techniques"** submitted in partial fulfilment of the requirement for the award of the degree of Doctor of Philosophy in the Department of Electrical Engineering, Delhi Technological University, Delhi is an authentic record of my own work carried out under the supervision of Prof. N.K.Jain and Prof. Uma Nangia.The matter presented in this thesis has not been submitted elsewhere for the award of a degree.

Place: Delhi **Jyoti Jain**

Date:

# DEPARTMENT OF ELECTRICAL ENGINEERING

Delhi Technological University

## CERTIFICATE

This is to certifiy that the thesis titled "**Multiobjective Economic Load Dispatch using Intelligent Techniques**" submitted by Ms. Jyoti Jain, a student of Doctor of Philosophy in the Department of Electrical Engineering at Delhi Technological University, is a work carried out by her under our supervision towards the partial fulfillment of the requirement of award of degree of Doctor of Philosophy and is an original contribution with existing knoweldege and faithful record of research work carried out by her under our guidance and supervision.

To the best of our knowledge this work has not been submitted in part or full for any degree or diploma to this university or elsewhere.

**Prof. N.K. Jain**

Electrical Engineering department
Delhi Technological University
Bawana road, Delhi-110042, India

**Prof. Uma Nangia**

Electrical Engineering Department
Delhi Technological University
Bawana road, Delhi-110042, India

# ACKNOWLEDGEMENT

First of all, I am extremely grateful to my Supervisors, Prof. N.K. Jain and Prof. Uma Nangia, for priceless guidance, scholarly inputs, constant and unconditional support I received throughout the research work. This feat was possible only because of the unconditional support provided by my supervisors. Persons with an amicable and positive disposition, sir and madam always made them accessible to me to clarify my doubts despite their hectic schedules. I consider it as a big opportunity to do my Ph.D. under their guidance and to learn from their research erudition and knowledge. I thank them again for their assistance and support.

I owe my gratitude to all the faculty members of Electrical Engineering, Department, Delhi Technological University (DTU), Delhi, for their constant support throughout the duration of this work. I am especially thankful to Head of Electrical Engineering Department, DTU, Delhi who gave me moral support and provided all infrastructural facilities required for successful completion of this work.

I would also like to thank Prof. Narender Kumar, Prof. Pragati Kumar, Prof Minishreejith, Prof. Rachna Garg, Prof. Dheeraj Joshi, Prof. Bharat Bhushan, Prof. Alka Singh, Mr. Manoj and other members of the Department of Electrical Engineering, DTU, Delhi, who have inspired and motivated me to carry out this ambitious work to its logical end.

I thank the Chairperson, Management, Director and Prof. Meena Tushir (HOD EEE) of Maharaja Surajmal Institute of Tecnology (MSIT) for deputing me to pursue Ph.D. degree. I am also thankfull to Prof. Meena Tushir and faculties of Electrical and Electronics Engineering Department of MSIT for their valueable support and motivating me to complete my work at time.

The completion of this work came at the expense of my long hours of absence from home. No words are adequate to express my indebtedness to my family members Shri Sunil Kumar Jain (Husband), my son Shrey Jain, my Mother and Mother in law, Father, brothers and sisters for all the pains and suffering they have undergone to bring me up to this stage. I am extremely thankful to all my relatives too for their sincere good wishes. I am also thankful to all who enlightened me during my research. Completion of this Ph.D. thesis was achieved with the support of several people. I would like to express my sincere gratitude to all of them.

Place: New Delhi Jyoti Jain

Date

# ABSTRACT

Power sector is one of the important infrastructures of a country, which is the prime mover of the overall economic development. The all India demand for electricity is increasing continuously across all segments of the economy such as agriculture, industry, commercial sector, domestic sector etc. Thermal power is the prime resource to meet the demand. Coal power generation increased 3% in 2018 (similar to the 2017 increase), and for the first time crossed the 10000 TWh. mark. Coal remains firmly in place as the largest source of power of overall generation. To meet the demand of all sectors, generation of electrical power is essential. Economic Load Dispatch problem is an optimization problem which minimizes the total fuel cost of all committed plants while meeting the demand and losses.

Real life problems may be nonlinear, non-differentiable and discontinuous. These cannot be solved using classical optimization techniques. Classical techniques have the tendency of settling down at local minima instead of the global best solutions. Therefore, intelligent techniques are being used to solve real life problems. But their computational efficiency is very slow and suffer from poor convergence. To overcome the limitation of Intelligent techniques, some improvements /modifications need to be carried out.

The optimal power system operation is achieved when various objectives of power systems: cost of generation, system transmission losses, environmental emission etc. simultaneously achieve their minimum value. But these objectives may be conflicting in nature and cannot be handled by conventional single objective optimization techniques. Single objective optimization techniques give the best value of objective under consideration whereas the values of other objectives may not be acceptable at all. Therefore, Multiobjective approach has been used to solve such problems.

In this research work, economic load dispatch (ELD) and multiobjective economic load dispatch (MELD) problem have been solved using intelligent techniques i.e. genetic algorithm (GA) and basic particle swarm optimization (BPSO). Using improvement and many modifications in basic particle swarm optimization (BPSO) new improved / modified algorithms i.e. initial selection based particle swarm optimization (IPSO IS), adaptive social acceleration constant based particle swarm optimization (ASACPSO) and feasibility oriented particle swarm optimization (FOPSO) have been developed, which have resulted in significant reduction in computational effort. Also, the Pareto – Front for MELD problem has been achieved in a single run (rather in a partial run) using FOPSO for IEEE 5, 14 and 30 bus systems considering cost of generation, system transmission losses and environmental emission. MELD problem for IEEE 5, 14 and 30 bus systems, considering cost of generation, system transmission losses and environmental emission is formulated using weighting method and Noninferior set has been generated by basic particle swarm optimization (BPSO). MELD problem for IEEE 5, 14 and 30 bus systems, considering cost of generation, system transmission losses is also formulated using constraint method and Noninferior set has been generated by genetic algorithm. In this research work a sincere effort has been made to improve the computational efficiency of intelligent techniques in general and to solve ELD and MELD problem in particular. Many improvements and modifications have been carried out in BPSO, which have resulted in significant reduction of computational effort.

# LIST OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| F | Multiobjective function to be optimized |
| $F_C$ | Cost of Generation in $/hr. |
| $F_L$ | Transmission Losses in MW |
| $F_E$ | Environmental Emission in Kg/hr. |
| f | Function taking care of Equality Constraints |
| $\boldsymbol{g_i}$ (x) | Inequality constraint |
| h | Number of objectives |
| I.P. | Ideal Point |
| k | Current iteration |
| ITmax | Maximum number of iterations |
| $a_i$, $b_i$, $c_i$ | Cost coefficients of $i^{th}$ generator |
| $d_i$, $e_i$, $f_i$ | Environmental emission coefficients of $i^{th}$ generator |
| $B_{ij}$ | Loss coefficients |
| $C_p, C_g$ | Acceleration coefficients |
| $F_{Cmax}$ | Maximum value of cost of generation |
| $F_{Cmin}$ | Minimum value of cost of generation |
| $F_{Lmax}$ | Maximum value of transmission loss |
| $F_{Lmin}$ | Minimum value of transmission loss |
| $F_{Emax}$ | Maximum value of environmental emission |
| $F_{Emin}$ | Minimum value of environmental emission |

| | |
|---|---|
| $\tau_i$ | Minimum relative attainment of $i^{th}$ objective |
| $\tau_C$ | Minimum relative attainment of generating cost |
| $\tau_L$ | Minimum relative attainment of transmission losses |
| $\tau_E$ | Minimum relative attainment of emission |
| NG | Total number of generators in the system |
| N | Number of Particles in Qualifying Set |
| $P_i$ | Active power generation of the $i^{th}$ generator |
| $P_{gimax}$ | Maximum power generation limit of $i^{th}$ generator |
| $P_{gimin}$ | Minimum power generation limit of $i^{th}$ generator |
| $P_D$ | Total power demand |
| P | Number of particles in the swarm |
| P2 | Remaining number of particles participating in PSO after removing the which satisfy the constraints, particles to the Repository set or Qualifying set. |
| PP | Intermediate variable |
| $r_p, r_g,$ | Random numbers between 0 and 1 |
| T.P. | Target Point |
| $V_{ij}^k$ | Velocity of $j^{th}$ particle of $i^{th}$ generator at $k^{th}$ iteration |
| W | Inertia weight factor |
| x | Variable |
| L | Lower bound limit |
| U | Upper bound limit |
| $x^L$ | Lower limit of x variable |
| $x^U$ | Upper limit of x variable |

| | |
|---|---|
| $W_1$ | Weight attached to Generation of cost |
| $W_2$ | Weight attached to System transmission losses |
| $W_3$ | Weight attached to Emission |
| Kount | Function Evaluation |
| $X_{ij}^{k}$ | Current position of $j^{th}$ particle of $i^{th}$ generator at $k^{th}$ iteration |
| $X_{pbestij}^{k}$ | Personal best position of $j^{th}$ particle of $i^{th}$ generator at $k^{th}$ iteration |
| $X_{gbesti}^{k}$ | Global best position of swarm for $i^{th}$ generator till $k^{th}$ iteration. |
| Z | Multi-objective function |
| $Z_{iatZ(i+k)}$ | Worst feasible value of $i^{th}$ objective |

# CHAPTER 1
# INTRODUCTION

## 1.1 OVERVIEW

Development of infrastructure is always a priority of any country for overall development of the country. Power sector is one of the important infrastructures of a country, which is the prime mover of overall economic development. With all round development and efforts of the government to supply electricity to each and every corner of the country the all India demand for electricity is also picking up continuously across all segments of the economy such as agriculture, industry, commercial sector, domestic sector etc. To meet the demand of all sectors, generation of electrical power is essential. Thermal power is the prime resource to meet the demand.

**Cost of generation:** Minimizing cost of generation of power is one of the most important objective because if cost of generation increases the power will be supplied to ultimate consumer at higher price. Cost of power is significant input cost of any product and if cost of power increases, the cost of every product or service gets increased. As an objective of multiobjective economic load dispatch (MELD) problem besides achieving other objectives the cost of generation of power is sought to be minimized. In other words, power generation and transmission has to be in such manner as to minimize cost.

**System transmission losses:** Power saved is power generated and power losses in transmission ultimately pushes up the cost of power transmitted to ultimate consumer besides waste of this significant resource. Transmission losses form 5-10% of total generation. As an objective of multiobjective economic load dispatch, (MELD) problem besides achieving other objectives the transmission losses are sought to be minimized. Because reduction of system transmission losses will result in improvement of voltage profile and this in turn would result in reduction of cost of generation. In other words, power generation and transmission has to be in such manner to minimize system transmission losses. India has one of the highest levels of electricity transmission and distribution (T&D) losses in the world. T&D losses represent electricity that is generated but does not reach intended customers. India's T&D losses are almost 20% of generation, more than twice the world average and nearly three times as large as T&D losses in the United States.

**Environmental emission:** The significance of control of pollution and environment emission is very important, especially in the case of thermal power generation. In the case of thermal power generation coal is the main fuel, which results in emission of pollutant gases and substances such as Sulphur oxides, Nitrogen oxides and Carbon dioxide into the atmosphere which is harmful for human as well as other life –forms. As an objective of multiobjective economic load dispatch, (MELD) problem besides achieving other objectives, the environmental emission is sought to be minimized. In other words, power generation and transmission has to be in such manner to minimize environmental emission.

In this research work, three important objectives of power systems – cost of generation, system transmission losses and environmental emission have been considered for multiobjective economic load dispatch (MELD) of IEEE 5, 14 and 30 bus systems.

Single objective optimization techniques give optimal solution of a problem with respect to only one aspect i.e. they give the best value of the objective function under consideration. However, the value of other objectives in such a solution may be such that these cannot be accepted. Hence, the results are of no use for utilities to facilitate decision-making. This situation paves the way for multiobjective approach to problem solving because in practical situation there are a number of objectives to be optimized, some of them may be conflicting.

There are two limbs of multiobjective decision-making process: analysis and decision-making [3]. *Analysis* of a problem gives information and details of the problem for making decisions. Multiobjective approaches pursue a different *decision - making* process. It needs clear consideration of the relative effects of the different objectives on the problem. These approaches to decision making highlight and emphasize the range of choices associated with a decision-making problem. The responsibility of assigning relative values to various objectives remains with the decision maker (power system operator). The significance of multiobjective approach can be understood from the fact that these provide sufficient information to facilitate decision-making process.

Multiobjective techniques are used to generate and evaluate more than one alternative. These techniques indicate to decision makers a range of choices beyond one optimal alternative identified by single objective techniques. A general rule for decision making which is assumed is that more information carefully presented is better than less information. The decision to accept or reject a single optimal alternative is an uninformed decision. Informed decision-making requires a knowledge of full range of possibilities provided by multiobjective analysis. Multiobjective analysis allows several noncommensurable effects to be treated without artificially combining them.

In this research work, Intelligent Techniques - Genetic Algorithm (GA) [2,14] and Basic

Particle Swarm Optimization (PSO) [10,11] have been used to solve economic load dispatch (ELD) and multiobjective economic load dispatch (MELD) problems. Many modifications of basic particle swarm optimization (BPSO) i.e. initial selection based particle swarm optimization (IPSO IS), adaptive social acceleration constant based particle swarm optimization (ASACPSO) and feasibility oriented particle swarm optimization (FOPSO) have been suggested, which have resulted in significant reduction in computational effort. Also, the Pareto – Front for MELD problem has been achieved in a single run (rather in partial run) for IEEE 5, 14 and 30 bus systems considering cost of generation, system transmission losses and environmental emission.

In this research work a sincere effort has been made to improve the computational efficiency of intelligent techniques in general and to solve ELD and MELD problem in particular. Many improvements and modifications have been carried out in BPSO, which have resulted in significant reduction of computational effort. There are various ways of measuring the time consumed in running a computer programme. First, the easiest and obvious method is to consider the time given by the computer. This method includes the time taken for input and output statements along with the time taken for computation. For this reason, it is not a correct measure of computational effort. Further, the time given by the computer for execution of the programme depends on technology. We know technology is changing fast.  Computers are becoming faster and faster day by day. So, the computational time given by a computer for running a particular programme at a particular instant of time cannot be compared with time taken by another computer after some time. Second, in case of iterative techniques, the number of iterations have been taken as measure of computational effort by many researchers. This also cannot reflect true computational time because the time taken for each iteration is not same. In the techniques study in this research work involves calculation of the value of objective function to be optimized. It is done several times before the optimal solution is obtained. This has been considered as the measure of computational effort. It is represented by "Kount". This represents the number of time a function has been evaluated during the optimization process. So new parameter "**Kount"** has been designed to measure the computational effort.

## 1.2 THE GOALS OF THE PROPOSED RESEARCH WORK

Following are specific goals of the research work presented in this thesis:

1. To improve the performance of existing meta-heuristic algorithms for the economic load dispatch problem considering single as well as multi-objective criterion.
2. To develop improved algorithms for maximum exploitation and exploration of the solution search space to ensure global solution.
3. Validation of modified / improved algorithms using several standard mathematical benchmark functions.
4. Implementation of modified / improved algorithms for ELD of IEEE 5 bus,14 bus and 30 bus systems.
5. Identification of objectives of Multiobjective Economic Load Dispatch problem.
6. Formulation of Multiobjective Economic Load Dispatch (MELD) Problem and creation / production / identification of Pareto-Optimal Front / Noninferior set.
7. Implementation of Intelligent Techniques, improved and modified Intelligent Techniques to Multiobjective Economic Load Dispatch (MELD) of IEEE 5bus,14 bus and 30 bus systems.

8. Achievement of Target Point / The Best Compromise Solution.

## 1.3 METHDOLOGY

The proposed research work has been carried out in these lines

**(1) Identification of objectives**

There are various important objectives of power systems: cost of generation, system transmission losses, environmental emissions, voltage stability, reliability etc. some of these may be conflicting in nature. In this research work, three important objectives of power systems have been considered-cost of generation ($F_C$), system transmission losses ($F_L$) and environmental emissions ($F_E$).

**(2) Formulation of Multiobjective Economic Load Dispatch (MELD) Problem and creation / production / identification of Pareto-Optimal Front / Noninferior set**

Generating techniques emphasize the development of information about a multiobjective problem that is presented to a decision maker that allows the range of choices and tradeoffs among objective to be well understood. In the present research work, weighting method, ε constraint method and Minimum Distance method have been used to formulate MELD problem and for creation/production/identification of Noninferior set /Pareto Optimal Front.

**(3) Development of Improved and Modified Intelligent Techniques**

Intelligent techniques are inherently computationally slower than conventional techniques; however, these techniques have other advantages. The conventional optimization techniques are unable to globally optimize a system effectively. They are generally single path search algorithms, starting from an initial condition and improving the control variables in every iteration. These techniques are trapped in a local optimum. Intelligent techniques have the capacity to deal with highly nonlinear optimization problem.

In this research work, a sincere effort has been made to improve the computational efficiency of intelligent techniques in general and for ELD and MELD problem in particular. Many improvements and modifications have been carried out in Basic Particle Swarm Optimization (BPSO), which have resulted in significant reduction of computational effort.

As already mentioned, new parameter Kount has been designed to measure the computational effort. Improved PSO is based on initial selection of particles and is named as (IPSO IS) has been developed to reduce computational effort. Some modifications have been carried out in BPSO leading to the development of two modified algorithms- Adaptive Social Acceleration Constant based Algorithm (ASACPSO) and Split Phase Economic Load Dispatch Algorithm (SPELDA).

A new algorithm Feasibility Oriented Minimum Distance Based Particle Swarm Optimization (FOPSO) has also been developed for MELD problem. This algorithm enables us to achieve Pareto Optimal Front in less than single run i.e. in a partial run for IEEE 5,14 and 30 bus systems.

**(4) Implementation of Intelligent Techniques to Economic Load Dispatch (ELD)**

Intelligent techniques have been implemented to solve ELD problem. The improved and modified intelligent techniques: ISPSO IS, ASACPSO, SPELDA have been successfully implemented on ELD problem of IEEE 5, 14 and 30 bus systems.

**(5) Implementation of Intelligent Techniques to Multiobjective Economic Load Dispatch (MELD) problem**

MELD problem has been formulated using weighting method and Ɛ - constraint method. Pareto-Optimal Front has been obtained by Basic Particle Swarm Optimization (BPSO) and Genetic Algorithm (GA). In this case, the MELD problem has to be executed many times to generate the Pareto-Optimal Front / Noninferior set. A modified algorithm FOPSO has been designed to obtain the Pareto – Optimal Front in less than a single run. Three important objectives of power systems – cost of generation, system transmission losses and environmental emission have been considered for multiobjective economic load dispatch (MELD) of IEEE 5, 14 and 30 bus systems.

**(6) Achievements of Target Point / The Best Compromise Solution**

The Target point / the best compromise solution has been identified from the Pareto - Optimal Front / Noninferior set using different methods: Maximization of minimum relative attainment, Fuzzy Logic system and Minimum distance method.

## 1.4 ORGANIZATION OF THESIS

The thesis has been organized in eight chapters.

**Chapter-1:** This chapter presents the overview, research goals, methodology and organization of thesis work.

**Chapter-2:** This chapter describes a literature survey on intelligent techniques and their applications in Economic Load dispatch and Multiobjective Economic Load Dispatch problem.

**Chapter-3:** This chapter deals with the Genetic Algorithms and its implementation to benchmark function. The effect of bit size on convergence of the function is presented. The application of Basic Particle Swarm Optimization (BPSO) to benchmark functions has also been studied.

**Chapter-4:** This chapter addresses the design and development of two improved / modified Basic Particle Swarm Optimization (BPSO) algorithms. These are Improved PSO (IPSO IS) based on initial selection of particles and Adaptive Social Acceleration Constant based PSO (ASACPSO) algorithm. These two algorithms have been implemented on benchmark functions and ELD problem for IEEE 5, 14 and 30 bus systems. In addition, the comparison of results of these improved algorithms with BPSO for all the systems is presented.

**Chapter-5:** Chapter 5 presents Split Phase Economic Load Dispatch algorithm (SPELDA) and its implementation to Economic Load dispatch of IEEE 5, 14 and 30 bus systems. The results of SPELDA have been compared with BPSO and Lambda iteration method.

**Chapter-6:** This chapter introduces the formulation of Multiobjective Economic Load Dispatch (MELD) problem using weighting method considering three objectives – cost of generation, system transmission losses and environmental emission. The Pareto-Optimal Front has been generated using BPSO for IEEE 5, 14 and 30 bus systems.

The Multiobjective Economic Load Dispatch (MELD) problem has also been formulated using constraint method considering two objectives - cost of generation and system transmission losses. The Pareto - optimal Front has been generated for IEEE 5, 14 and 30 bus systems using GA. Target point of MELD problem has been identified using Maximization of Minimum Relative Attainment and Fuzzy Logic System.

**Chapter-7:** In this chapter, a new proposed algorithm called as Feasibility Oriented Minimum Distance Based Particle Swarm Optimization (FOPSO) to solve Multiobjective Economic Load Dispatch problem considering two objectives and three objectives of power system simultaneously has been presented. The implementation of algorithm on IEEE 5, 14 and 30 bus system to generate Pareto-Optimal front in less than single run (Partial run) has been discussed.

**Chapter- 8:** Chapter 8 provides conclusions drawn from various improved and modified intelligent techniques applied to solve the Economic Load Dispatch (ELD) and Multiobjective Economic Load Dispatch problem (MELD) problem. Some suggestions are also presented for further work in the areas covered in this thesis.

# CHAPTER 2
# LITERATURE SURVEY

## 2.1 INTRODUCTION

To meet the demand of all sectors, generation of electrical power is essential. Economic Load Dispatch problem is an optimization problem [93] which minimizes the total fuel cost of all committed plants while meeting the demand and losses. Real life problems may be nonlinear, non-differentiable and discontinuous. These cannot be solved using classical optimization techniques [143, 145]. Classical techniques have the tendency of settling down at local minima instead of the global best solutions. The optimal power system operation is achieved when various objectives of power systems: cost of generation, system transmission losses, environmental emission etc. simultaneously achieve their minimum value. But these objectives may be conflicting in nature and cannot be handled by conventional single objective optimization techniques. Single objective optimization techniques give the best value of objective under consideration whereas the values of other objectives may not be acceptable at all. Therefore, multiobjective approach [1, 15, 17, 18, 21, 29] has been used to solve such problems. Literature survey has been carried out on intelligent techniques, particularly on Genetic Algorithm, Particle Swarm Optimization and their application to solve Economic Load Dispatch (ELD) and Multiobjective Economic Load Dispatch (MELD) problem.

## 2.2 ELD USING CONVENTIONAL TECHNIQUES

The conventional techniques to solve ELD problem are Simplex linear programming [12], Steepest descent gradient [150], Lambda iteration method, Modified lambda iteration method [129], Merit order reduced gradient, Newton - Raphson method [32], Interior point method, base point and participation factor method, integer programming etc. However, these methods require the incremental cost curves to be monotonically increasing or piece-wise linear. The input/output characteristics of modern units are inherently highly nonlinear due to valve-point effect, ramp rate limits etc. Consideration of highly nonlinear characteristics of the units requires highly robust algorithms to avoid getting stuck at local optima [5, 30, 92, 93, 130].

## 2.3 ELD / MELD USING INTELLIGENT TECHNIQUES

Stochastic search algorithms like Genetic Algorithm (GA) [9, 22, 24, 31, 43], Evolutionary strategy (ES) [13, 44], Evolutionary programming (EP) [20, 23, 33], Pattern search [58], Differential evolution [59, 148, 189], Aartificial bee colony [88, 139, 163], Harmony search [78,100,107,164,180], Biogeography based optimization [89,108,109, 138, 146], Teaching learning-based optimization [147], Cuckoo search [151], Aunt colony optimization [60], Bacterial foraging [149,174], Genetic-fuzzy [45, 167, 177, 187] Firefly algorithm [110, 114, 142, 152], Jaya optimization algorithm [159], etc. have been used to solve the ELD / MELD problem. Saoussen Brini et al. [77] suggested solution of economic environmental dispatching (EED) of hybrid power system including wind and solar thermal energies, using Strength Pareto evolutionary algorithm (SPEA).

In [168] ELD problem for dual objectives has been solved by Flower pollination algorithm (FPA). Results were compared with PSO, Personal best-oriented PSO (PPSO), Adaptive personal-best oriented PSO (APPSO), Modified particle swarm optimization (MPSO) and Adaptive real coded GA (ARCGA). Results shows that cost was minimum for FPA and highest for PSO. W.T. Elsayed et al. [165] suggested Modified social spider algorithm (MSSA) for solving ELD problem and observed that the algorithm overcome the problem of premature convergence. Opposition-based krill herd algorithms [173,179] use behavior of krill heard to find optimization with large population size. Adaptive cuckoo search algorithm has been implemented to solve ELD problem of ten-unit system with multiple fuel options and valve point loading effect in [166]. They compared the results with Genetic algorithm (GA), Improved Genetic Algorithm with Multiplier Updating (IGA-MU) and PSO. Floating search space [181], Grey wolf [162], Efficient cultural particle swarm optimization [99], Ameliorated grey wolf optimization [190], Improved stochastic fractal search algorithm [192], Hybrid intelligent algorithms (Particle swarm optimization (PSO) and Artificial fish swarm algorithm (AFSA)) [193], Deep learning [191] etc. have been used to solve ELD problem.

X. Yu et al. [189] proposed Ensemble multiobjective differential evolution algorithm (EMODE) to solve economic and emission dispatch problem. Results were found to be better than PSO, differential evolution (DE), recursive, improved recursive, Pareto differential evolution (PDE), Nondominated Sorting Genetic Algorithm-II (NSGA-II) and Strength Pareto Evolutionary Algorithm-2 (SPEA-2) and Many Objective Differential Evolution (MODE). This algorithm provides high quality solution. D. Poornima et al. [175] suggested A-loss coefficient method to calculate the transmission losses. A-loss coefficients were derived for any transmission line from the knowledge of load flow analysis at few operating conditions using perturbation method. A loss coefficients were used to solve MOELD problem for 6 generating units for 283.4 MW load using weighted sum approach and Strength Pareto Genetic Algorithms. Conventional methods Newton-Raphson method (NR method) and Genetic algorithms were also used to solve MOELD problem. Results were compared with these algorithms; it was found that Strength Pareto genetic algorithm gives better result. T. M.

Mohammadian et al. [182] suggested the new evolutionary PSO using the three operators Mutation, Crossover and Reproduction to enable the search process to skip local optimal points and enhance computational efficiency. Dynamic inertia weight, cognitive and social weight coefficients were included to improve the exploration and exploitation for smooth convergence.

## 2.4. GENETIC ALGORITHM

Holland [2] presented the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under GA. K.F. Man et al. [14] presented the basic fundamental of GA. In GA size of population was varied according to problem. Crossover and mutation two main operators were required to generate new population for next generation. Probability of crossover rate was varied between 0.6 to 1. Probability of mutation rate less than 0.1 was required to give good performance. Setting of crossover rate and mutation rate was complex for nonlinear optimization problem. Global, Diffusion and Migration were types of parallel GA to enhance the computational speed. In Global GA total population worked as a single breeding mechanism. In migration GA, population was divided in subpopulation. Diffusion GA considered the population as a continuous structure. They also discuss some. advantages of GA. GA is used to solve constraint type problem simply by coding of chromosome. It can be used to solve multimodal, nondifferientable, noncontinuous problems. It can be easily interfaced to existing simulations and models. To understand the working of GA knowledge of few mathematics was required. Some shortcoming of GA has also been discussed in this paper. Some objective functions may be difficult to optimize by GA. Such functions are called as GA-deceptive functions. There is no guarantee of obtaining the global optimal point using GA, although it has the tendency to do so. GA is not suited for analyses that would provide guaranteed response time. It is unwise to apply GA directly to a real system without any simulation model. Following are the applications of GA: Parameter and system identification, Control systems engineering, Robotics, Pattern recognition, Speech recognition, Planning and scheduling, Engineering designs, Classifier system. GA can apply to various fields using integration with other Technologies-Neural network, Fuzzy logic systems. GA can be used to detect Brain tumor [183], Workshop scheduling [185], Parallel indexing the color and feature extraction of images [153], Software testing [76], Feature subset selection [154] etc.

### 2.4.1 Economic Load Dispatch using Genetic Algorithm

Results of GA in terms of accuracy was not effective for large variables. Larger string was not able to search better solutions. These drawbacks of GA were overcome by design of two phase GA (TGA) [9]. In two phase GA, approximate solutions were searched in first phase with shorter string. In second phase the better solution was searched in details around the approximate solution. The decimal coding of variables was compared with the binary coding. Simple GA and two phase GA were implemented on 20 units system to evaluate the optimal solution. The uniform crossover was used to generate new solutions. The crossover rate was 0.9. In GA mutation rate was 0.03, while in TGA it was 0.065 for Phase 1 and 0.015 for Phase 2. In case of Two phase GA, if number of iteration becomes 50, then phase 1 moved to phase 2. Convergence characteristics and accuracy of Two phase GA were better than simple GA. GA and Micro GA models have been suggested in [22] to solve ELD problem of combined cycle and cogeneration power plants in the Thailand. The eastern region of Electricity Generating Authority of Thailand (EGAT) system has 50 buses,55 lines and 15 generators with the installed capacity of 5,206.6 MW. The fifteen generating units in the eastern region of EGAT system consists of one steam turbine plants, eight combined cycle plants, two cogeneration plants and one hydro plant. The cogeneration plants were considered to perform in the same way as combined cycle plants. Three types of load: light load, day load and peak load had been considered. Open cycle operation of the gas turbine causes the cost characteristics of such plants to be, generally non smooth and continuously non-differentiable. GA was coded in binary form and length of string depended on accuracy required. Three types of GA operator's reproduction, crossover and mutation (creep mutation and jump mutation) were applied to solve ELD problem. Probability of crossover was between 0.6 to 0.8 and probability of mutation was between 0.0001 to 0.1. Roulette wheel selection method was used to generate new population. For conventional GA and light load case creep mutation rate was 0.04 and jump mutation rate was 0.01 provided the best solution for ELD. In case of Micro GA, population size was varied and best result was found for population size 28. Results of both the GA models provide reduction in cost for three types of load. It was also observed that Micro GA was capable to give best results for small population size i.e. 27 to 30 in comparison to GA (population size 50). The Micro GA had distinct advantage of providing faster solution as compared to the conventional GA.

H.Ling et al. [31] suggested an improved Genetic Algorithm for Economic Load Dispatch (ELD) with valve-point loading. In the conventional GA genetic operations - crossover and mutation were carried out in a random manner but in this paper crossover operation modified in the form of arithmetic crossover, heuristic crossover and simple crossover. Mutation was modified in the form of uniform and non-uniform mutation. With the proposed crossover operation, the probability of obtaining good off springs increased whereas search domain of the selected gene became smaller. Improved GA was used to solve economic load dispatch problem and the results showed that the proposed GA performed more efficiently and had faster convergence rate. Younes Mimoun et al. [43] developed a combination of two genetic algorithms (GAGA), one to determine the values of the genetic operators and the other to optimize the cost function. Tournament selection has been used to select the strings for crossover. Single point crossover as a first operator was used to explore search space. Mutation as a second operator was used to prevent the premature stopping. Crossover rate between 60% to 100% and mutation rate 0.1% to 5% have been used in GA to solve Economic power dispatch (EPD) of 13 generators to meet the 2520 MW demand. Results of two genetic algorithms (GAGA) has been compared with GA1, GA2, GA3, GA4 and it is found GAGA converged to the global optimum in minimum cost 23681.313 $/hr. and time 33.897 sec. in comparison to GA i.e. Cost 23693.211$/hr. and time 64.87 sec. The iteration required to converge the function reduced when adaptive probabilities for crossover rate and mutation rate were used for GA1, GA2, GA3 and GA4.

Lily Chopra et al. [111] suggested Refined Genetic algorithm to solve the economic load dispatch problem. In GA bit size 16 and population size 20, roulette wheel selection, one-point crossover with 0.5 crossover probability and 0.01 mutation probability was used. In Refined Genetic Algorithm (RGA), computational speed was increased using Elitism for bit size 16 with population size 100 by decreasing the probability of crossover from 0.7 to 0.6 exponentially, mutation rate was increased from 0.001 to 0.1 exponentially. Hong et al. [24] studied the effectiveness of GA for a system consisting of multiple co-generators and multiple buyers in a deregulated market. They implemented it successfully on IEEE 30-bus system and IEEE 118-bus system. Sunny Orike et al. [140] proposed a strategy that replaces the worst solutions of the new population with the best solutions of the current population. Genetic algorithm with this developed concept was called elitist genetic algorithm (EGA). Constrained elitist genetic algorithm (CEGA) used the equality constraint and inequality constraint to solve Economic load dispatch problem of Nigerian power plant. The proposed approach was found to perform better than conventional and Micro genetic algorithm. Bishnu Sahu et al. [141] suggested the application of GA and Quadratic programming concept to solve ELD problem for IEEE 14 and 30 bus systems. They derived the transmission loss formulae in terms of generating power.

### 2.4.2 Multiobjective economic load dispatch using genetic algorithm

J. X. Xu et al. [16] presented application of GA to solve Multiobjective Economic and Environmental Emission Dispatch problem. The combination of economic and the environmental objectives were represented by single objective equation as given below:

$$F = W * F_C + (1-W) * F_E$$

When W = 0 and W = 1, only environmental objective and economic objective have been considered respectively. By varying the value of W Pareto Front has been obtained. To evaluate the function three operators of GA, selection, one-point crossover and mutation have been used. Mutation is required to ensure that no point in the search space has a zero probability of being explored. The reproduction process iterates until no improvement on the solution can be obtained.

M.A. Abido [28] presented the comparative study of Multiobjective Evolutionary Algorithm (MOEA) for Environmental Economic Emission Dispatch (EED) problem. Niched Pareto genetic algorithm (NPGA), Non-dominated Sorting Genetic Algorithm (NSGA) and Strength Pareto Evolutionary Algorithm (SPEA) were applied to the standard IEEE 30 bus, 6 generator test system. Tournament selection was applied to individual in the current population was applied to SPEA.

M.A. Abido [25] presented a novel Multiobjective evolutionary algorithm (MOEA) for Environmental Economic Emission dispatch (EED) problem considering both equality and inequality constrains. A new Non-dominated sorting genetic algorithm (NSGA) was used which preserved the diversity of population and overcame the problem of premature convergence. Hierarchical clustering mechanism was used to reduce the Pareto-Optimal set to a desired number. Fuzzy set theory was used to extract the best solution. Real coded genetic algorithm with blend crossover and non-uniform mutation operator was implemented on IEEE 30 bus, 6 generator test systems. The results were compared with Linear programming (LP), Multiobjective stochastic search technique (MOSST). Novel NSGA was found to be the most efficient.

Chao - Lung Chiang [40] proposed an integration of multiple fuel changes and valve point loading effects for solving power economic dispatch (PED) of third order cost function units for a load of 1400 MW using Improved genetic algorithm with multiplier updating (IGA-MU). IGA-MU was more effective than conventional GA-MU approach.

Lahouari Abdelhakem Koridak et al. [62] presented bi- objective Economic and emission dispatch problem by Genetic algorithm with line flow constraints. The algorithm was applied to IEEE 30 bus system with 06 generating units and 41 transmission lines with four tap changing transformers. The total system load demand was 283.4 MW. Two objectives were optimized using a factor of hybridization (Hi). Tournament selection method, whole linear crossover and nonuniform mutation operator were used in GA. Elitist strategy was used to preserve the best possible solution. The proposed GA was found to be faster and more effective than the GA and Evolutionary programming (EP).

M. A. Abido [47] presented Multiobjective evolutionary algorithm (MOEA) for Environmental Economic Emission dispatch (EED) problem. Three MOEA algorithms -

Non-Dominated Sorting Genetic Algorithm (NSGA), Niched Pareto Genetic Algorithm (NPGA) and Strength Pareto Evolutionary Algorithm (SPEA) have been applied to economic / environmental electric power dispatch problem of standard IEEE 30 bus, 6 generator test system. A comparative study among the MOEA techniques has been carried out. In all the techniques (NSGA, NPGA, SPEA) Pareto optimal set was generated by implementing average linkage based hierarchical clustering algorithm and the best compromise solution has been selected by Fuzzy based mechanism.

Y. S. Brar et al. [45] presented Multiobjective load dispatch problem of 5 generators, 11 nodes power system set using Genetic fuzzy technique. They obtained real and reactive power transmission line flows by Generalized Z-bus Distribution Factor (GZBDF). Fuzzy based number of functional operating constraints such as equality and inequality constrains for real and reactive powers flows were included as penalties in the fitness functions, which guaranteed the optimal solutions. A fuzzy based penalty was imposed on any unsatisfaction of equality and inequality constraints. The proposed method was applied for economic emission power dispatch problem with and without security constraint and was observed to give accurate and feasible solutions with reasonable computational time.

In [113,176], Noninferior set was obtained for IEEE 5, 14 and 30 bus systems considering two objectives, cost of generation and transmission losses and for three objectives, considering cost of generation, transmission losses and environmental pollution using GA. MELD problem was formulated by ε - constraint method. In both the papers Noninferior set was obtained for IEEE 5, 14 and 30 bus systems by executing multiple runs of optimization problem. R. Quiddir et al. [39] suggested Economic dispatch of Electrical power plant of Western Algeria using Genetic algorithm. Parameters of

GA i.e. population size = 10; Crossover probability = 0.85 and Mutation probability = 0.1 has been used to solve EDP of 505 MW Plant for two cases. In first case transmission line losses calculated by Newton Raphson method i.e. 15.94 MW was constant, and in the second case transmission losses were considered as a linear function of real generated power. Results of both the cases were compared with Fletcher – Reeves and Fletcher methods. It was observed that economic cost obtained by GA was minimum in both the cases.

C. L. Chiang [54] developed the IGAMU, which hybridize the Improved Genetic Algorithm (IGA) with the Multiplier updating (MU). An Improved Genetic Algorithm (IGA), has two operators an improved evolutionary direction operator (IEDO) and a migrating operator, to reduces the effort required to explore the solutions, search and to maintain the diversity in small population size. The system constraints of power economic load dispatch (PELD) problems were managed by introduction of Multiplier updating (MU) and it was able to avoid deforming the augmented Lagrange function. Proposed IGAMU had been implemented on real life problem of power economic load dispatch (PELD) of different sizes. A binomial mutual crossover was used for small population size to increase the local diversity of individuals. To implement the improved evolutionary direction operator (IEDO) three best solutions in each generation has been selected and then new solution becomes superior to the original best one and it reduces the blind search. A migration was included in the IGA to regenerate a newly diverse population, preventing individuals from gradually clustering and thus significantly increasing the amount of search space explored for a small population. The original objective function could be scaled to prevent ill-conditioning by updating penalty parameters and multipliers. The advantages of the proposed IGAMU are that the IGA efficiently searches the optimal solution in the economic dispatch process and the MU effectively tackles system constraints. IGAMU has the following setting of parameters: iteration number of the IEDO operation 4; the population size 5. First case: 13-unit system considering valve-point loadings to meet load demand of 2520 MW, without transmission loss; the proposed IGAMU has been implemented on this system and compared with GA-MU and found more effective and efficient than the GA–MU. Case 2: 15 online units supplying a system demand of 2650 MW. Among these dispatching generators, units 2, 5 and 6 have three prohibited operation zones (POZs), and unit 12 has two POZs, forming 192 decision subspaces for this realistic system. This complex optimization problem contained one objective function with 15 variable parameters, one equality constraint, and four inequality constraints, since four units had the POZs. Results shows that the proposed IGAMU also has the lowest feasible cost of all methods tested. For case 2: Algorithm has been compared with deterministic crowding GA (DCGA), integrated artificial intelligence (ETQ), is a hybrid algorithm of thee algorithms: Evolutionary Programming (EP), Tabu search (TS) and Quadratic programming (QP), evolutionary strategy optimization ESO and GA-MU. Hence, for PELD problems of different size and complexity, the proposed IGAMU proved to be the best algorithm among those surveyed. The proposed algorithm combines the IGA and MU, it only adopts the IGA to solve the objective function and does not concern the penalty parameters or multipliers. The MU could manage system constraints by automatically updating the penalty parameters and multipliers Therefore, the proposed algorithm was easier to implement than fixed penalty- based optimizations. IGAMU had straightforward concept; easy implementation; better effectiveness than previous methods; better effectiveness and efficiency than the GA-MU; automatic adjustment of the randomly assigned penalty to an appropriate value and the requirement for only a small population in realistic PELD problems.

## 2.5 PARTICLE SWARM OPTIMIZATION

James Kennedy and Russell Eberhart [10] developed Particle swarm optimization inspired by behavior of flock of birds and school of fishes and recommended the same for solving wide range of nonlinear functions. They suggested that Personal best (Pbest) and Global best (Gbest) function value used by PSO had analogy with crossover operation used by Genetic algorithm. Like other evolutionary computational models, it uses the concept of fitness. A set of agents (Particles) is used to evolve the function value. These agents (Particles) were used to explore the search space with changing velocity and positions to obtain the optimal solution. Russell Eberhart and James Kennedy [11] introduced new optimizer using Particle swarm theory. They tested and analyzed three versions of PSO. (i) 'GBEST model', which uses the information of group's best value and (ii) two versions of 'LBEST model' one with a neighborhood of six and other with a neighborhood of two. All the models were tested on benchmark functions and it was also proposed that these models can be used for training of neural network and robot task learning. Yuhui Shi et al. [19] introduced a new parameter called inertia weight (IW) in Particle swarm optimization technique. Its effect on the performance of PSO was studied using a mathematical benchmark function- Schaffer's f6 function. It was observed that when IW was in the range (0.9-1.2), the technique had a bigger chance of reaching to the global optimum in a lesser number of iterations. It was also observed that linearly decreasing IW improved the performance of PSO to a large extent. K.E. Parsopoulas et al. [26] published a review paper in which the authors have covered research papers up to 2002. In this paper ability of PSO in tackling multiobjective, minmax, integer programming and $l_1$ errors-in-variables problems as well as in noisy and continuously changing environments has been described. The authors also concluded that PSO gives promising results even when the size of the swarm is very small. Another survey paper was published by Keisuke et al. [73]. In this paper, they have covered papers up to 2008. In this paper, basically the progress of PSO, since its inception in 1995 is reviewed and modifications in the basic PSO for improving exploitation and exploration is also suggested. Many variants were investigated and proposed for further improvements.

C.A. Floudas et al. [74] presented an overview of research progress in global optimization during 1998-2008. The areas of twice continually differentiable non-linear optimization, mixed- integer non-linear optimization, optimization with differentiable algebraic models, semi-infinite programming, optimization with grey box / nonfactorable models and bi-level nonlinear optimization were covered. Boonserm et al. [41] also presented an investigation on PSO. N. K. Jain et al. [169] presented a review paper on PSO, which covered research papers from 1995 to 2016.

Many variations of PSO have been proposed for the basic PSO: Comprehensive learning particle swarm optimizer (CLPSO) [46], Orthogonal learning particle swarm optimization (OLPSO) [101], self-learning particle swarm optimizer (SLPSO) [112], evolutionary game based particle swarm optimization (EGPSO) [55] and particle filter based on organizational adjustment particle swarm optimization (OAPSO-PF) [136]. Improved particle swarm optimization has been suggested in [50,61,67, 80, 85, 94, 98, 124]. Ajith Abraham et al. [48] implemented PSO and ACO algorithms on some mathematical benchmark functions as Griewank function, Schwefel function, Quadratic function and also on real world applications as Travelling sales man problem and Data mining problem. They also analysed and discussed the results in detail. Ismael et al. [49] used pattern search based algorithm for the global minimization of a function without the use of derivatives and conveyed to the stationary points starting from any arbitrary points. Zhiyu you et al. [86] proposed an Adaptive weight PSO with constriction factor (CF-AW-PSO) to overcome the problem of premature convergence. The value of inertia weight was set according to changes in the value of objective function. The algorithm was implemented on four standard benchmark functions and also compared with different types of optimization algorithms of PSO. The proposed algorithm showed better performance. A number of papers were published on applications of PSO: Quadratic assignment problem (QAP) [51], Location assignment problem [57], Weight optimization for evaluation [75], Equipment possession quantity [70], Mechanical optimization problem of single gear reduction [71], Flexible job shop scheduling problem [91], Medical imaging [102], Color quantization [87], Power quality and reliability improvement of distribution system [104], Artificial neural networks training with uncertain data [105], Discrete combinatorial optimization problem [106], Economic power dispatch problem with generator constraints [144], Optimum design of PID controller in AVR system [35], Reactive power compensation [127] etc. Saibal K. Pal et.al [114] compared Firefly algorithm with PSO for solving noisy non linear problems and reported firefly algorithm to be better than PSO for higher levels of noise.

J.J. Saiman et. al. [115] compared distributed generator sizing using three types of PSO - Rank evolutionary PSO [REPSO] method, Evolutionary PSO [EPSO] and traditional PSO. REPSO was shown to be superior than PSO and EPSO for determining the optimum size of distributed generation in 69 bus radial distribution system. They observed that implementation of EP in PSO allowed all the particles to move towards the optimum value faster.

Bharat Bhushan and Pillai [142] compared PSO and firefly algorithm (FFA). Ten standard nonlinear functions were chosen. Elapsed time and mean value of the function were evaluated for PSO and FFA. PSO was found to be faster than FFA for most of the nonlinear functions. The mean value of function and elapsed time were also found to be much smaller for PSO.

Many researchers suggested modifications in the original algorithm.W.B. Langdon et al. [36] suggested Kernel to provide the values for each particle of a swarm which guides the unit as a whole. They solved one- dimensional multi-model 3-peaks and Rastrigin function problem using kernel. M. S. Voss [37] suggested principal component PSO: PCPSO in which particles were made to fly in two separate spaces simultaneously, one in traditional n-dimensional space and a rotated m-dimensional z- space where $m \leq n$. PCPSO algorithm has been implemented on Greiewank function.

Jaco F. Schutte et al. [38] studied the variants of PSO algorithms and applied to Dixan - Szego test set. The variations studied were: constant inertia weight, linear inertia reduction, limit on maximum velocity, constriction factor, dynamic inertia and maximum velocity reduction. They observed that constriction and dynamic inertia weight both affected reliability and found dynamic inertia reduction to be less sensitive than constriction factor. Wei-Bing Liu et al. [55] introduced Evolutionary game PSO (EGPSO) in which the behaviour of particles was modelled using replicator dynamics and multi - start technique. This technique overcame premature convergence and had better convergence property than traditional PSO. Wei Zu et al. [56] proposed a new technique PSOED based on particle swarm equilibrium distribution in which a sub-optimum trap i.e. clustering of particles within a subarea of problem scope is avoided. This technique was applied to various benchmark functions and was found to be better than basic PSO and GA. Serkan et al. [103] introduced multi-dimensional PSO (MD-PSO) where swarm particles can seek both positional and dimensional optima. They also proposed FGBF (Fractional global best function) technique to avoid premature convergence and applied to multi model dynamic environment, to track the global optima with minimum error. Yutaka et al. [64] proposed combinations of the particle swarm optimization and the simultaneous perturbation optimization method to optimize test functions to know convergence properties such as convergence rate or convergence speed. The proposed technique had good global search and effective local search capability. Md. Sakhawat Hossen et al. [72] also tried an adaptive Particle swarm optimization based on behavior of spider. They presented a comparison with traditional PSO and formed the suggested method to improve the performance. Chen-Chien et al. [63] experimented hybridization of PSO with Nelder - Mead simplex approach for multi-dimensional optimization problems. They reported the new algorithm increases the convergence rate and accuracy. Junqi Zhang et al. [65] proposed a new algorithm combining PSO with advanced and retreat strategy and clonal mechanism. They reported their algorithm to be computationally more efficient and more accurate. Zhi- Xiang - hou [66] developed an adaptive Particle swarm optimization algorithm and claimed this to be more effective and highly accurate. Yan Jiang et al. [50] proposed a population of points sampled randomly from the feasible space. Then the population is partitioned into several sub-swarms, each of which is made to evolve based on particle swarm optimization (PSO) algorithm. At periodic stages in the evolution, the entire population is shuffled, and then points are reassigned to sub-swarms to ensure information sharing. This method elevates the ability of exploration and exploitation. Simulations for three benchmark test functions show that IPSO possesses better ability to find the global optimum than that of the standard PSO algorithm. Rui Li et al. [79] proposed algorithm could expand the control point of the searching area and optimize convergence speed. It sets swarm for each control point and then every swarm searches best point collaboratively through shared information, so it avoids the premature deficiency in traditional PSO algorithm. Xu Jun et al. [80] suggested three binary versions of PSO: Binary improved particle swarm optimization (BIPSO), Binary simulated annealing particle swarm optimization algorithm (BSAPSO), Binary cross particle swarm optimization algorithm (BCPSO). The results of all three algorithms were compared on the basis of convergence speed, global optimization capacity and the stability of algorithm. It was found that, the binary improved particle swarm algorithm, was better than the other three algorithms. Jun Tang et al. [81] formulated fine tuning hybrid PSO and found it to be better than other forms of PSO when applied on some benchmark functions. Wei Wang et al. [82] suggested improvement in premature convergence of high dimensional function. They tried Chaotic search for jumping out of local optimum. They demonstrated better convergence property and accuracy than traditional PSO for high dimensional problems.

Bilal Benmessahel et al. [85] illustrated the effect of excluding the redundant particle from current iteration. Huanhuan Ji et al. [90] proposed a bi-swarm particle swarm optimization with cooperative co-evolution (BPSO-CC). In this model second swarm was generated from the first swarm which conducted the local search. They implemented the proposed technique on benchmark functions in dimensions of 100 to 500 and it observed that BPSO-CC performed better than the standard PSO (SPSO) in terms of speed and precision. Zhe Li et al. [87] have introduced a new PSO with parallel processing and color quantization. Nai-Jen Li et al. [96] proposed an improved PSO in which different weights with different particles of the swarm have been attached. Na Li et al. [95] explained that a basic PSO can miss the optimum point. They suggested a modified PSO with niche particle and Bernoulli trap. Changming Ji et. al. [97] developed Catfish effect particle swarm optimization (CE-PSO) by introducing catfish particles through a startup device. This algorithm was implemented to solve reservoir optimal scheduling problem and compared with standard PSO and chaotic PSO (CPSO).CE-PSO was found better than PSO and C-PSO in terms of global search ability and convergence speed.

Weidong Ji and Keai Wang [98] combined PSO with gradient method, which avoids immature convergence. Kyle Robert Harison [116] hybridized GA with PSO. He found that new version of PSO overcame premature convergence. Yan Zhe ping et al. [117] also presented a PSO with two sub populations. Nikhil Padhye et al. [118] suggested three different PSOs with boundary handling approaches. In this paper, the authors have proposed two boundary handling methods - inverse parabolic spread distribution and inverse parabolic confined distribution. These were compared with existing boundary handling methods: Random, Periodic, Set on boundary, SHR (Shrink : the goal of SHR method to re-adjust the particle’s velocity) and Exponential distribution for four test functions. Inverse parabolic spread distribution was found to be the most robust and consistent method.

Zahra Beheshti et al. [134] proposed binary accelerated PSO. They have shown that new PSO required only common controlling parameters viz no. of generations and population size. Luis Miguel Rios et al. [135] presented a review of derivative free algorithms including PSO for constrained problems. They combined twenty-two (22) such algorithms and implemented on a test set of 502 problems. It was observed that all solvers provided the best solution for at least some of the test problems and there is no single solver which provides best result for all the problems. Zhimin chen et al. [136] presented an organizational adjustment PSO based particle filter (OAPSO-PF) algorithm which allowed the particles to adopt to environment and reach the global optimum.

Qi Luo et al. [67] developed a hierarchical structure poly – particle swarm optimization (HSPPSO) approach using the hierarchical structure concept of control theory. This algorithm was implemented on four benchmark functions – Spherical, Rosenbrock, Griewank and Rastrigin and was also compared with PSO. HSPPSO was found to search better for global optimum and converged faster.

Some researchers suggested the upgradation of velocity to improve the speed and convergence of basic PSO. Arasomwan Akube et al. [137] tried a PSO which upgrades the velocities based on Euclidean distance between particles. Chunming Yang et al. [42] proposed a new particle swarm optimization method (NPSO) considering the personal worst and global worst position as a guide for updating the velocity in PSO. The authors compared NPSO and PSO based on four different benchmark functions. The parameters were set to the following for both PSO & NPSO: $c_1=c_2=2$; Range of dimension [-50,50], of each function:2, 5, 10. With these settings, NPSO was found to give better results. But this was not a definite conclusion and more work needed to be done to find out condition under which these techniques give better results.

Lu Baiquan et al. [119] have suggested a control system based strategy for designing PSOs. In this paper they revised the formulas for speed and position of PSO which resulted in improvement in speed of convergence as well as premature convergence. The authors carried out simulations on 13 benchmark functions and found the new algorithm GPSO to be more robust. GPSO was based on stability theory of discrete system to analyze the existing particle swarm optimization to overcome the stuck in local minima and low convergence speed. Proposed GPSO is better than existing PSO in the robustness and the convergence. Many researchers have attempted to hybridize the PSO with various other techniques of optimization and claimed improvements from performance point of view. Ying Ping Chen et al. [53] proposed a new hybrid methodology called particle swarm optimization with recombination and dynamic linkage discovery (PSO-RDL) and implemented on four benchmark functions and a real world power system problem of economic dispatch. It was observed that the performance of PSO-RDL was comparable to that of advanced evolutionary algorithms – classical EP, fast EP (FEP), modified EP, improved FEP, as well as modified PSO (MPSO). Li Jian et al. [68] calculated the new positions of particles with the help of Genetic PSO (GPSO) as well as original PSO termed as OPSO in every iteration and then selected the better positions. They defined this method as Dual-PSO, and claimed it be more consistent in comparison to GPSO as well as OPSO. They did experiments on constraint functions. Jiao –Wei et.al. [69] implemented Elite Particle Swarm Optimization with mutation. In this method bad particles were replaced by elite particles. To avoid local convergence mutation was used. Suggested techniques were compared with linearly decreasing weight PSO and demonstrated an improvement.

Jong-Bae Park et al. [94] suggested an improved PSO using chaotic sequences combined with the conventional linearly decreasing inertia weights and adopting a crossover operation scheme to increase both exploration and exploitation capability of the PSO. The proposed Improved particle swarm optimization (IPSO) was applied to three different ED problems with valve-point effects, prohibited operating zones with ramp rate limits as well as transmission network losses, and multi-fuels with valve-point effects. For each ED problem, four strategies are applied and compared: The conventional PSO with the proposed constraint treatment strategy (CTPSO); PSO with chaotic sequences (CSPSO); PSO with crossover operation (COPSO); PSO with both chaotic sequences and crossover operation (CCPSO). It was also applied to the large-scale power system of Korea. Zhicheng Qu et al. [124] proposed a new algorithm MPSO, which uses novel mutation operator to increase the global search ability. Five constrained benchmark functions were selected to study MPSO and compared its performance with PSO and another variant of PSO. The results of MPSO were found to be better in all cases except for constrained quadratic function. Ming-chen et al. [125] combined mutative scale chaos method with PSO to give better precision, success ratio, robustness and efficiency. Liu Jin –Yue [126] also experimented mutation in PSO to achieve better convergence rate.

PSO with discrete crossover has been suggested by AP Engelbrecht [128] to increase convergence speed and quality of solutions. Six discrete crossover operators were proposed for incorporation into a global best particle swarm optimizer. The performance of these discrete crossover operators was compared with that of the global best particle swarm optimizer and amongst one another to identify the best performing discrete crossover operators. The best operators were then compared with particle swarm optimizers those make use of blending crossover operators.

M. N. Abdullah et al. [155] proposed the time-varying acceleration coefficients to update the velocity of particles to prevent the premature convergence and enhance the performance and robustness of PSO. They transformed the Multiobjective Environmental Economic load dispatch problem into a single-objective problem using the weighting method and determined the Pareto–Optimal front by varying weights. Therefore, multiple runs were carried out to determine the Pareto front.

N. Mishra et al. [156] formulated the MELD problem as weighted sum of fuel cost and environmental pollution objectives. They implemented PSO to obtain the Pareto front for different loading conditions by varying relative weights. The problem was executed as many times as the number of Pareto-optimal points required.

### 2.5.1 Economic Load Dispatch using Particle Swarm Optimization

Hardiansyah et al. [120] presented application of PSO to solve the economic load dispatch problem. The results were demonstrated for standard 3-generator and 6-generator systems with & without consideration of transmission losses. The final results obtained using PSO were compared with conventional quadratic programming technique and were found to be encouraging. In this paper linearly decreasing inertia weight was varied from 0.9 to 0.4.

Jaya Sharma et al. [131] presented a review on application of PSO to solve economic load dispatch problem. Conventional PSO has many issues like global optimal solution, global search ability, premature convergence, convergence speed and stuck in local optima. This review paper presents many new algorithms proposed by different authors to address these issues. The modifications proposed were: dynamic inertia weight, fuzzy tuned inertia weight PSO with wavelet theory, simulated annealing PSO (SA-PSO), Genetic PSO (G-PSO) and Quantum inspired PSO (Q-PSO). All these modifications were applied to solve economic load dispatch problem. The use of such modifications led to better global optimum solution and the global search ability of the algorithm also improved. The use of decreasing inertia weight PSO (DIW-PSO), self-organizing hierarchical PSO (SOH-PSO) were used to avoid the premature convergence. This also resulted in faster convergence.N. Phanthuna [132] presented case study for 40 generation units with 6 load patterns to solve economic load dispatch problem by PSO. Test results show that PSO have more stable convergence characteristics than other stochastic methods. Hardiansyah suggested a new approach in [133] for ELD problem with valve point effect. He developed modified PSO using Gaussian and Cauchy probability distributions to ensure convergence of the Particle Swarm Algorithm. The results were obtained for six, thirteen and forty generating unit systems.

In [184] PSO with moderate random search strategy (MRSPSO) has been applied to solve ELD for six generator systems considering ramp rate limit for improving the ability of particles to increase the convergence rate by exploring their solution space more effectively. The result of MRPSO was compared with other heuristic techniques and was found to be better than all other techniques. R. Shankar et al. [161] suggested application of PSO and CPSO to power system economic load dispatch with ramp rate limit constraints.

### 2.5.2 Multi objective Economic Load Dispatch using Particle Swarm Optimization

Multiobjective approach to optimization has been attempted through PSO [26,178], Discrete Multiobjective PSO [83], a competitive and co-operative co-evolutionary approach [84], graph based PSO [121], and vector evaluated PSO [122]. In all these papers the results have been discussed qualitatively. Lingfeng Wang et al. [52] proposed fuzzified multi-objective particle swarm optimization (FMOPSO) algorithm for obtaining Pareto – Optimal front for Economic Environmental dispatch problem. They compared the solutions obtained with those obtained by Weight aggregation (WA) and Multiobjective optimization evolutionary algorithm (MOEA) and found that solution obtained by these two approaches were dominated by FMOPSO. Also the diversity of solutions obtained by proposed approaches was found to be the highest due to use of diversity preservation mechanisms - fuzzification, Nitching and turbulence factor.

B. Taheri et al. [177] suggested Multiobjective Economic load dispatch problem by considering environmental pollution for 6 generating units using particle swarm optimization algorithm. The best solution was obtained by fuzzy decision function. Yu et al. [189] proposed Ensemble multiobjective differential evolution algorithm (EMODE) to solve Economic and Emission dispatch problem for 6 generators. Results were compared with PSO, Differential Evolution (DE), recursive and improved recursive. The proposed algorithm accelerated optimization with population diversity. T. Aruldoss et al. [34] presented a novel and efficient method for solving economic dispatch problem (EDP), by integrating the PSO technique with sequential quadratic programming (SQP) technique. This was used to prevent the premature convergence and poor time tuning of the final solution. Gbest value of PSO is the starting point of SQP. Gbest was replaced with the final solution obtained using the SQP. PSO-SQP was tested on 3 cases of EDP. Case 1: PSO-SQP hybrid technique was applied on 3 generating units for power demand 850 MW. Swarm size was varied from 10 to 50 in steps of 10. The inertia weight was varied from 0.9 to 0.4 in the steps of 0.1. The results obtained by the proposed method were found to be better than those obtained by other methods. Case 2: PSO - SQP technique was applied to 13 generating units for power demand of 1800 MW and 2520 MW. The problem was solved for two different power demands to show the effectiveness of the proposed method in producing quality solutions. Fuel cost for both power demands was found to be minimum and significant reduction in simulation time was also achieved. Case 3: In this case 3, PSO-SQP technique was applied on 40 generating units for 10,500 MW demand. They compared results with EP, EP-SQP, PSO and it found mean cost value and simulation time by the proposed method to be comparatively lesser. They also found that proposed technique gave high quality solutions with fast converging characteristics. Moderate Random

Search Algorithm (MRPSO) [157] has been applied to solve MELD problem considering fuel cost and emissions. MELD problem was formulated as weighted sum of objectives and only single solution was generated using MRPSO. MRPSO was compared with other algorithms and MRPSO was found to be the most efficient and fastest. In [160], MELD problem has been formulated as weighted sum of cost of generation and transmission losses. Pareto Front was obtained for IEEE 5, 14 and 30 bus systems by PSO algorithm. MELD was solved many times to obtain the Pareto Optimal front. It is because the weights are to be varied randomly to obtain the entire range of Pareto - optimal Front.

In [123], IEEE 118 bus and 14 generating units MELD problem was formulated as weighted sum of fuel cost and environmental pollution. For different loading conditions Pareto Front was obtained by varying relative weights. The problem was executed as many times as the number of points required on the Pareto- Front. Another variation of PSO-Time varying acceleration based PSO [TV- PSO] was proposed in [158] for solving environmental / economic dispatch problem. Pareto - Front was obtained for IEEE 30 bus system by executing the problem many times. In [155] multiobjective environmental economic dispatch problem was formulated as weighted sum of objectives. The Pareto Front was generated using PSO algorithm in which time-varying acceleration coefficients were used to update the velocity of particles. Again, multiple runs were carried out to obtain the Pareto Front. In [159] combined economic emission dispatch (EED) was solved by PSO, BPSO, DE and Jaya algorithm. All the algorithms were compared for various loads for 14 test systems.

## 2.6 RESEARCH GAPS

After carrying out a literature survey on intelligent techniques and particularly Particle swarm optimization, it has been identified that the following are the research gaps in intelligent techniques:

1. It is believed that Intelligent techniques for optimization are slower as compared to the conventional techniques at least for well-behaved problems.
2. Above belief is based on the fact that, conventional techniques start with an initial estimated solution and proceed towards the optimal point where as Intelligent techniques start with an initial population of points and the population of points converges to the optimal point. This is the main weakness of intelligent techniques from computational efficiency point of view. However, the strengths of Intelligent techniques from computational efficiency point of view need to be explored. Therefore, the possibility of Intelligent techniques to be faster than conventional techniques cannot be ruled out, rather shouldn't be ruled out, at least by innovative minds, without having honestly worked on this aspect.
3. The computational efficiency of technique should be measured by a parameter which is independent of technology and the time taken by each iteration.
4. The noninferior set and the Pareto-optimal Front for MELD problem considering two objectives: cost of generation and system transmission losses simultaneously and considering cost of generation, system transmission losses and environmental emission simultaneously have been achieved by executing multiple runs at least equal to the points on Pareto- optimal front. The effort should be made to attain the Pareto-optimal Front in a single run or less than a single run. This is also a motivation for the present research work. Some success has also been achieved.

## Research Publication

[1] N. K. Jain, Uma Nangia, Jyoti Jain,"A review of Particle Swarm Optimization", Journal of the Institution of Engineers India) : Series B, Springer, 99(4), pp.407-411.

# CHAPTER 3
# INTELLIGENT OPTIMIZATION TECHNIQUES

## 3.1 INTRODUCTION

Real life problems may be nonlinear, non-differentiable and discontinuous. These cannot be solved using classical optimization techniques. These techniques make use of differential calculus in locating the optimum solution and generate a single point at each generation by a deterministic computation. The sequence of points approach an optimal solution. Classical techniques have the tendency of settling down at local minima instead of the global best solutions.

Recently, a number of intelligent search techniques: GA [2, 9,14, 16, 22], PSO [10, 11], ABC [88, 139, 163], TLBO [147], Opposition-based krill herd algorithms [173, 179], GWO [190, 162], Deep learning [191], Firefly Algorithm [110, 114], Improved particle swarm optimization [171, 172], Efficient Cultural Particle Swarm Optimization [99], and Improved stochastic fractal search algorithm [192], etc. have been developed. These techniques are derivative free, simple to implement and have the capacity to overcome the problem posed by local optima in large search space.

In this chapter, two intelligent techniques: GA and PSO are discussed. Both GA and PSO have been extensively employed to solve complex problems of various fields including engineering, economics, marketing product design, manufacturing scheduling, trading strategies, aircraft wing design, queuing problems, economic load dispatch, power system control, process control, power quality and reliability improvement in distribution system, optimal design of PID controller in Automatic Voltage Regulator (AVR), Medical imaging, flexible job scheduling problem, multiple fault diagnosis problem etc.

## 3.2 GENETIC ALGORITHM

Genetic Algorithm (GA) is a global search technique used in computing to find solutions of optimization and search problems. Genetic Algorithm is a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. Computationally, GA is a maximization process. Minimization problems are transformed into maximization problems by some suitable transformation. In general, a fitness function F(x) is first derived from the objective function and used in successive genetic operations. In this algorithm, population of points is generated at each generation. The best point in the population approaches an optimal solution using different operators of Genetic Algorithm. GA works with a coding of the parameter set and not the parameter themselves. GA uses payoff (objective function) information, not derivatives or other auxiliary knowledge. GA uses probabilistic transition rules not deterministic rules. The evolution process stops when some predefined stopping condition is satisfied.

### 3.2.1 GA Operators

Following are the important GA operators:

**Reproduction:** Reproduction is a first operator applied on a population. It selects good string in a population and forms a mating pool. This operator is known as selection operator. Selection rules select the individuals, called parents those contribute to the population at the next generation. Parents are selected according to their fitness values. This process also determines which populations (solutions) are to be preserved, called Elitism; in this process, the best individuals of the current generation is transferred to the next generation without applying the crossover and mutation operators. Selection process also decides which population deserve to die out. The primary objective of the selection operator is to emphasize the good solutions and eliminate the bad solutions in a population while keeping the population size constant.

**Crossover**: Crossover is a genetic operator, responsible for the search of new strings. It combines two parents (present generation) to form children (new generation) for the next generation / iteration.

**Mutation**: Mutation rules apply random changes to individual parents to form children. The mutation operator changes 1 to 0 and vice versa with a small mutation probability. This is used to maintain diversity in the population.

### 3.2.2 Steps to Optimize a Function Using Genetic Algorithm

Following are steps to optimize a function using Genetic Algorithm:

(1) Design the algorithm: Choose the population size, operators and stopping criteria.

(2) Initially generate population randomly (in binary form) between the ranges of variables. Population in binary form is called as string.

(3) Decode the population in decimal values and scale the decimal values in the range defined by the variables of function (using upper and lower range).

(4) Calculate the fitness value for each string.

(5) Select the individual strings according to their fitness value for the next generation. The strings, which have higher fitness value have more probability to generate new population.

(6) Apply the crossover operator on 80% of population (one point, two point, mid-point etc.) to generate new population.

(7) Implement the mutation operator on 10 % population. The most common form of mutation is to take a bit from strings (chromosome) and alter it with some predetermined probability. Practical aspect of mutation is to include the member from outside having different gene to create the next generation. The purpose of this is to widen the search space. Another purpose is to create a member of population, which may prove to be better than any other member of the population. It is also equivalent to carryout crossover of a member of present population with a member external to the population.

(8) If stopping criterion is met, then exit with optimal solution, otherwise go to step 3. (stopping criteria is observed when there is no change in the optimal value achieved among successive generations)

## 3.3 APPLICATION OF GA ON ROSENBROCK FUNCTION

Genetic Algorithm (GA) is implemented on a mathematical benchmark function to study the effect of its various parameters on convergence. Then it is implemented on real life problem of Economic Load Dispatch (ELD) of IEEE 5, 14 and 30 bus systems.

Mathematically, Rosenbrock function is defined as:

Minimize

$$F(x) = 100 * (x_1^2 - x_2)^2 + (1 - x_1)^2 \tag{3.1}$$

Range $x_1 \in (0, 2)$ $x_2 \in (0, 2)$

Since GA maximizes the function, so using transformation, we define the objective function as

Maximize

$$f(x) = 1/(1+F(x)) \tag{3.2}$$

A variable 'x' whose bounds are given by $x^L$ and $x^U$ is represented by a string of 'M' binary bits, and its decimal equivalent is,

$$x = ((x^U - x^L)/(2^M - 1)) * (\sum_{S=0}^{M-1} 2^S b_n) \quad (3.3)$$

$b_n$= $n^{th}$ bit value (0 or 1)

If a continuous variable is to be represented with Δx accuracy, then number of bits in a string is computed as:

$$2^M > ((x^U - x^L)/\Delta x) + 1 \quad (3.4)$$

Let accuracy Δx = 0.2858, then

$$2^M > \left((2-0)/0.2858\right) + 1$$

$$2^M > 7.99$$

M = 3 bit

The bit size is chosen according to the desired accuracy. Table 3.1 shows the various combinations of bit size and population size. Column (3) of Table 3.1 shows the bit size needed for the corresponding accuracy shown in column (2). Column (4) of Table 3.1 shows the population size chosen for corresponding bit size.

**TABLE 3.1**
**Bit size and population size**

| **S. No. (1)** | **Δx (2)** | **Bits (3)** | **Population Size (4)** |
|---|---|---|---|
| Combination 1 | 0.004 | 9 | 10 |
| Combination 2 | 0.008 | 8 | 10 |
| Combination 3 | 0.07 | 5 | 6 |
| Combination 4 | 0.2858 | 3 | 7 |

## 3.4 COMPUTATIONAL RESULTS AND DISCUSSION

Rosenbrock function has been optimized manually as well as using GA Tool box to study the effect of various parameters on convergence. Flow chart to optimize the function is shown in Figure 3.1.

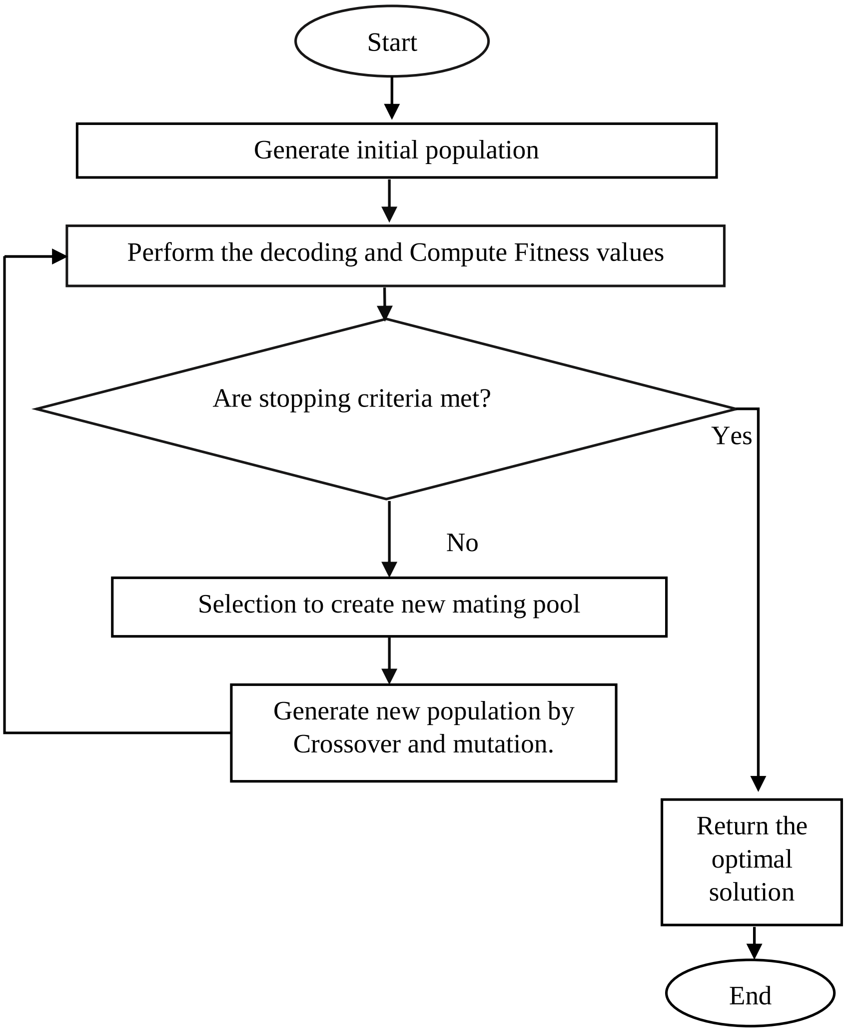



**Fig. 3.1** Flow Chart of Genetic Algorithm

### 3.4.1 Minimization of Rosenbrock Function Manually

Rosenbrock function has been solved manually for combinations of bit size and population size shown in Table 3.1.

## Bit size 9 and Population size 10

Table 3.2 shows the results of generation 1 for Rosenbrock function for combination 1. For population size 10, variables $x_1$ and $x_2$ are generated randomly in binary as shown in columns (2) and (3). Columns (4) and (5) represent decimal values of binary string, calculated using equation (3.3). Columns (6) and (7) represent the decimal values of variables in the specified range of variables. Columns (8) and (9) represent function values for maximization and minimization respectively. The operation of column (10) represents the ratio of respective value of function to sum of f(x) and this is called pre select. Column (11) shows the ratio of respective value of f(x) to average value and this is called expected count. Column (12) depicts the actual count, which means the probability of participation of parent string to move to the next generation. Actual count has been chosen, based on value of column (11). The values that are less than 0.5, in column (11) will not participate in future generation. Therefore, the actual count for such values are 0. The values that are more than 1.5 in column (11) will participate twice in the next generation. Therefore, for such values 'actual count' has been taken as 2. As a thumb rule, actual count has been taken equal to the 'nearest whole number'. Since the actual count for rows 1, 4, 6, 8 and 9 of column (12) are zero, therefore these populations are discarded. The average value of f(x) comes out to be 0.11856. The difference between maximum and minimum value of f (x): (0.4788-0.0025) is 0.4765. Therefore, next generation is generated by performing selection, crossover, mutation and elitism operation. Table 3.3 shows the new population for generation 2 obtained from the previous generation. Similar procedure is adopted to create new population for different generations as adopted in Table 3.2 and 3.3. This process is continued until all the population reaches at the same point (stopping criteria) and this value is the representation of optimal point. This was achieved in nineteen (19) generations.

**TABLE 3.2**
**Results of Generation 1 for Rosenbrock Function**
**(Bit size 9, Population size 10)**

| S. No. (1) | $x_1$ in binary (2) | $x_2$ in binary (3) | Decimal Value of $x_1$ (4) | Decimal Value of $x_2$ (5) | Decimal Value of $x_1$ in specified Range of $x_1$ (6) | Decimal Value of $x_2$ in specified Range of $x_2$ (7) | F(x) (8) | f(x) (9) | f(x)/ sum (10) | Expected count (11) | Actual Count (12) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 111111111 | 111111100 | 511.00 | 508.00 | 2.0000 | 1.9883 | 405.7105 | 0.0025 | 0.0021 | 0.0207 | 0 |
| 2 | 101010101 | 111111100 | 341.00 | 508.00 | 1.3346 | 1.9883 | 4.3969 | 0.1853 | 0.1562 | 1.5624 | 2 |
| 3 | 100100100 | 100000111 | 292.00 | 263.00 | 1.1429 | 1.0294 | 7.6805 | 0.1152 | 0.0971 | 0.9714 | 1 |
| 4 | 100010001 | 111100001 | 273.00 | 481.00 | 1.0685 | 1.8826 | 54.8988 | 0.0179 | 0.0151 | 0.1508 | 0 |
| 5 | 101010110 | 111100011 | 342.00 | 483.00 | 1.3386 | 1.8904 | 1.0886 | 0.4788 | 0.4037 | 4.0372 | 4 |
| 6 | 100011100 | 111000011 | 284.00 | 451.00 | 1.1115 | 1.7652 | 28.0634 | 0.0344 | 0.0290 | 0.2901 | 0 |
| 7 | 100001111 | 110000010 | 271.00 | 386.00 | 1.0607 | 1.5108 | 14.8842 | 0.0630 | 0.0531 | 0.5309 | 1 |
| 8 | 111110000 | 111100010 | 496.00 | 482.00 | 1.9413 | 1.8865 | 355.1221 | 0.0028 | 0.0024 | 0.0237 | 0 |
| 9 | 100000000 | 111111111 | 256.00 | 511.00 | 1.0020 | 2.0000 | 99.2180 | 0.0100 | 0.0084 | 0.0841 | 0 |
| 10 | 100010101 | 100000011 | 277.00 | 259.00 | 1.0841 | 1.0137 | 2.6211 | 0.2762 | 0.2329 | 2.3286 | 2 |

Sum of f(x)= 1.1859 Average of f(x)=0.11856

**TABLE 3.3**
**New Population for Generation 2**

| $x_1$ (Old) | $x_2$ (Old) | $x_1$ (New) | $x_2$ (New) |
|---|---|---|---|
| 101010101 | 111111100 | 101010100 | 111110111 |
| 100100100 | 100000111 | 100100101 | 100001100 |
| 101010110 | 111100011 | 101010101 | 111100011 |
| 100010101 | 100000011 | 100010110 | 100000011 |
| 101010110 | 111100011 | 101010101 | 111101100 |
| 101010101 | 111111100 | 101010110 | 111110011 |
| 101010110 | 111100011 | 101010101 | 111100011 |
| 100010101 | 100000011 | 100010110 | 100000011 |
| 101010110 | 111100011 | 101010010 | 111100001 |
| 100001111 | 110000010 | 100001110 | 110000011 |

Table 3.4 shows the best fitness value achieved in each generation. In Table 3.4, column (1) shows the generation number. Columns (2) and (3) show the best value of $x_1$ and $x_2$ in the corresponding generation. Columns (4) and (5) show the sum and average of function value in generation. Column (6) shows the best function value in the corresponding generation.

**TABLE 3.4**
**Best fitness value of function in each generation**
**(Bit size 9, Population size 10)**

| Generation (1) | $x_1$ (2) | $x_2$ (3) | Sum of f(x) (4) | Average of f(x) (5) | Best fitness Value (6) |
|---|---|---|---|---|---|
| 1 | 1.3386 | 1.8904 | 1.1859 | 0.1186 | 0.4788 |
| 2 | 1.3346 | 1.8904 | 2.47 | 0.275109 | 0.4341 |
| 3 | 1.0254 | 1.0098 | 3.5917 | 0.449 | 0.1749 |
| 4 | 1.0254 | 1.0137 | 3.7287 | 0.3729 | 0.8743 |
| 5 | 1.0254 | 1.0098 | 4.1778 | 0.4642 | 0.8511 |
| 6 | 1.0724 | 1.1311 | 5.7357 | 0.6373 | 0.9605 |
| 7 | 1.0254 | 1.0881 | 4.8058 | 0.534 | 0.8817 |
| 8 | 1.0098 | 1.0254 | 6.1967 | 0.6885 | 0.9966 |
| 9 | 1.0254 | 1.0685 | 5.806 | 0.8294 | 0.9714 |
| 10 | 1.0098 | 1.0098 | 5.6355 | 0.8051 | 0.9902 |
| 11 | 1.0098 | 1.0254 | 4.8067 | 0.9613 | 0.9966 |
| 12 | 1.0059 | 1.0215 | 4.4796 | 0.8959 | 0.9905 |
| 13 | 1.0059 | 1.0215 | 4.6221 | 0.9244 | 0.9905 |
| 14 | 1.0098 | 1.002 | 4.90056 | 0.9811 | 0.9902 |
| 15 | 1.002 | 1.002 | 4.915 | 0.983 | 0.9996 |
| 16 | 1.002 | 1.002 | 4.9588 | 0.9918 | 0.9996 |
| 17 | 1.002 | 1.002 | 4.9588 | 0.9918 | 0.9996 |
| 18 | 1.002 | 1.0059 | 4.9679 | 0.9936 | 0.9996 |
| 19 | 1.002 | 1.002 | 4.998 | 0.9996 | 0.9996 |

The population ( $x_1$, $x_2$) at each generation is shown in Fig 3.2 to 3.20 for combination 1 (bit size 9 and population size 10).

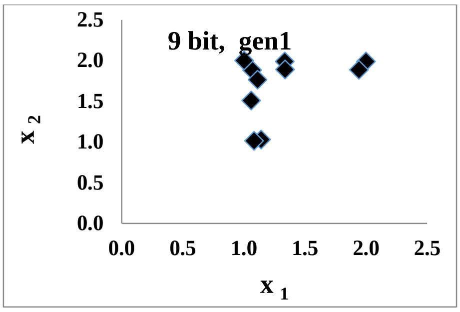



**Fig. 3.2** Population at Generation 1

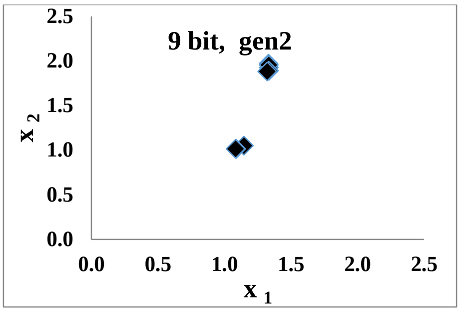



**Fig. 3.3** Population at Generation 2

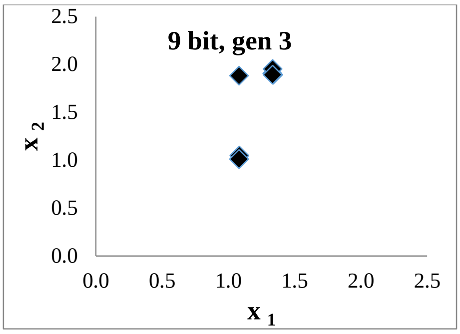



**Fig. 3.4** Population at Generation 3

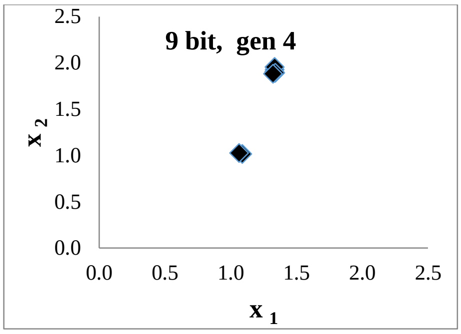



**Fig. 3.5** Population at Generation 4

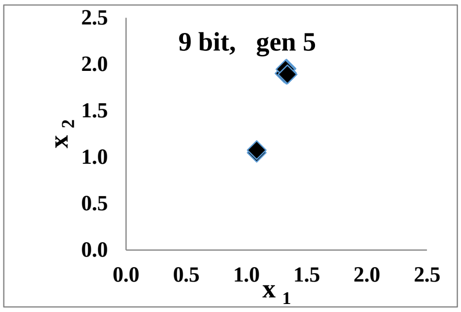



**Fig. 3.6** Population at Generation 5

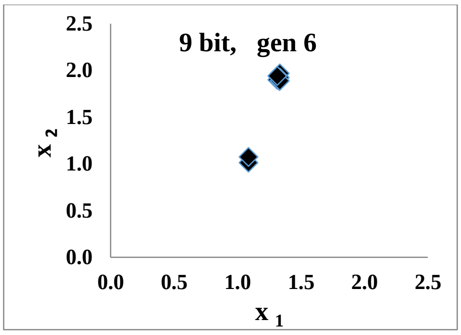



**Fig. 3.7** Population at Generation 6

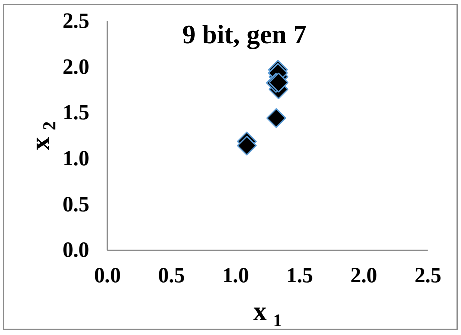



**Fig. 3.8** Population at Generation 7

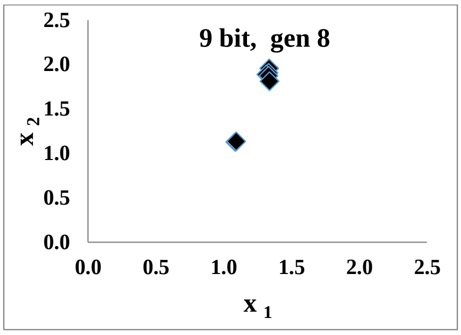



**Fig. 3.9** Population at Generation 8

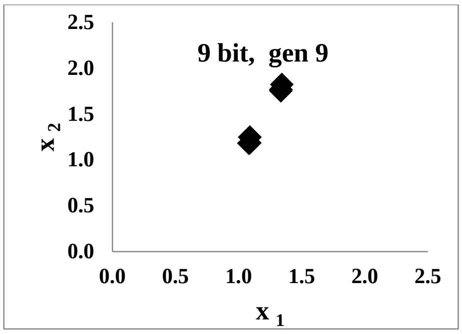



**Fig. 3.10** Population at Generation 9

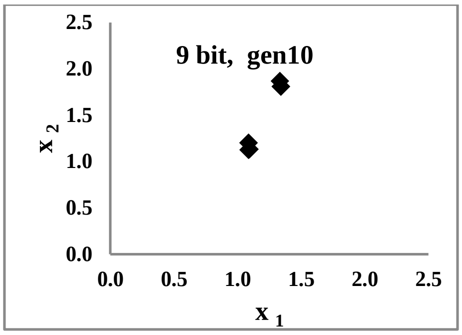



**Fig. 3.11** Population at Generation 10

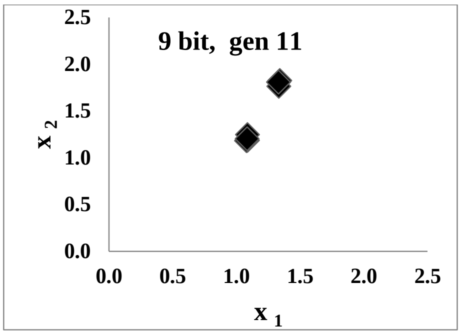



**Fig. 3.12** Population at Generation 11

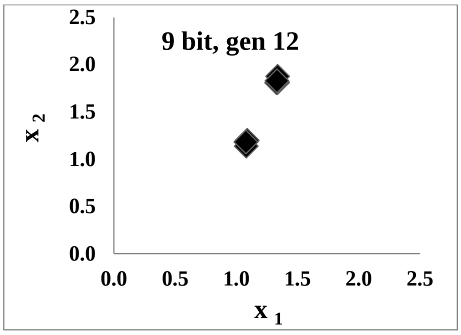



**Fig. 3.13** Population at Generation 12

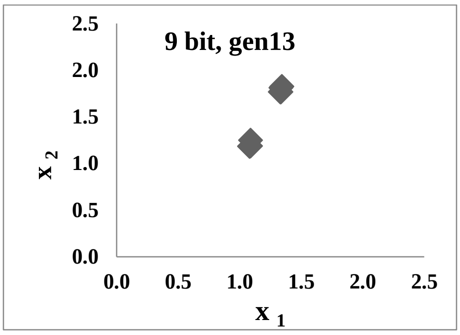



**Fig. 3.14** Population at Generation 13

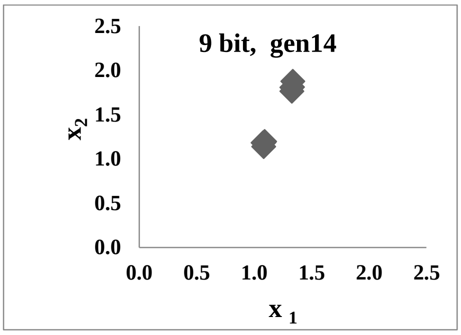



**Fig. 3.15** Population at Generation 14

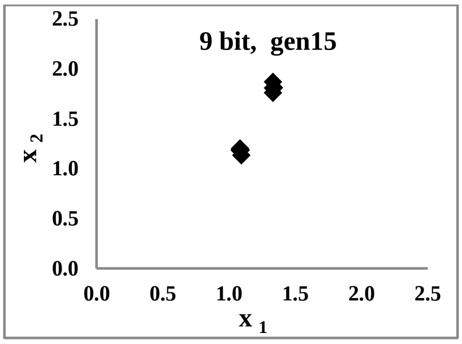



**Fig. 3.16** Population at Generation 15

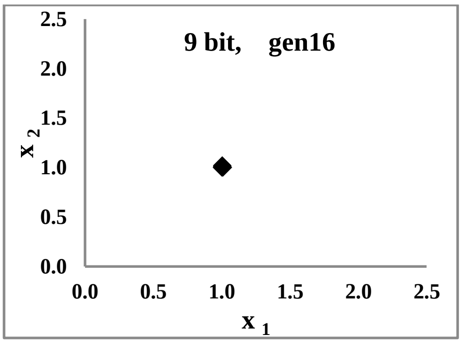



**Fig. 3.17** Population at Generation 16

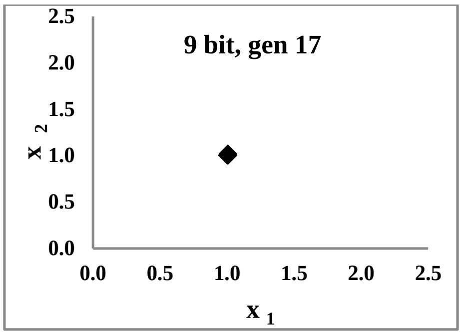



**Fig. 3.18** Population at Generation 17

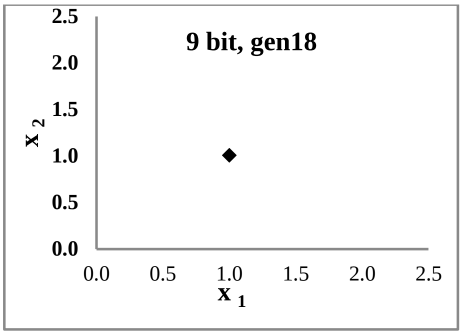



**Fig. 3.19** Population at Generation 18

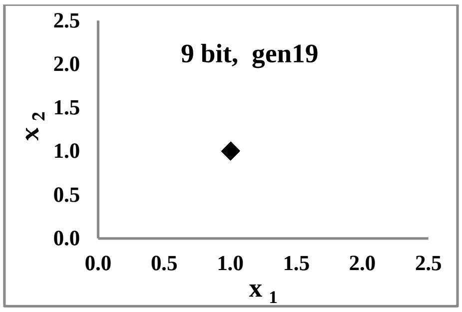



**Fig. 3. 20** Population at Generation 19

## Bit size 8 and Population size 10

The Rosenbrock function has been solved manually for bit size 8 and population size 10. The results are summarised in Table 3.5.

**TABLE 3.5**
**Best Fitness Value of Function in each Generation**
**(Bit Size 8, Population Size 10)**

| Generation (1) | $x_1$ (2) | $x_2$ (3) | Sum of f(x) (4) | Average of f(x) (5) | Best fitness Value (6) |
|---|---|---|---|---|---|
| 1 | 1.0039 | 1.0118 | 0.1602 | 0.0801 | 0.9985 |
| 2 | 1.0039 | 1.0118 | 1.5468 | 0.7734 | 0.9985 |
| 3 | 1.0039 | 1.0039 | 1.4901 | 0.7451 | 0.9984 |
| 4 | 1.0118 | 1.0118 | 1.2747 | 0.6374 | 0.9859 |
| 5 | 1.0039 | 1.0039 | 1.5783 | 0.7892 | 0.9984 |
| 6 | 1.0039 | 1.0039 | 1.2392 | 0.6196 | 0.9984 |
| 7 | 1.0039 | 1.0039 | 1.8901 | 0.9451 | 0.9984 |
| 8 | 1.0039 | 1.0118 | 1.4946 | 0.7473 | 0.9985 |
| 9 | 1.0039 | 1.0039 | 1.9969 | 0.9984 | 0.9984 |
| 10 | 1.0039 | 1.0039 | 1.9968 | 0.9984 | 0.9984 |
| 11 | 1.0039 | 1.0039 | 1.9968 | 0.9984 | 0.9984 |

Figures (3.21) to (3.31) show the population ($x_1$, $x_2$) at each generation for 8-bit with population size 10.

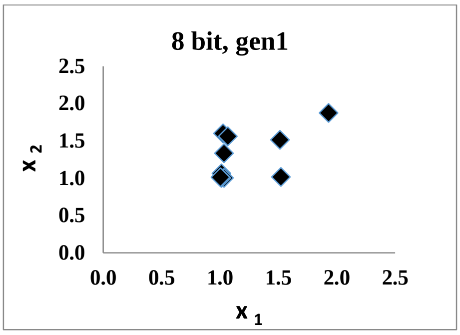



**Fig. 3.21** Population at Generation 1

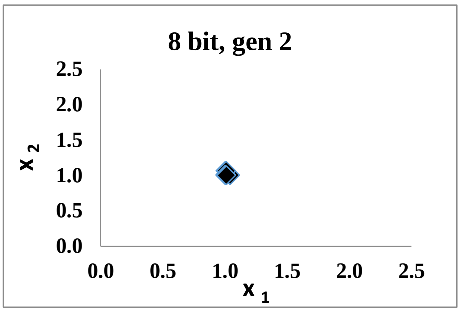



**Fig. 3.22** Population at Generation 2

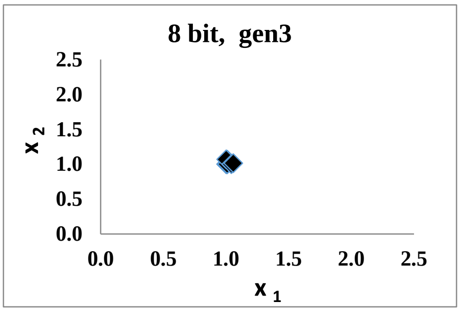



**Fig. 3.23** Population at Generation 3

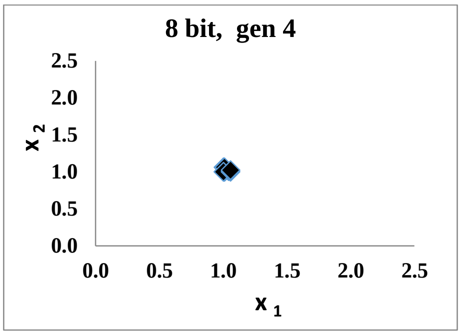



**Fig. 3.24** Population at Generation 4

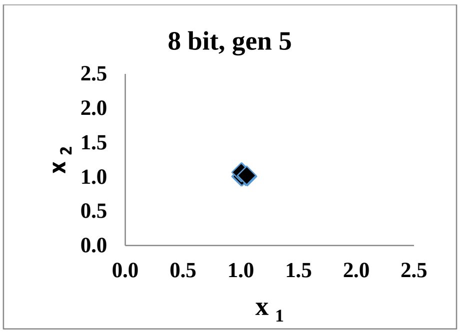



**Fig. 3.25** Population at Generation 5

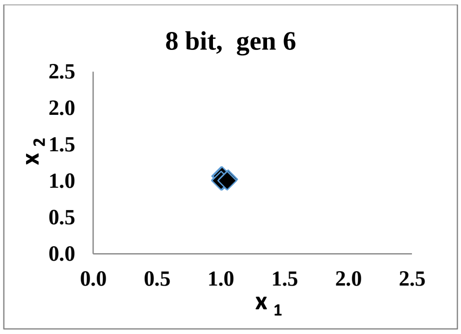



**Fig. 3.26** Population at Generation 6

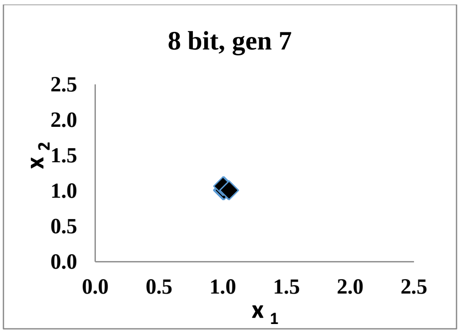



**Fig. 3.27** Population at Generation 7

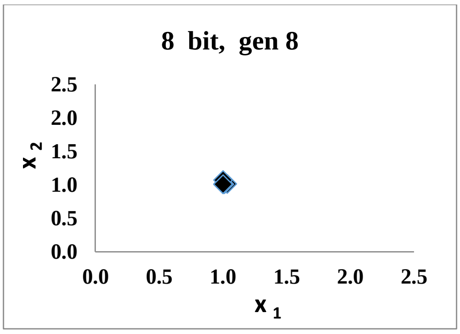



**Fig. 3.28** Population at Generation 8

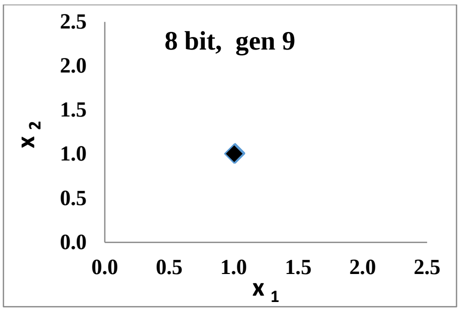



**Fig. 3.29** Population at Generation 9

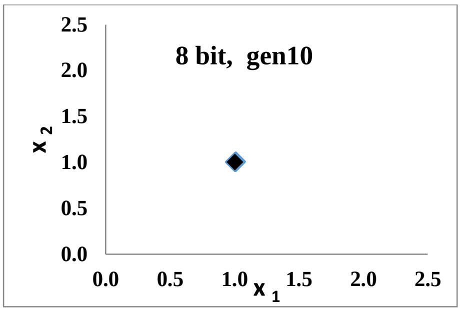



**Fig. 3.30** Population at Generation 10

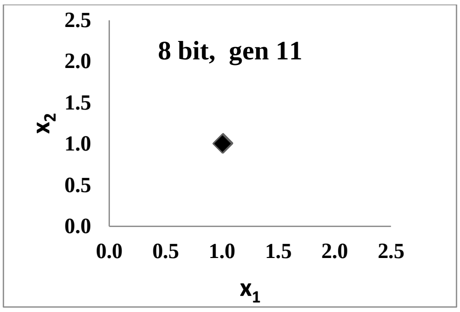



**Fig. 3.31** Population at Generation 11

## Bit size 5, Population size 6

The Rosenbrock function has been solved manually for bit size 5 and population size 6.

The results are summarised in Table 3.6.

**TABLE 3.6**
**Best Fitness Value of Function in each Generation**
**(Bit Size 5, Population Size 6)**

| Generation (1) | $x_1$ (2) | $x_2$ (3) | Sum of f(x) (4) | Average of f(x) (5) | Best fitness Value (6) |
|---|---|---|---|---|---|
| 1 | 1.0323 | 1.0968 | 1.9501 | 0.325 | 0.9103 |
| 2 | 1.1613 | 1.2258 | 0.8624 | 0.1437 | 0.3947 |
| 3 | 1.0323 | 1.0968 | 2.6536 | 0.4423 | 0.9103 |
| 4 | 1.0323 | 1.0968 | 1.9729 | 0.3288 | 0.9103 |
| 5 | 1.0323 | 1.0968 | 1.9729 | 0.3288 | 0.9103 |
| 6 | 1.0323 | 1.0968 | 2.8482 | 0.4747 | 0.9103 |
| 7 | 1.0323 | 1.0968 | 2.4895 | 0.4149 | 0.9103 |
| 8 | 1.1613 | 1.0323 | 3.3098 | 0.5516 | 0.9103 |
| 9 | 1.1613 | 1.0323 | 3.3098 | 0.5516 | 0.9103 |
| 10 | 1.1613 | 1.0323 | 4.1267 | 0.6878 | 0.9103 |
| 11 | 1.1613 | 1.0323 | 5.462 | 0.4103 | 0.9103 |

Figures (3.32) to (3.42) show the population $(x_1, x_2)$ at each generation for bit size 5 and population size 6.

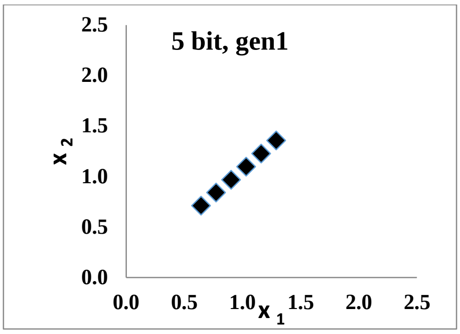



**Fig. 3.32** Population at Generation 1

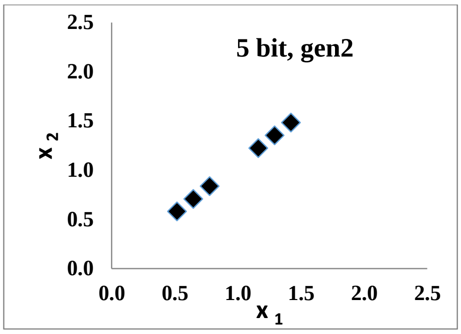



**Fig. 3.33** Population at Generation 2

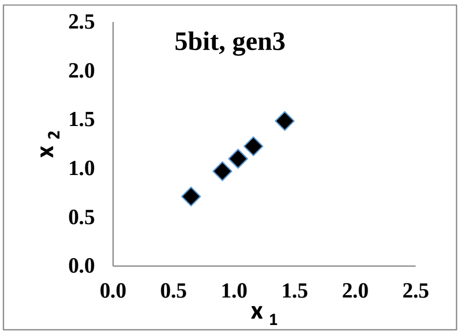



**Fig. 3.34** Population at Generation 3

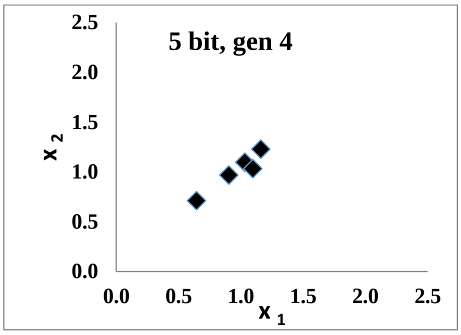



**Fig. 3.35** Population at Generation 4

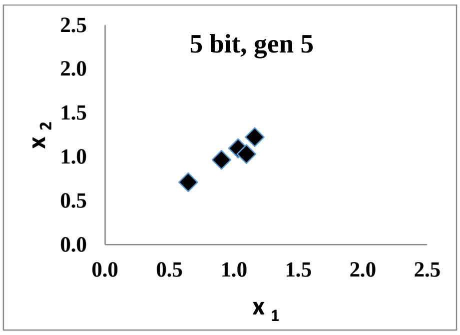



**Fig. 3.36** Population at Generation 5

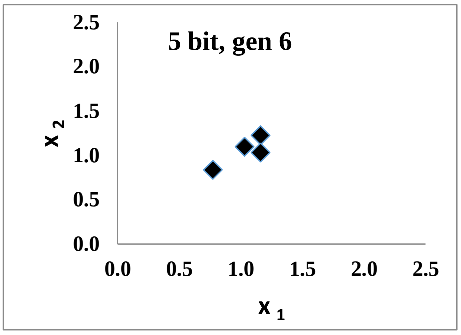



**Fig. 3.37** Population at Generation 6

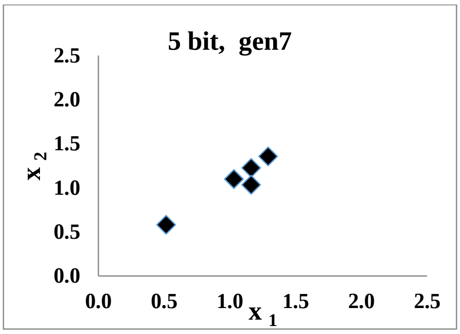



**Fig. 3.38** Population at Generation 7

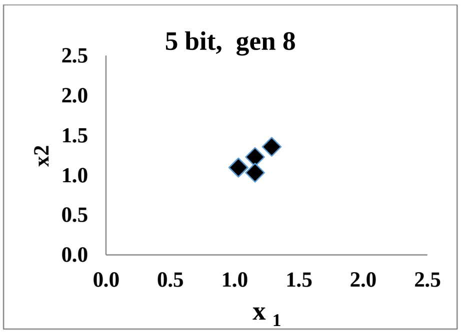



**Fig. 3.39** Population at Generation 8

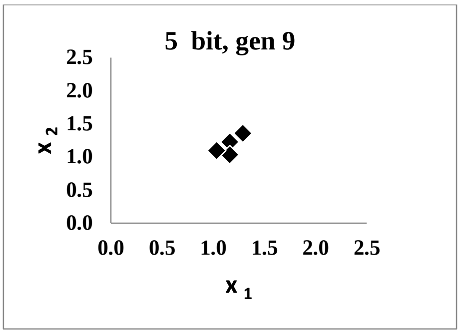



**Fig. 3.40** Population at Generation 9

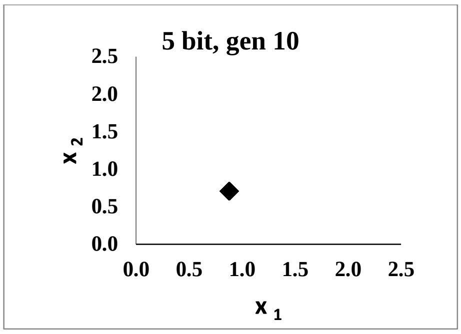



**Fig. 3.41** Population at Generation 10

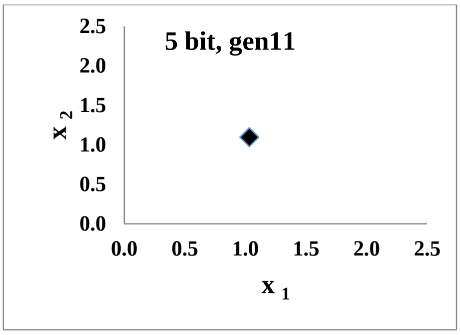



**Fig. 3.42** Population at Generation 11

## Bit size 3 and Population size 7

The Rosenbrock function has been solved manually for bit size 3 and population size 7.

The results are summarised in Table 3.7.

**TABLE 3.7**
**Best Fitness Value of Function in each Generation**
**(Bit size 3, Population size 7)**

| Generation (1) | $x_1$ (2) | $x_2$ (3) | Sum of f(x) (4) | Average of f(x) (5) | Best fitness Value (6) |
|---|---|---|---|---|---|
| 1 | 1.4286 | 1.7143 | 0.1997 | 0.0285 | 0.0844 |
| 2 | 1.4286 | 1.7143 | 0.2631 | 0.0376 | 0.0844 |
| 3 | 1.4286 | 1.7143 | 0.4608 | 0.0658 | 0.0844 |
| 4 | 1.4286 | 1.7143 | 0.1780 | 0.0254 | 0.0844 |
| 5 | 1.4286 | 1.7143 | 0.3942 | 0.0788 | 0.0844 |
| 6 | 1.4286 | 1.7143 | 0.3942 | 0.0844 | 0.0844 |
| 7 | 1.4286 | 1.7143 | 0.3377 | 0.0844 | 0.0844 |

Fig. (3.43) to (3.49) show the population ($x_1$, $x_2$) at each generation for bit size 3 and population size 7.

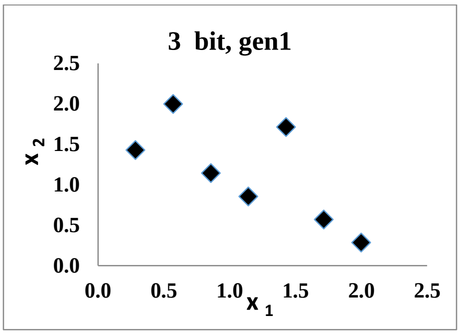



**Fig. 3.43** Population at Generation 1

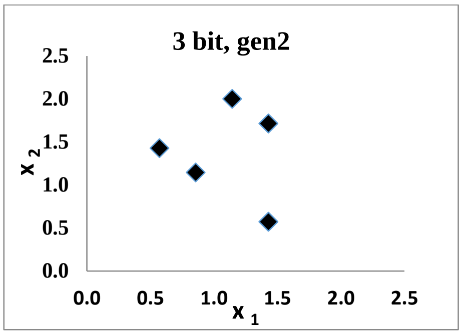



**Fig. 3.44** Population at Generation 2

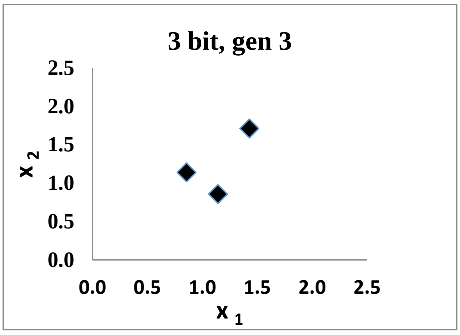



**Fig. 3.45** Population at Generation 3

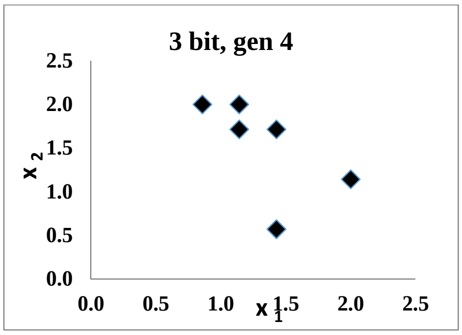



**Fig. 3.46** Population at Generation 4

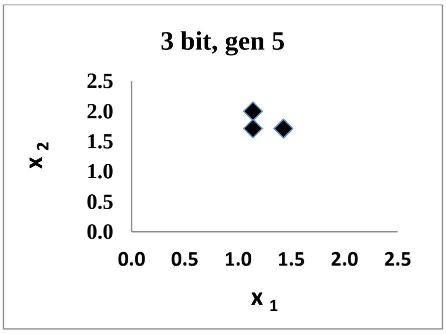



**Fig. 3.47** Population at Generation 5

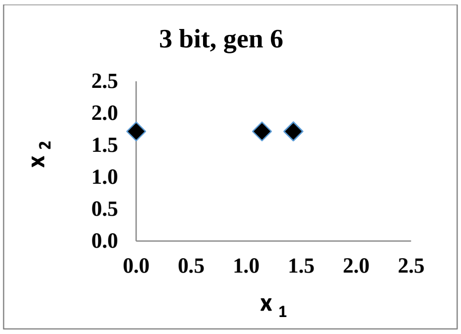



**Fig. 3.48** Population at Generation 6

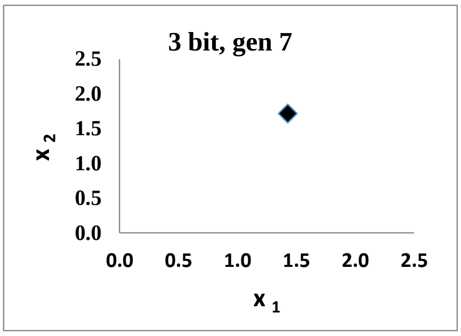



**Fig. 3.49** Population at Generation 7

Table 3.8 summarizes the results of all the combinations of bit size and population size as shown in Table 3.4 to 3.7.

**TABLE 3.8**
**Number of Generations for various combinations**
**(Bit size and Population size)**

| S. No. (1) | Bit Size (2) | Population Size (3) | Probability crossover (4) | Probability mutation (5) | $x_1$ (6) | $x_2$ (7) | No. of Generations (8) |
|---|---|---|---|---|---|---|---|
| 1 | 9 | 10 | 0.8 | 0.01 | 1.002 | 1.002 | 19 |
| 2 | 8 | 10 | 0.8 | 0.1 | 1.0039 | 1.0039 | 11 |
| 3 | 5 | 6 | 0.8 | 0.01 | 1.1613 | 1.0323 | 9 |
| 4 | 3 | 7 | 0.8 | 0.01 | 1.4286 | 1.7143 | 7 |

In all the combinations, probability of crossover was fixed to o.8 whereas probability of mutation was varied from 0.01 to 0.1. Best fitness value achieved in each generation for all the combinations is shown in Table 3.4 to 3.7. The population $(x_1, x_2)$ at each generation for all the combinations has also been plotted. Fig. 3.2 to 3.20 represent population at each generation for bit size 9 and population size 10. Fig. 3.21 to 3.31 represent population at each generation for bit size 8 and population size 10. Similarly, Fig. 3.32 to 3.42 represent population at each generation for bit size 5 and population size 6 and Fig. 3.43 to 3.49 represent population at each generation for bit size 3 and population size 7 combinations respectively.

For all combinations of bit size and population size, initial population searches the larger area defined by function. As generation increases, population search area decreases. One-point crossover is applied for combination 1 of Table 3.1. Two-point crossover is applied to combination 2 (bit size 8 and population size 10). In this case, exploration area reduces in second generation itself. Population converged to single point in eleven generations only whereas it took nineteen generations for first combination where one-point crossover operator was used. For smaller bit size and population size, population not converge accurately, as shown in Fig. 3.32 to 3.42 for combination 3 (bit size 5, population size 6) and Fig. 3.43 to 3.49 for combination 4 (bit size 3, population size 7) respectively.

As the population converges, the average fitness will approach that of the best individual. When bit string size increases, accuracy of the optimal point improves. It is observed from Table 3.8 that the best results are obtained for combination 1 (bit size 9 and population size 10) i.e. the function converged very near to the optimal value. For smaller bit sizes (Combination 3 and 4), the function did not converge accurately.

### 3.4.2 Effect of Population size

Effect of population size on convergence has also been studied with the help of GA Toolbox of MATLAB. Population size is varied from 10 to 120 in steps of 10 and 120 to 200 in steps of 20 respectively. Other parameters were fixed to the values as given below:

Crossover rate = 0.98 Generations = 15,000 Fitness limit = 1* E-7

Stall Generations = 15,000 Function Tolerance = 1* E-7 Time Limit = Infinite

Nonlinear constraint = 1* E-7

Results of Rosenbrock function using MATLAB Toolbox are shown in Table 3.9. Column (2) of Table 3.9 represents the variations in population size from 10 to 120 in steps of 10 and 120 to 200 in steps of 20 respectively. Column (3) shows the number of generations required to optimize the function. Columns (4) and (5) show the values of variables $x_1$ and $x_2$ to optimize the function. Column (6) shows the function values for corresponding population size.

**TABLE 3.9**
**Number of Generations for Various Population Sizes**

| S. No. (1) | Population size (2) | Generation (3) | $x_1$ (4) | $x_2$ (5) | fvalue (6) |
|---|---|---|---|---|---|
| 1 | 10 | 2309 | 1 | 0.999 | 9.94E-08 |
| 2 | 20 | 2052 | 1 | 1 | 7.97E-08 |
| 3 | 30 | 995 | 1 | 1 | 6.38E-08 |
| 4 | 40 | 554 | 1 | 0.99 | 9.63E-08 |
| 5 | 50 | 665 | 1 | 0.999 | 9.25E-08 |
| 6 | 60 | 358 | 1 | 1 | 8.52E-08 |
| 7 | 70 | 166 | 1 | 0.999 | 8.10E-08 |
| 8 | 80 | 195 | 1 | 0.999 | 8.52E-08 |
| 9 | 90 | 189 | 1 | 1 | 4.90E-08 |
| 10 | 100 | 1371 | 1 | 1 | 7.82E-08 |
| 11 | 110 | 183 | 1 | 1 | 5.00E-08 |
| 12 | 120 | 158 | 1 | 1 | 2.73E-08 |
| 13 | 140 | 139 | 1 | 1 | 7.29E-08 |
| 14 | 160 | 76 | 1 | 1 | 7.19E-08 |
| 15 | 180 | 44 | 1 | 1 | 7.24E-08 |
| 16 | 200 | 63 | 1 | 1 | 7.28E-08 |

An attempt has been made to find the relation between population size and number of generations required to optimize the function, following models have been tried for curve fitting of data of Table 3.9.

i. $Y = a*e^{(-kt)}$ (3.5)

ii. $Y=a*e^{(-kt)}+ c$ (3.6)

iii. $Y = c + (c'/x)$ (3.7)

First model has given the best result and has been discussed here.

Constants ‘a’ and ‘k’ have been calculated using Mathematica Toolbox.

a = 3179.2, k = 0.0313985

$Y = 3179.2 *e^{(-0.0313985t)}$ (3.8)

Fig. 3.50 shows the graphical representation of Model (i) represented by equation (3.8).

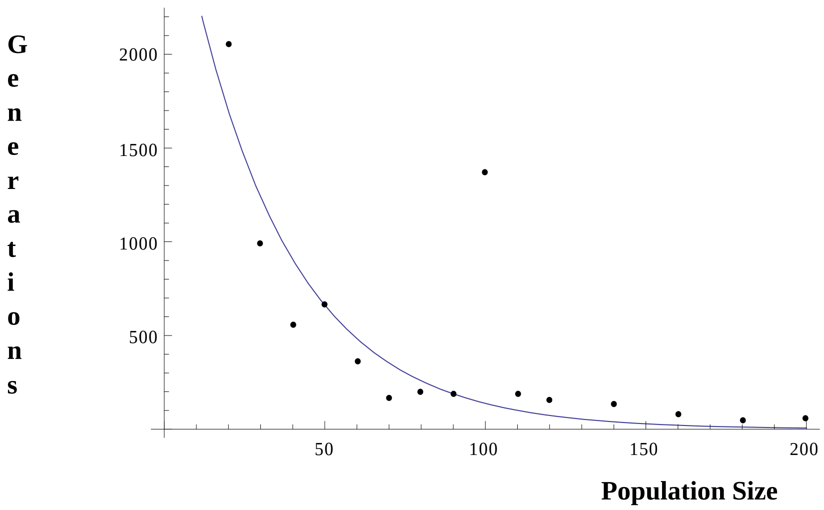



**Fig. 3.50** Curve fitting of data in Table 3.9

## 3.5 BASIC PARTICLE SWARM OPTIMIZATION (BPSO)

Eberhart and Kennedy [10] introduced the Basic Particle Swarm Optimization (BPSO) in 1995. BPSO is an intelligent, gradient free, population based self-adaptive, stochastic optimization technique. It emulates the flocking behaviour of birds to solve optimization problem. BPSO is used to find solutions of difficult, nonlinear, numeric maximization and minimization problems. It is used in optimal design of electrical networks, design of aircraft to find optimal trajectories of space vehicles, optimal production, planning, controlling and scheduling.

Basic Particle Swarm Optimization (BPSO) has simple and easy concept to implement and is computationally efficient. It is a mathematical modelling and simulation of food searching activities of a swarm of birds (particles). The optimization process uses a number of particles constituting a swarm that moves around a pre-defined search space looking for the best solution. Each particle is treated as a point in the N dimensional space in which the particle adjusts its "flying" according to its own flying experience as well as the flying experience of other neighboring particles of the swarm. Each particle keeps track of its coordinates in the pre-defined space, which are associated with the best solution (fitness) that it has achieved so far. This value is called $x_{pbest}$. Another best value $x_{gbest}$ that is tracked by the PSO is the best value obtained so far by any particle in the whole swarm. The concept consists of changing the velocity of each particle toward its pbest and the gbest position at the end of each iteration. Each particle tries to modify its current position and velocity according to the distance between its current position and $x_{pbest}$ and the distance between its current position and $x_{gbest}$.

In an N-dimensional search space, position vector and velocity vector of particle j is represented by vectors $x_j = (x_{1j}, x_{2j}, \ldots\ldots x_{ij} \ldots x_{Nj})$ and $v_j = (v_{1j}\ v_{2j}, \ldots v_{ij} \ldots v_{Nj})$ respectively. Let $x_{pbestij}$ be the personal best position of particle j for $i^{th}$ variable. $x_{gbesti}$ be the global best positions from all the particles for $i^{th}$ variable. The modified velocity and position of each particle can be calculated using current velocity and distance from $x_{pbest}$ and $x_{gbest}$ as follows:

$$v_{ij}^{k+1} = W * v_{ij}^{k} + C_p r_p(x_{pbestij}^{k} - x_{ij}^{k}) + C_g r_g(x_{gbesti}^{k} - x_{ij}^{k}) \quad i = 1,2,\ldots..N;\ j=1,2\ldots..P \quad (3.9)$$

Where i represents the $i^{th}$ dimension, j represents the $j^{th}$ particle and k represents the $k^{th}$ iteration.

Position of each particle is updated using equation (3.10)

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \qquad i = 1, 2\ldots.N;\ j=1, 2\ldots..P \qquad (3.10)$$

This is explained in Fig. 3.51.

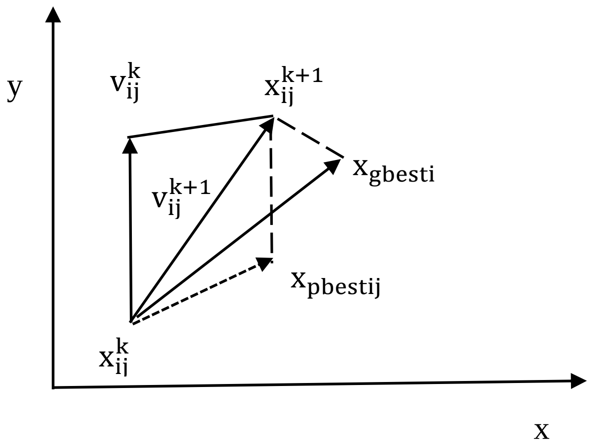



**Fig. 3.51** Updated Position of a Particle

Linearly decreasing inertia weight is calculated using equation (3.11)

$$W = W_{max} - ((W_{max} - W_{min}) * k/IT_{max}) \quad (3.11)$$

$\mathbf{x_{pbest}}$ and $\mathbf{x_{gbest}}$ can be defined as follows:

**Personal best ($\mathbf{x_{pbest}}$):** The personal best position associated with $j^{th}$ particle is the best position that the particle has visited yielding the highest fitness value for that particle.

**Global best ($\mathbf{x_{gbest}}$):** The best position that any particle in the swarm has visited yielding the highest fitness value.

One of the stopping criteria is maximum number of iterations $IT_{max}$, which also influences the performance of BPSO. However, this stopping criterion does not indicate the achievement of optimum, but it is used to prevent the program to run indefinitely. Second stopping criteria is the difference between the previous and the current fitness, which is calculated and checked against the tolerance value for all the particles. If it lies within the tolerance, iteration stops and global best value is the optimal solution. If the program stops because of satisfaction of this criterion, would confirm the achievement of optimum.

### 3.5.1 Parameters of Basic Particle Swarm Optimization

Various parameters of BPSO are:

**Swarm:** It is an apparently disorganized population of moving particles that tend to cluster together towards a common optimum while each particle seems to be moving in a random direction.

**Particle (x):** It is a candidate solution, in an N-dimensional space. At time t, the $j^{th}$ particle $x_j(t)$ can be described as $x_j(t) = [x_{1j}(t), x_{2j}(t), \ldots, x_{ij}(t) \ldots x_{Nj}(t)]$, where $x_{ij}(t)$ is the position of the $j^{th}$ particle with respect to the $i^{th}$ dimension, at the time 't'.

**Population size (q):** Population size is the number of particles in a swarm. If the number of particles in the swarm is less than a critical value, the algorithm does not converge.

**Velocity (v):** It is the velocity of a moving particle, can be represented at time t, for $j^{th}$ particle as $v_j(t) = [\, v_{1j}(t), v_{2j}(t), \ldots \ldots . v_{ij}(t) \ldots \ldots \ldots\ v_{Nj}(t)]$.

**Inertia weight factor (W):** Inertia weight factor determines the weightage of a particle's previous velocity in the velocity update equation. Higher the value of this factor, greater the influence of the previous velocity. Therefore, this parameter determines the "inertia" of a particle, hence the name inertia weight factor. Inertia weight forces the particle to move in the same direction. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search.

**Cognitive learning acceleration factor($\mathbf{C_p}$):** The parameter, which appears as a constant coefficient in the second term of the velocity update equation, is represented by $C_p$, called as personal acceleration constant. An increased value of $C_p$, improves the local search capability of the particles and a reduction in $C_p$ hampers the local searching by the population.

**Social learning acceleration factor ($C_g$):** Social learning acceleration factor denoted by $C_g$, is used in the third term of the velocity update equation. Higher value of $C_g$ enhances the global search ability of the population. In the initial stage of optimization $C_g$ should be more than $C_p$ for better exploration and in final stage of optimization, $C_p$ should be more than $C_g$ for better convergence.

**Random Factors ($r_p$, $r_g$):** In equation (3.9), random factors are associated with the cognitive as well as social learning terms. These are useful for better exploration. These are important only when problem to be solved is new and problem solver does not have any idea about the solution of the problem.

### 3.5.2 Steps to Optimize a Function

Following are the steps to optimize the function using Basic Particle Swarm Optimization:

1. Initialize parameters of PSO i.e. $C_p$, $C_g$, $r_p, r_g$, $IT_{max}$, q, W, Ɛ.
2. Set iteration count k=0.
3. Initialize the position and velocity of particles within the range of variables.
4. Calculate the fitness function value for each particle.
5. Determine $x_{pbest}$ and $x_{gbest}$ corresponding to zeroth iteration. $x_{pbest}$ is the initial value of position assigned to the particles. $x_{gbest}$ is the position of particle corresponding to minimum function value.
6. Update the velocities and positions of particles.
7. Increase iteration count by one, i.e. k=k+1.
8. Calculate the function value for updated position of particles.

9. Update $\mathbf{x_{pbest}}$ and $\mathbf{x_{gbest}}$. Compare the fitness function values of current and previous iteration for each particle. $x_{pbest}$ is the position corresponding to minimum fitness function value for each particle i.e. if $F(x_{ij}^{k})$ is less than $F(x_{pbestij}^{k})$, then assign the value of $x_{pbestij}^{k}$ as $x_{ij}^{k}$ (do it for all the particles). $x_{gbest}$ is the position of particle corresponding to minimum function value attained so far by swarm. Determine the best value of $x_{pbestij}$ considering its fitness value. If F(best of $x_{pbestij}$) is less then $F(x_{gbesti})$, then assign the value of best of $x_{pbestij}$ to the $x_{gbesti}$.
10. Determine fitness function value for each particle.
11. Compare the fitness function value of two consecutive iterations. If it is less than Ɛ ($10^{-6}$) for all particles go to 12, else go to (7).
12. Display $x_{gbesti}$ as the optimal solution and the fitness function value corresponding to $x_{gbesti}$ .

## 3.6 APPLICATION OF BPSO ON ROSENBROCK FUNCTION

The Optimization of Rosenbrock function has been done manually as well as using MATLAB Programming to study the effect of parameters of BPSO on convergence of function. The parameters of BPSO have been fixed to the values as given below:

| | | |
|---|---|---|
| q = 10; | $r_p$ = 0.4; | $r_g$ = 0.5; |
| $C_p$ = 2; | $C_g$ = 2; | $IT_{max}$ = 10; |
| $W_{max}$ = 0.9; | $W_{min}$ = 0.4; | Ɛ = $1 * 10^{-6}$ |

Linearly Decreasing Inertia weight "W" as shown by equation (3.11) has been considered. Initially velocity and positions of 10 particles are generated randomly. To understand the working of Basic Particle Swarm Optimization (BPSO), it is implemented manually on benchmark Rosenbrock function using the steps shown in section 3.5.2.

## 3.7 COMPUTATIONAL RESULTS AND DISCUSSION

Rosenbrock function has been solved manually using BPSO. Flow Chart for Basic Particle Swarm Optimization is shown in Fig 3.52. Table 3.10 shows velocity, position of particles and function values at zeroth iteration. Column (1) shows the particle number which is generated randomly at zeroth iteration. Columns (2) and (3) represent initial velocities of particles generated randomly. Columns (4) and (5) show the initial positions of the corresponding particle at zeroth iteration. Personal best values for each particle will be their own position in the zeroth iteration. Global best value of position will be the position of that particle, corresponding to which function value is minimum.

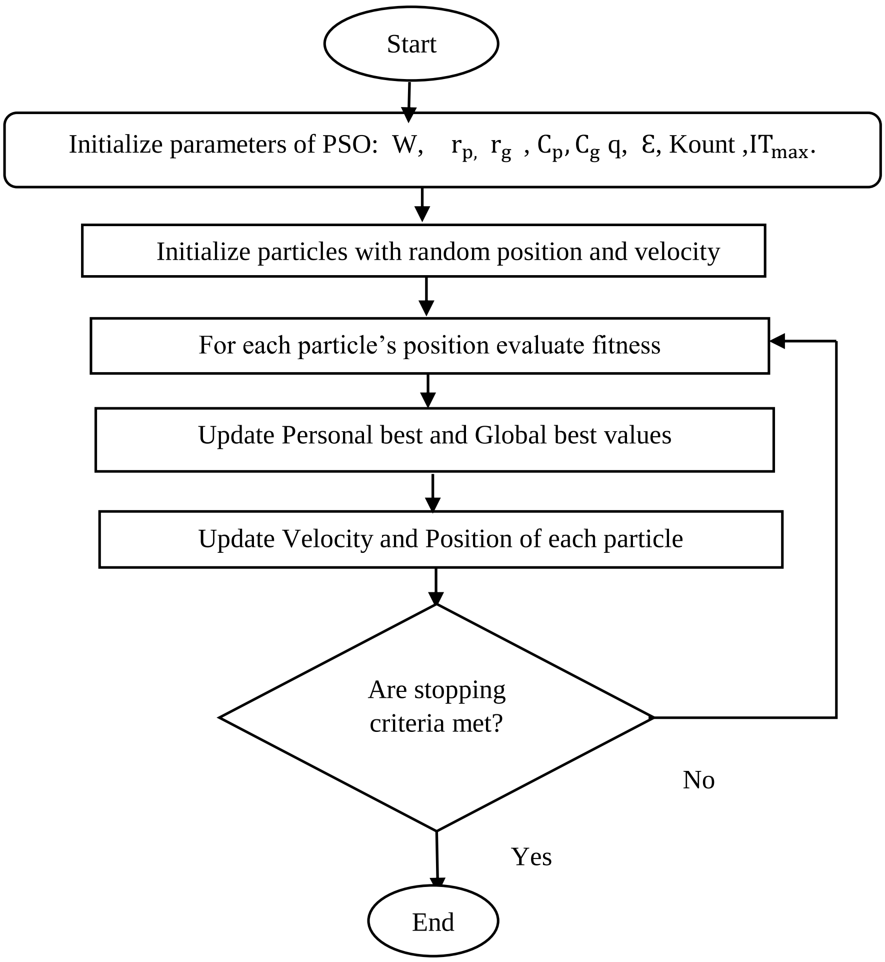



**Fig. 3.52** Flow chart for basic PSO

In Table 3.10 global best value is highlighted in columns 4 and 5 of row 2. Column 6 shows the function value calculated using equation (3.1) for each particle.

**TABLE 3.10**
**Velocities and Positions at zeroth Iteration**

| Particles No.(1) | $v_1$ (2) | $v_2$ (3) | $x_1$ (4) | $x_2$ (5) | function value (6) |
|---|---|---|---|---|---|
| 1 | 0.3111 | 0.5949 | 1.2886 | 0.4155 | 155.08339 |
| **2** | 0.9234 | 0.2622 | **0.7572** | **0.6025** | **0.14391** |
| 3 | 0.4302 | 0.6028 | 1.6232 | 0.9418 | 287.00591 |
| 4 | 0.1848 | 0.7112 | 1.0657 | 0.461 | 45.52855 |
| 5 | 0.9049 | 0.2217 | 0.7015 | 1.6886 | 143.24978 |
| 6 | 0.9797 | 0.1174 | 1.878 | 0.3895 | 985.08872 |
| 7 | 0.4389 | 0.2967 | 1.7519 | 0.4518 | 685.61934 |
| 8 | 0.1111 | 0.3188 | 1.1003 | 0.3414 | 75.57137 |
| 9 | 0.2581 | 0.4242 | 1.245 | 0.4553 | 119.90231 |
| 10 | 0.4087 | 0.5079 | 1.1741 | 0.8714 | 25.74644 |

The position of particles at zeroth iteration is graphically represented in Fig. 3.53.

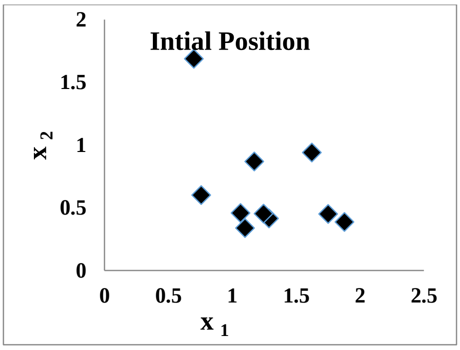



**Fig.3.53** Position of particles at zeroth Iteration

It is observed that Particles are spread over large search area. Now the first iteration starts. In this the velocity of particles and position of particles are updated using equation (3.9) and (3.10) respectively. These updated velocities of Particles in first iteration are shown in columns (2) and (3) respectively and updated positions of particles of first iteration are shown in columns (4) and (5) respectively of Table 3.11. For updated position of particles, function values have been calculated using equation (3.1) which are shown in column (6) of Table 3.11.

**TABLE 3.11**
**Result of Rosenbrock function at First Iteration**

| Particles No. (1) | $v_1$ (2) | $v_2$ (3) | $x_1$ (4) | $x_2$ (5) | function value (6) |
|---|---|---|---|---|---|
| 1 | -0.2669 | 0.6926 | 1.0216 | 1.1065 | 0.4155 |
| 2 | 0.7848 | 0.2228 | 1.5420 | 0.8253 | 241.3727 |
| 3 | 0.50033 | 0.1730 | 1.1228 | 1.1148 | 2.1454 |
| 4 | -0.1514 | 0.7460 | 0.9142 | 1.2070 | 13.7797 |
| 5 | 0.8248 | -0.8976 | 1.5263 | 0.7909 | 237.0814 |
| 6 | -0.2880 | 0.3127 | 1.5899 | 0.7022 | 333.6423 |
| 7 | -0.6216 | 0.4028 | 1.1302 | 0.8546 | 17.8932 |
| 8 | -0.2486 | 0.5320 | 0.8516 | 0.8738 | 2.2182 |
| 9 | -0.2684 | 0.5077 | 0.9765 | 0.9630 | 0.0092 |
| 10 | -0.0695 | 0.1628 | 1.1045 | 1.0342 | 3.4673 |

The position of particles for First iteration is shown in Fig.3.54.

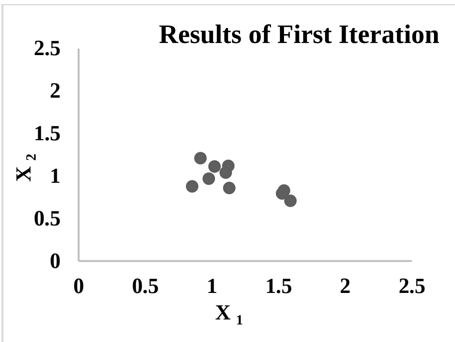



**Fig. 3.54** Position of particles at First Iteration

It is observed from Fig.3.54 that particles come closer in first iteration and therefore the search area reduces. It is also observed from column (6) of Table 3.10 and Table 3.11 that the difference in function values of their corresponding rows is not less than pre-defined value i.e.$10^{-6}$, therefore, Third iteration starts. Personal best value and global best values of position are updated. Using these values, the velocity and position of particles is updated. The iterations continue till the stopping criteria are met.

Table 3.12 shows the results of second iteration.

**TABLE 3.12**
**Result of Rosenbrock function at Second Iteration**

| Particles No. (1) | $v_1$ (2) | $v_2$ (3) | $x_1$ (4) | $x_2$ (5) | function value (6) |
|---|---|---|---|---|---|
| 1 | -0.2586 | 0.409037 | 0.7630 | 1.5172 | 87.4811 |
| 2 | -0.5655 | 0.1377 | 0.9765 | 0.9630 | 0.0092 |
| 3 | -0.5465 | -0.01335 | 0.5763 | 1.1015 | 59.3753 |
| 4 | -0.05883 | 0.3528 | 0.8554 | 1.5598 | 68.5946 |
| 5 | -0.54978 | 0.172125 | 0.9765 | 0.9630 | 0.00929 |
| 6 | -0.8438 | 0.511012 | 0.7461 | 1.2133 | 43.17359 |
| 7 | -0.65099 | 0.430691 | 0.4792 | 1.2853 | 111.7170 |
| 8 | -0.07398 | 0.515254 | 0.7776 | 1.3887 | 61.51340 |
| 9 | -0.21473 | 0.406216 | 0.7618 | 1.3692 | 62.28767 |
| 10 | -0.18361 | 0.059107 | 0.9209 | 1.0933 | 6.014429 |

Column (1) of Table 3.12 shows the particle number. Columns (2) and (3) show the velocity of particles calculated using equation (3.9). Columns (4) and (5) show the position of particles calculated using equation (3.10). Column (6) shows the function value.

Position of particles for Second iteration is shown in Fig. 3.55.

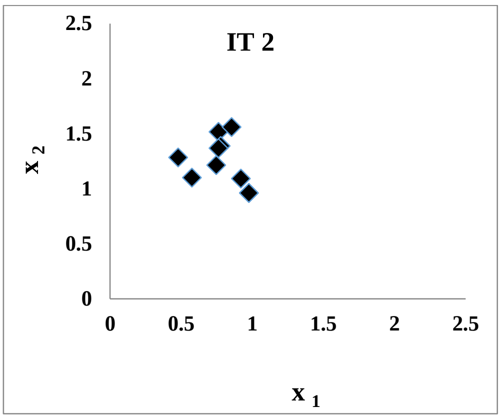



**Fig. 3.55** Position of particles at Second Iteration

It is observed from Fig.3.55 that the particles have come closer compared to that in the first iteration and the search area has reduced further. But the stopping criterion is still not met. Therefore, the procedure continues and the results upto ten iterations have been determined. Table 3.13 shows the results of Third iteration.

**TABLE 3.13**
**Result of Rosenbrock function at Third Iteration**

| Particles No. (1) | $v_1$ (2) | $v_2$ (3) | $x_1$ (4) | $x_2$ (5) | function value (6) |
|---|---|---|---|---|---|
| 1 | 0.226503 | -0.57458 | 0.989516 | 0.942618 | 0.13351 |
| 2 | -0.42413 | 0.103275 | 0.552456 | 1.066345 | 58.13330 |
| 3 | 0.427591 | -0.1378 | 1.003912 | 0.963737 | 0.19452 |
| 4 | 0.124078 | -0.61446 | 0.979527 | 0.945427 | 0.02015 |
| 5 | -0.41234 | 0.129094 | 0.56425 | 1.092164 | 60.06430 |
| 6 | -0.40241 | 0.133027 | 0.343732 | 1.346329 | 151.27300 |
| 7 | 0.529857 | -0.34385 | 1.009134 | 0.941535 | 0.59017 |
| 8 | 0.202631 | -0.45143 | 0.980284 | 0.937307 | 0.05632 |
| 9 | 0.225469 | -0.42653 | 0.987322 | 0.942759 | 0.013703 |
| 10 | 0.064785 | -0.13321 | 0.985766 | 0.960115 | 0.01370 |

Results of Table 3.13 is graphically represented in Fig.3.56 for position of particles for Third iteration.

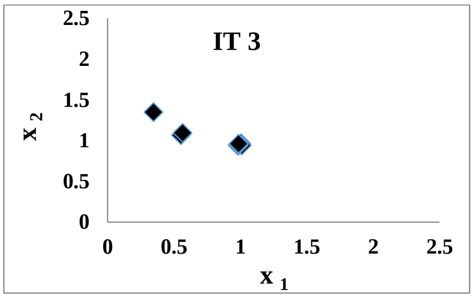



**Fig. 3.56** Position of particles at Third Iteration

Results of function value at updated position of particles for Fourth iteration is shown in Table 3.14.

**TABLE 3.14**
**Result of Rosenbrock function at Fourth Iteration**

| Particles No. (1) | $v_1$ (2) | $v_2$ (3) | $x_1$ (4) | $x_2$ (5) | function value (6) |
|---|---|---|---|---|---|
| 1 | 0.466542 | -0.1136 | 1.018998 | 0.9527 | 0.73333 |
| 2 | 0.271987 | -0.09712 | 1.275899 | 0.8666 | 58.0348 |
| 3 | 0.083913 | -0.41248 | 1.063439 | 0.5329 | 35.7589 |
| 4 | 0.123701 | -0.03873 | 0.687951 | 1.0534 | 33.7559 |
| 5 | 0.673094 | -0.39656 | 1.016826 | 0.9497 | 0.7087 |
| 6 | 0.338351 | -0.21916 | 1.347485 | 0.7223 | 119.660 |
| 7 | 0.138143 | -0.29024 | 1.118427 | 0.6470 | 36.4723 |
| 8 | 0.147091 | -0.27826 | 1.134413 | 0.6645 | 38.755 |
| 9 | 0.036169 | -0.09029 | 1.021934 | 0.8698 | 3.0463 |
| 10 | 0.17091 | -0.28826 | 1.164413 | 0.6545 | 37.7552 |

Results of Table 3.14 is graphically represented in Fig.3.57 for position of particles at Fourth iteration.

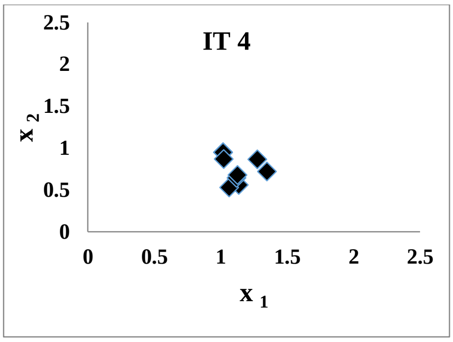



**Fig. 3.57** Position of particles at Fourth Iteration

Similarly, position of particles at Fifth, Sixth, Seventh, Eighth, Ninth and Tenth iteration are shown in Fig. 3.58, 3.59, 3.60, 3.61, 3.62 and 3.63 respectively.

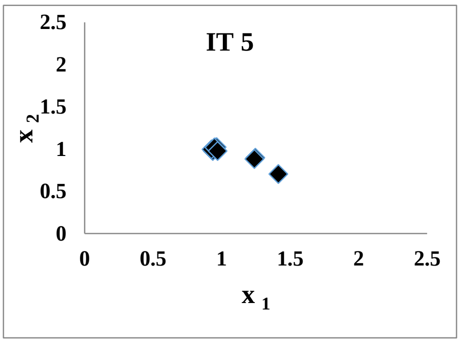



**Fig. 3.58** Position of particles at Fifth Iteration

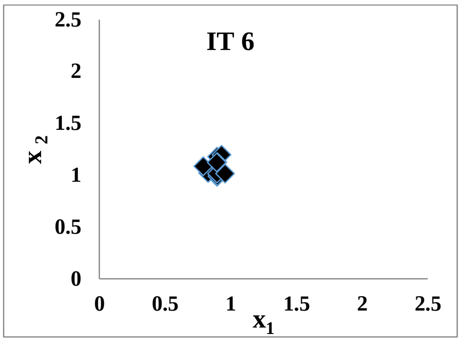



**Fig. 3.59** Position of particles at Sixth Iteration

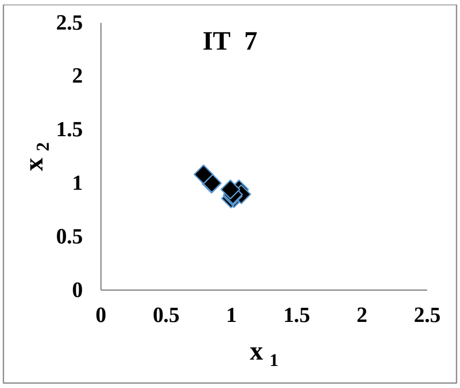



**Fig. 3.60** Position of particles at Seventh Iteration

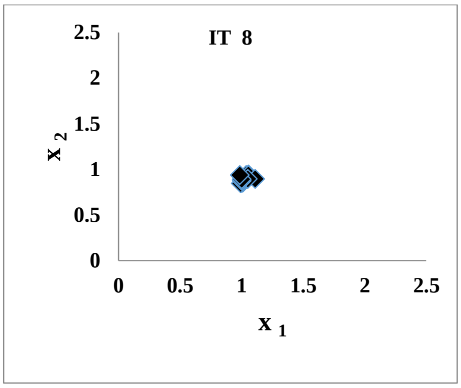



**Fig. 3.61** Position of particles at Eighth Iteration

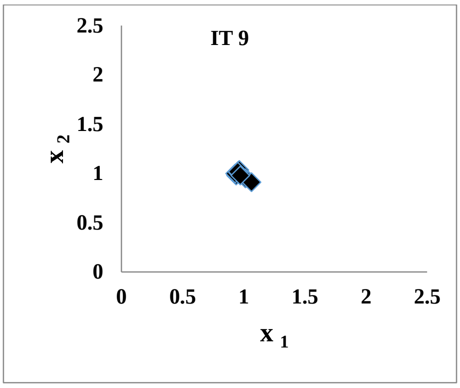



**Fig. 3.62** Position of particles at Ninth Iteration

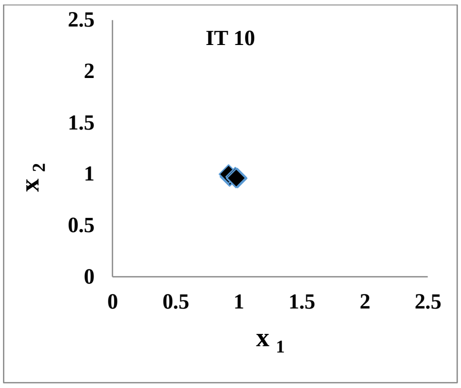



**Fig. 3.63** Position of particles at Tenth Iteration

Fig. 3.53 shows the particles at zeroth iteration. It is observed that they are spread over a large area. In the First iteration shown in Fig.3.54, it is observed that exploration area has reduced in comparison to initial position of particles.

As the iteration number increases, size of exploration area reduces further. In Figures 3.61 and 3.62 which show position of particles at eighth and ninth iteration respectively, search area is very small. At tenth iteration, all the particles have reached very close to optimum value and therefore, the function value obtained is also close to optimum. The function will get optimized accurately if the process is carried out for more number of iterations. In this section, the aim is to explain the process of optimization in detail for the given parameters of PSO.

### 3.7.1 Effect of Variation of Population Size (q) and $IT_{max}$

Population Size (q) and Maximum number of iterations ($IT_{max}$) have been varied systematically to observe the effect on convergence. One of the parameters is varied while all other are kept fixed. This study has been conducted with the help of MATLAB programme in which $r_p$ and $r_g$ are fixed to 0.4 and 0.5 respectively while $C_p$ and $C_g$ are fixed to two (2) and linearly decreasing inertia weight has been considered.

## Population Size (q)

In this case, all the parameters have been fixed to the values shown above and $IT_{max}$ has been fixed to 100. The population size is varied from 10 to 60. The results are shown in Table 3.15. Column (1) of Table 3.15 shows the number of particles. Columns (2) and (3) show the values of variables at which function is optimized.Columns (4) and (5) show the optimized value of function and number of iterations required to optimize the function. It is observed that the function does not get optimimized for population size of 10 and 20 for $IT_{max} = 100$. Howere,it gets optimized accurately for 30 and above particle size.

**TABLE 3.15**
**Effect of Population Size**

| **q (1)** | **$x_1$ (2)** | **$x_2$ (3)** | **function value (4)** | **Iteration k (5)** |
|---|---|---|---|---|
| 10 | 1.220362 | 1.489734 | 4.85797E-002 | 100 |
| 20 | 9.53764e-001 | 9.080800e-001 | 2.39098E-003 | 100 |
| 30 | 1 | 1 | 3.989808E-020 | 62 |
| **40** | **1** | **1** | **1.828370E-022** | **51** |
| 50 | 1 | 1 | 1.828370E-022 | 60 |
| 60 | 1 | 1 | 8.692590E-020 | 58 |

It is further observed that for a population size of 40, function gets optimized with maximum accuracy in minimum number of iterations.

## Maximum Iteration ($IT_{max}$)

In this study, for each fixed value of population size, maximum number of iteration ($IT_{max}$) is varied from 30 to 200. The results are shown in Table 3.16. The value of other parameters have been fixed to the values given below:

| | | |
|---|---|---|
| $q = 10$; | $r_p = 0.4$; | $r_g = 0.5$; |
| $C_p = 2$; | $C_g = 2$; | $IT_{max} = 10$; |
| $W_{max} = 0.9$; | $W_{min} = 0.4$; | $\varepsilon = 1 * 10^{-6}$ |

The population size has been fixed to 30, 40, 50 and 100. The results are shown in Table 3.16. In Table 3.16, columns (2) and (3) represent population size and maximum number of iteration $IT_{max}$ respectively. Columns (4) and (5) represent the value of variables $x_1$ and $x_2$ at which function is optimized. Columns (6) represents the function value. Column (7) represents the iterations required for convergence.

**TABLE 3.16**
**Effect of $IT_{max}$ on accuracy of convergence**

| S. No. (1) | q (2) | $IT_{max}$ (3) | $x_1$ (4) | $x_2$ (5) | Function value (6) | k (7) |
|---|---|---|---|---|---|---|
| 1 | 30 | 30 | 1 | 1 | 1.145576e-011 | 30 |
| 2 | 30 | 40 | 1 | 1 | 1.375645e-011 | 32 |
| 3 | 30 | 50 | 1 | 1 | 2.5987932e-013 | 40 |
| 4 | 30 | 60 | 1 | 1 | 6.940071e-015 | 41 |
| 5 | 30 | 70 | 1 | 1 | 1.216929e-013 | 39 |
| 6 | 30 | 80 | 1 | 1 | 2.343508e-014 | 41 |
| **7** | **30** | **110** | **1** | **1** | **4.763705e-020** | **55** |
| 8 | 30 | 140 | 1 | 1 | 1.675177e-019 | 50 |
| 9 | 30 | 200 | 1 | 1 | 1.821156e-018 | 53 |
| 10 | 40 | 90 | 1 | 1 | 9.198076e-018 | 48 |
| 11 | 40 | 100 | 1 | 1 | 1.944461e-017 | 45 |
| 12 | 40 | 120 | 1 | 1 | 1.149075e-017 | 48 |
| 13 | 40 | 140 | 1 | 1 | 1.216215e-019 | 53 |
| 14 | 40 | 160 | 1 | 1 | 2.827675e-019 | 52 |
| **15** | **40** | **180** | **1** | **1** | **1.093662e-020** | **57** |
| 16 | 40 | 200 | 1 | 1 | 2.085538e-018 | 54 |
| 17 | 50 | 30 | 1 | 1 | 1.415376e-012 | 30 |
| 18 | 50 | 50 | 1 | 1 | 3.085142e-014 | 36 |
| 19 | 50 | 60 | 1 | 1 | 6.603557e-016 | 41 |
| 20 | 50 | 70 | 1 | 1 | 1.532676e-014 | 41 |
| 21 | 50 | 80 | 1 | 1 | 1.308451e-015 | 47 |
| 22 | 50 | 100 | 1 | 1 | 4.974816e-018 | 50 |
| 23 | 50 | 120 | 1 | 1 | 5.543777e-017 | 47 |
| **24** | **50** | **140** | **1** | **1** | **2.060528e-021** | **57** |
| 25 | 50 | 160 | 1 | 1 | 1.193581e-017 | 48 |
| 26 | 50 | 180 | 1 | 1 | 5.891339e-021 | 58 |
| 27 | 50 | 200 | 1 | 1 | 2.357654e-019 | 56 |
| 28 | 50 | 220 | 1 | 1 | 9.522462e-018 | 51 |
| 29 | 100 | 50 | 1 | 1 | 2.65228e-015 | 40 |
| 30 | 100 | 60 | 1 | 1 | 3.845000e-019 | 45 |
| 31 | 100 | 70 | 1 | 1 | 4.959005e-018 | 48 |
| 32 | 100 | 80 | 1 | 1 | 5.829558e-020 | 49 |
| 33 | 100 | 90 | 1 | 1 | 9.005843e-019 | 49 |
| 34 | 100 | 100 | 1 | 1 | 30268523e-021 | 57 |
| 35 | 100 | 110 | 1 | 1 | 2.1000047e-016 | 43 |
| 36 | 100 | 120 | 1 | 1 | 3.910962e-020 | 57 |
| 37 | 100 | 140 | 1 | 1 | 4.449501e-019 | 51 |
| 38 | 100 | 160 | 1 | 1 | 2.142912e-018 | 50 |
| 39 | 100 | 180 | 1 | 1 | 1.868118e-019 | 53 |
| **40** | **100** | **200** | **1** | **1** | **6.037815e-022** | **58** |
| 41 | 100 | 220 | 1 | 1 | 2.402596e-020 | 59 |

From Table 3.16, it is observed that maximum accuracy is obtained for the following combinations of population size (P) and $IT_{max}$ which are shown in Table 3.17.

**TABLE 3.17**
**Best Combinations of P and $IT_{max}$**

| **P** | **$IT_{max}$** | **$x_1$** | **$x_2$** | **Function value** | **k** |
|---|---|---|---|---|---|
| 30 | 110 | 1 | 1 | 4.763705e-020 | 55 |
| 40 | 180 | 1 | 1 | 1.093662e-020 | 57 |
| 50 | 140 | 1 | 1 | 2.060528e-021 | 57 |
| 100 | 200 | 1 | 1 | 6.037815e-022 | 58 |

## 3.8 CONCLUSIONS

### 3.8.1 Genetic Algorithm

GA has been implemented on Rosenbrock function for various combinations of bit size and population size. For combination 1 and 2 (bit size 9 and 8, population size 10), function converged at near optimal value. In case of small bit size (3 and 5) and population size (7 and 6) function did not converge. It is concluded that population size and bit size should be large enough so that it can support sufficient genetic variation and therefore higher accuracy can be achieved. Model (i) is the best model to represent the relation between generation and population size. In this model maximum number of points coincide with the graph points.

### 3.8.2 Basic Particle Swarm Optimization

Basic Particle swarm optimization (BPSO) algorithm has been implemented to optimize the Rosenbrock function manually as well as by MATLAB programme. Two important parameters of BPSO - population size (P) and maximum number of iterations ($IT_{max}$) have been varied keeping other parameters fixed to some predefined value to study their effect on convergence of accuracy. For small population size, i.e. 10 and 20, function does not get optimized accurately in specified $IT_{max}$. For small population sizes, if $IT_{max}$ is increased then function gets optimized accurately but the convergence is slower. The maximum accuracy and fast convergence is achieved for population size 40. For population size greater than 40, convergence is slower. It is observed that the population size should not be less than 20 to optimize the function accurately.

## Research Publications :


[1] N. K. Jain, Uma Nangia, Jyoti Jain, "Effect of population and bit size on optimization of function by Genetic Algorithm," International Conference on Computing for Sustainable Global Development, Bharti Vidyapeeth's Institute of Computer Applications and Management (BVICAM), Delhi, pp.194-199, (16-18 March, 2016).

[2] N. K. Jain, Uma Nangia, Jyoti Jain, "Impacts of PSO Parameters on its Convergence," IEEE second International Conference on Power Electronics, Intelligent Control and Energy, (ICPEICES 2016), Delhi Technological University (DTU), Delhi, (22-24 October, 2018).

# CHAPTER 4
# IMPROVED PSO ALGORITHMS

## 4.1. INTRODUCTION

The concept of Basic Particle Swarm Optimization (BPSO) was introduced by Eberhart and Kennedy [10,11] in 1995. BPSO in its original form had various issues - chances of getting trapped in local minima, poor computational efficiency, poor accuracy and premature convergence. Therefore, two improved algorithms have been developed to improve the performance of Basic Particle Swarm Optimization (BPSO).

The two improved algorithms proposed in this chapter are: Particle Swarm Optimization based on Initial Selection of Particles called as IPSO IS [171, 172] and Adaptive Social Acceleration Constant based PSO (ASACPSO) [186]. These have been discussed in section 4.2 and 4.3 respectively. In IPSO IS, the improvement has been carried out by choosing the particles based on minimum function value and ASACPSO has been developed using best value of social acceleration constant. Both algorithms have been implemented on Mathematical benchmark functions and Economic Load dispatch problem of IEEE 5, 14 and 30 bus systems.

## 4.2 IMPROVED PSO BASED ON INITIAL SELECTION OF PARTICLES (IPSO IS)

In this section, an Improved version of Particle Swarm Optimization - IPSO IS has been developed, which is based on selection of better population out of initially generated population of points. The minimum population size "Pmin" at which the objective function converges has been observed from Basic Particle Swarm Optimization (BPSO). In improved PSO based on Initial Selection of Particles (IPSO IS), initial population size of "q" upto two to four times of Pmin has been considered. From this initial population size, better particles based on minimum function value are selected. This population of particles is then used to minimize the function.

### 4.2.1 Implementation of IPSO IS and BPSO on Benchmark functions

Following are the steps required to optimize Mathematical Benchmark function by Improved Particle Swarm Optimization based on Initial Selection of Particles (IPSO IS) algorithm:

1. Initialize the parameters of IPSO IS i.e. q = initial number of particles generated randomly, P = Number of selected particles, $C_p$, $C_g$, $r_p$, $r_g$, $IT_{max}$, W, Ɛ, Kount.
2. Set Iteration count k = 0.
3. Initialize the velocities and positions for the variables. (No. of elements in the vectors is equal to number of particles i.e. "q").
4. Calculate the fitness function value for each particle using equation (3.1).
5. *Select "P" particles based on minimum function value.*
6. Determine $x_{pbestij,}$ for all the particles "P". The personal best position associated with $j^{th}$ particle for $i^{th}$ variable is the best position that the particle has visited yielding the lowest fitness value for that particle.
7. Determine the $x_{gbesti}$. If f (best of $x_{pbestij}$) is less than f ($x_{gbesti}$), then assign the value of best of $x_{pbestij}$ to the $x_{gbesti.}$ The best position associated with $j^{th}$ particle that any particle in the swarm has visited yielding the lowest fitness value for that particle. This represents the best fitness of all the selected particles "P" particles of a swarm at any point of time.

8. Update the velocities of all particles using equation (3.9) and positions of all particles using equation (3.10).

9. Determine the fitness function value for each particle.

10. Check f, the stopping criteria is met, if yes, go to 11; else go to 4.

11. Display $x_{gbesti}$ as the optimal solution and the fitness corresponding to it as the optimum function.

Flow chart for improved PSO based on Initial Selection of Particles (IPSO IS) is shown in Fig.4.1.

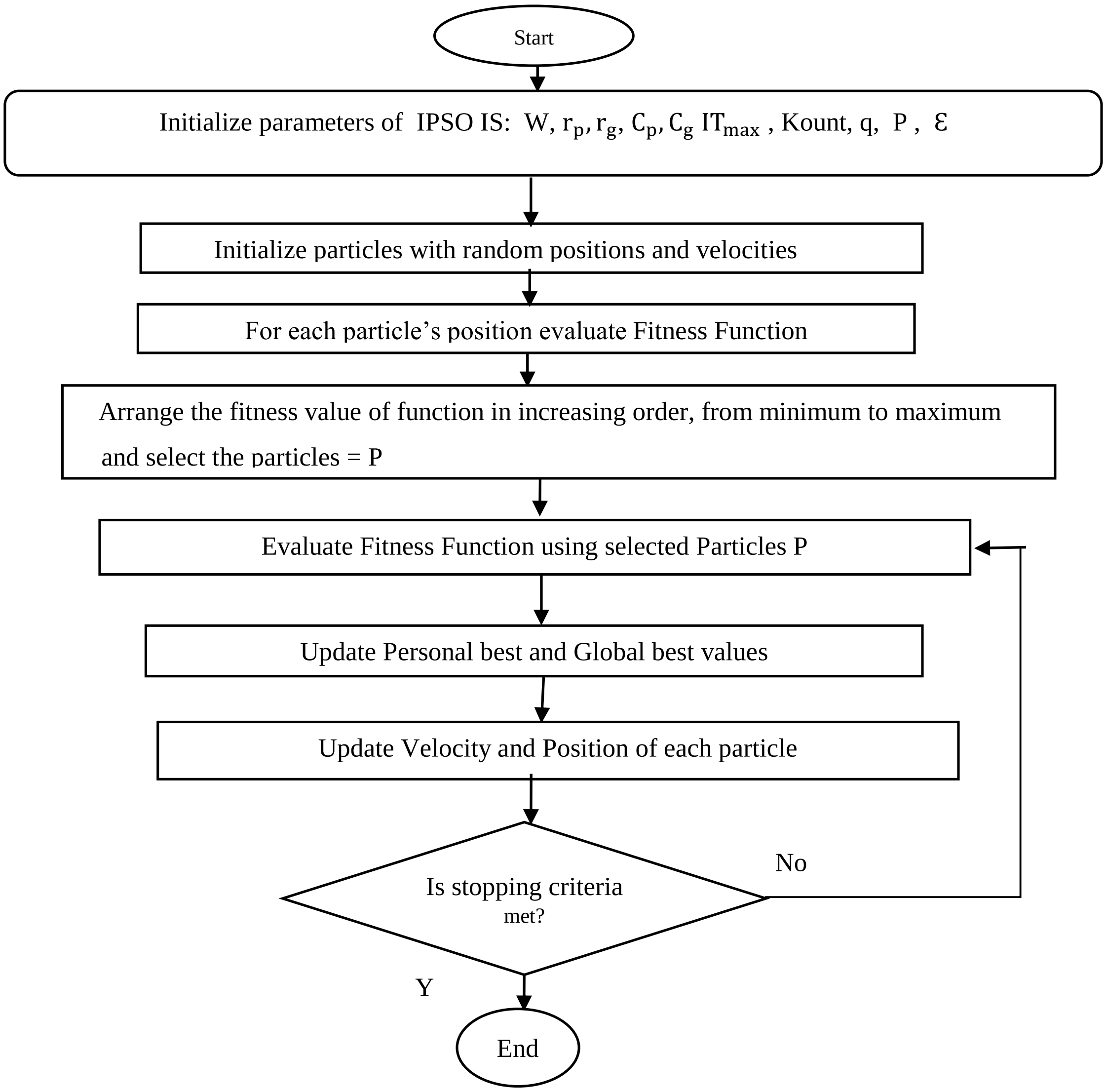



**Fig. 4.1** Flow chart for IPSO IS

The following values of parameters have been considered for Basic Particle Swarm Optimization (BPSO) and Improved Particle Swarm Optimization based on Initial Selection of Particles (IPSO IS):

$r_p = r_g = 1$; $C_g = C_p = 1$; $W = 0.6$ ; $IT_{max} = 1000$ ; P = Selected particles

q = Initial particles. $\mathcal{E} = 1 * 10^{-6}$

The Experimentation has been done on following four mathematical benchmark functions.

1. Rosenbrock Function
2. Beale Function
3. Booth Function
4. Hyper Ellipsoid Function

#### 4.2.1.1 Rosenbrock function

Mathematically, it is defined as

$$f = 100 * (x_1^2 - x_2)^2 + (1 - x_1)^2 \tag{4.1}$$

f (1,1) = 0 Range: $x_1$ Ɛ (0, 2) and $x_2$ Ɛ (0, 2)

In Basic Particle Swarm Optimization (BPSO), the initial population and velocities are generated randomly using rand command of MATLAB programme, where q is number of initial particles and this population is used to minimize the Mathematical Benchmark Function. Initially Basic Particle Swarm Optimization (BPSO) has been applied by varying number of particles from 10 to 100 in step of 10. The results are shown in Table 4.1. The column (2) shows the number of particles. The columns (3) and (4) show the number of Iterations (k) and Kount (N) required to optimize the function respectively. **Kount** (N) in this case is number of function evaluations, i.e. how many times a function has been evaluated during the optimization process. This was taken as the measure of computational time. The measure of computational time in seconds or minutes will depend on the speed of the processor used in computer, i.e. the type of computer being used. Therefore, the time has been measured in terms of function evaluations.

**TABLE 4.1**
**Results of BPSO by varying number of Particles**

| S. No. (1) | q (2) | k (3) | Kount (N) (4) | Result (5) |
|---|---|---|---|---|
| 1 | 100 | 102 | 20601 | Ok |
| 2 | 90 | 89 | 16201 | Ok |
| 3 | 80 | 101 | 16321 | Ok |
| 4 | 70 | 89 | 12601 | Ok |
| 5 | 60 | 89 | 10801 | Ok |
| 6 | 50 | 75 | 7601 | Ok |
| 7 | 40 | 86 | 6961 | Ok |
| 8 | 30 | 82 | 4981 | Ok |
| 9 | 20 | 90 | 3641 | Ok |
| 10 | 10 | 75 | 1521 | No convergence |

The results of Table 4.1 show that the Kount (N) increases as the number of particles increases. It is also observed, that Basic Particle Swarm Optimization (BPSO) could not optimize the function with number of particles, "q" = 10.

IPSO IS has been applied to Rosenbrock function. In IPSO IS, it is experimented to choose better particles "P" from the initially generated "q" particles. Better particles have been obtained based on the function value. For one particular value of "P", various values of "q" have been tried i.e. by fixing the number of particles, "P" to ten, various values of initial particle sizes "q" have been tried by varying "q" from 10 to 100 in steps of ten. The best results in terms of minimum Kount for each "P" and corresponding "q" have been tabulated in Table 4.2. In Table 4.2 columns (2) and (3), show the size of initial particles, "q" and the size of selected particles, "P" respectively. Column (4) shows the number of Iterations (k) required to optimize the function using IPSO IS. Column (5) and (6) show the (number of function evaluations) Kount, for Improved PSO based on initial selection of particles (IPSO IS) and BPSO respectively, i.e. Kount (S): Kount for IPSO IS and Kount (N): Kount for BPSO.

**TABLE 4.2**
**Best number of Particles "q" for given selected Particles "P"**

| S. No. (1) | q (2) | P (3) | k (4) | Kount (S) IPSO IS (5) | Kount (N) BPSO (6) | Results (7) | % Saving in Kount (8) |
|---|---|---|---|---|---|---|---|
| 1 | 100 | 100 | 102 | 20601 | 20601 | Ok | NA |
| 2 | 100 | 80 | 81 | 13141 | 16321 | Ok | 19.48 |
| 3 | 100 | 70 | 84 | 11931 | 12601 | Ok | 5.31 |
| 4 | 100 | 40 | 71 | 5821 | 6961 | Ok | 19.58 |
| 5 | 90 | 90 | 89 | 16201 | 16201 | Ok | 0 |
| 6 | 90 | 60 | 69 | 8431 | 10801 | Ok | 19.48 |
| 7 | 70 | 20 | 64 | 2651 | 3641 | Ok | 37.3 |
| 8 | 50 | 50 | 75 | 7601 | 7641 | Ok | 0.52 |
| 9 | 50 | 30 | 54 | 3321 | 4981 | Ok | 49.9 |
| 10 | 50 | 10 | 68 | 1421 | N.A | No-Convergence | 100 |
| 11 | 50 | 5 | 59 | 646 | N.A. | No-Convergence | 100 |

It is observed from Table 4.2, that number of Kount for IPSO IS (Kount (S)) are always less than the number of Kount for BPSO (Kount (N)). The last column (8) shows the percentage saving in Kount for Rosenbrock function. It was observed that the Rosenbrock function could not be optimized by BPSO for particle size ten (10). However, when ten (10) particles were selected out of '50' using IPSO IS, the function could be optimized in 1421 Kount and 68 Iterations.

Further investigations have been done on other standard test functions.

### 4.2.1.2 Beale Function

Mathematically, Beale function is defined as

$$f = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2 \quad (4.2)$$

$f(3, 0.5) = 0$ Range $x_1 \geq -4.5$, $x_2 \leq 4.5$

Initially the particles for variables $x_1$ and $x_2$ have been generated in their range by MATLAB command unifrnd (-4.5, 4.5, q, 1) and velocity by rand (1, q). Initially Basic PSO (BPSO) has been applied by varying number of particles from 100 to 10 in step of 10. The results are shown in Table 4.3 for Beale function.

**TABLE 4.3**
**Results of BPSO by varying number of Particles**

| S. No. (1) | q (2) | k (3) | Kount (N) (4) | Result (5) |
|---|---|---|---|---|
| 1 | 100 | 78 | 15801 | Ok |
| 2 | 90 | 103 | 18721 | Ok |
| 3 | 80 | 110 | 17761 | Ok |
| 4 | 70 | 79 | 11201 | Ok |
| 5 | 60 | 74 | 9001 | Ok |
| 6 | 50 | 84 | 8501 | Ok |
| 7 | 40 | 69 | 5601 | Ok |
| 8 | 30 | 82 | 4981 | Ok |
| 9 | 25 | 73 | 3701 | Ok |
| 10 | 20 | NA | NA | No convergence |
| 11 | 10 | NA | NA | No convergence |

It is observed from Table 4.3 that minimum value of "q" for which Beale function could be minimized is 25. Next, an Improved PSO based on Initial Selection of Particles is applied to Beale Function keeping "q" fixed to 25 and varying "P" from size 25 to 10. The results are shown in Table 4.4.

**TABLE 4.4**
**Best number of selected Particles "P" for initial Particles "q" = 25**

| S. No. (1) | q (2) | P (3) | k (4) | Kount (S) (5) | Result (6) | % Saving in Kount (7) |
|---|---|---|---|---|---|---|
| 1 | 25 | 25 | 73 | 3702 | Ok | NA |
| 2 | 25 | 20 | 57 | 2326 | Ok | 37.16 |
| 3 | 25 | 15 | 66 | 2021 | Ok | 45.40 |
| 4 | 25 | 10 | 49 | 1016 | Ok | 72.55 |

It is observed that using IPSO IS, this function could be optimized for particle size 10. Saving in terms of function evaluations was achieved in all cases and maximum saving was achieved when 10 particles are selected out of 25.

### 4.2.1.3 Booth Function

Mathematically, it is defined as

$$f = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \qquad (4.3)$$

f (1,3) = 0 Range $x_1 \geq -10$, $x_2 < 10$

The results are shown in Table 4.5 for Booth function.

**TABLE 4.5**
**Results of BPSO by varying number of Particles**

| S.No (1) | q (2) | k (3) | Kount (N) (4) | Result (5) |
|---|---|---|---|---|
| 1 | 100 | 74 | 15001 | Ok |
| 2 | 90 | 85 | 15481 | Ok |
| 3 | 80 | 80 | 12961 | Ok |
| 4 | 70 | 67 | 9521 | Ok |
| 5 | 60 | 87 | 10561 | Ok |
| 6 | 50 | 77 | 7801 | Ok |
| 7 | 40 | 81 | 6561 | Ok |
| 8 | 30 | 87 | 5281 | Ok |
| 9 | 20 | 84 | 3401 | Ok |
| 10 | 10 | 59 | 1201 | Ok |

Here we observe that minimum value of q for which Booth function could be minimized is 10. Next, an Improved PSO based on Initial Selection of Particles (IPSO IS) is applied to Booth function keeping “q” fixed to 10 and varying “P” from size 10 to 5. The results are shown in Table 4.6.

**TABLE 4.6**
**Best number of selected Particles “P” for initial size “q” = 10**

| S. No. (1) | q (2) | P (3) | k (4) | Kount (S) (5) | Result (6) | % Saving in Kount (7) |
|---|---|---|---|---|---|---|
| **1** | 10 | 10 | 59 | 1201 | Ok | NA |
| **2** | 10 | 8 | 54 | 883 | Ok | 26.48 |
| **3** | 10 | 6 | 91 | 1109 | Ok | 7.67 |
| **4** | 10 | 5 | 86 | 876 | Ok | 27.06 |

### 4.2.1.4 Axis Parallel Hyper Ellipsoid Function

Mathematically, it is defined as

$$f=\sum_{i=1}^{n} i\, x_i^2 \qquad (4.4)$$

f (x) = 0 at i =1: n here n=2 Range $-5.2 \leq x_i \leq 5.12$

Initially Basic PSO (BPSO) has been applied by varying number of particles from 50 to 5 for Axis Parallel Hyper Ellipsoid Function. The results are shown in Table 4.7

**TABLE 4.7**
**Results of BPSO by varying number of Particles**

| S. No. (1) | q (2) | k (3) | Kount (4) | Result (5) |
|---|---|---|---|---|
| 1 | 50 | 60 | 6101 | Ok |
| 2 | 40 | 73 | 5921 | Ok |
| 3 | 30 | 51 | 3601 | Ok |
| 4 | 25 | 47 | 2401 | Ok |
| 5 | 20 | 52 | 2121 | Ok |
| 6 | 15 | 79 | 2401 | Ok |
| 7 | 10 | NA | NA | No-convergence |
| 8 | 5 | NA | NA | No-convergence |

It is observed from results of Table 4.7 that minimum value of "q" for which Axis Parallel Hyper Ellipsoid Function could be minimized is 15. However, the minimum Kount was obtained for 20 particles. Therefore, an initial size "q" of 20 and 15 particles has been considered. Initial selection of Particles based algorithm (IPSO IS) is applied on Axis Parallel Hyper Ellipsoid function and results are shown in Table 4.8.

**TABLE 4.8**
**Best numbers of selected Particles "P" for initial Particles "q" = 20 and 15**

| S. No. (1) | q (2) | P (3) | k (4) | Kount (S) (5) | Result (6) | % Saving in Kount (7) |
|---|---|---|---|---|---|---|
| 1 | 20 | 20 | 52 | 2121 | Ok | NA |
| 2 | 20 | 5 | 73 | 756 | Ok | 64.35 |
| 3 | 15 | 15 | 79 | 2401 | Ok | NA |
| 4 | 15 | 10 | 50 | 1026 | Ok | 51.63 |

The results of BPSO for minimum particle size “q” i.e. Pmin have been summarized in Table 4.9. Pmin was the minimum number of particles at which basic PSO optimized accurately.

**TABLE 4.9**
**Minimum Value of Population Pmin for BPSO**

| S. No. (1) | Function (2) | Pmin (3) | k (4) | Kount (N) (5) |
|---|---|---|---|---|
| 1 | **Rosenbrock** | 20 | 90 | 3641 |
| 2 | **Beale** | 25 | 73 | 3701 |
| 3 | **Booth** | 10 | 59 | 1201 |
| 4 | **Axis Parallel Hyper Ellipsoid** | 15 | 79 | 2401 |

Various combinations of “P” and “q” have been tried for all the functions. Table 4.10 shows the results for all test functions for values of “q” higher than Pmin and values “P” less than or equal to Pmin for each function.

**TABLE 4.10**
**Results of IPSO IS for Values of “q” Higher than Pmin and “P” Less than Pmin**

| S. No. (1) | Functions (2) | q (3) | P (4) | k (5) | Kount (6) | Saving in Kount % (7) |
|---|---|---|---|---|---|---|
| **1** | **Rosenbrock** | 20 | 20 | 90 | 3641 | NA |
| | | 50 | 10 | 79 | 1641 | 54.92 |
| | | 50 | 5 | 59 | 646 | 82.25 |
| **2** | **Beale** | 25 | 25 | 73 | 3702 | NA |
| | | 25 | 10 | 49 | 1016 | 72.55 |
| | | 50 | 8 | 55 | 939 | 74.63 |
| **3** | **Booth** | 10 | 10 | 59 | 1201 | NA |
| | | 10 | 5 | 86 | 876 | 27.06 |
| | | 50 | 8 | 43 | 747 | 37.80 |
| **4** | **Axis Parallel Hyper Ellipsoid** | 15 | 15 | 79 | 2401 | NA |
| | | 15 | 10 | 50 | 1026 | 51.63 |
| | | 50 | 5 | 62 | 676 | 71.84 |

Table 4.10 shows the reduction in Kount using IPSO IS. It has been observed that every function converged for some minimum size of particles, Pmin. In order to reduce the Kount, the particles have been selected out of double or more than double of initial size. In case of Rosenbrock function Basic Particle Swarm Optimization (BPSO) algorithm required 20 minimum particles for convergence, whereas IPSO IS, selecting 5 particles from 50 initial particles gave saving of 82.25% in Kount. Axis Parallel Hyper Ellipsoid function converged for minimum particle size (Pmin) 15. Now, 5 particles have been selected out of double or more than double the value of Pmin i.e. out of 50. When 5 particles out of 50 are selected, the Kount reduces considerably i.e. a saving of 71.84% has been achieved. The saving has been calculated using the formula given below:

**Saving in Kount = ((Kount (N) – Kount (S)) / Kount (N)**

Saving in Kount is plotted in Fig. 4.2. Brown color shows Kount (N) required for "Pmin" of Basic PSO and Black color shows Kount (S) of Improved PSO (IPSO IS) based on Initial Selection of particles for "q = 50" and minimum selected particles "P < Pmin".

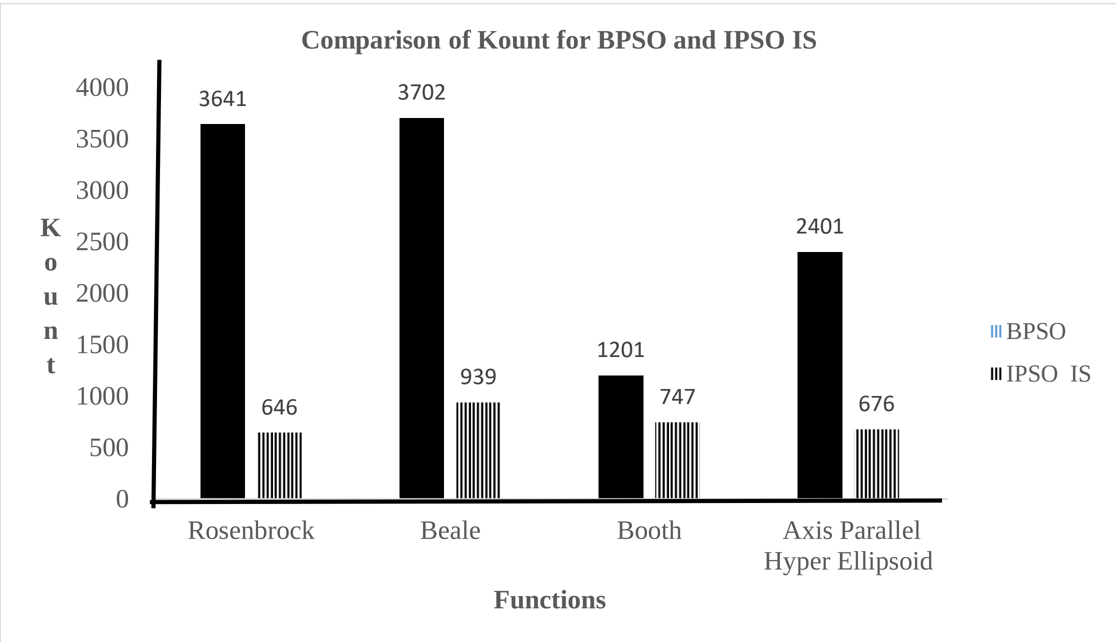



**Fig. 4.2** Comparison of Kount for BPSO and IPSO IS

Results of BPSO for Pmin and IPSOIS for "q" = 50 and "P" = 10 has been presented in Table 4.11.

**TABLE 4.11**
**Results of IPSO IS for "q" = 50 and "P" = 10**

| S. No. (1) | Function (2) | q (3) | P (4) | k (5) | Kount (S) (6) | %Saving in Kount (S) (7) |
|---|---|---|---|---|---|---|
| **1** | Rosenbrock | 20 | 20 | 90 | 3641 | NA |
| | | 50 | 10 | 79 | 1641 | 54.93 |
| **2** | Beale | 25 | 25 | 73 | 3701 | NA |
| | | 50 | 10 | 71 | 1481 | 59.98 |
| **3** | Booth | 10 | 10 | 59 | 1201 | NA |
| | | 50 | 10 | 66 | 1381 | *-14.99 |
| | | 50 | 8 | 43 | 747 | 37.8 |
| **4** | Axis Parallel Hyper Ellipsoid | 15 | 15 | 79 | 2401 | |
| | | 50 | 10 | 42 | 901 | 62.47 |

From the above Table 4.11 it was observed that saving in Kount for "q" = 50 and "P" = 10 were 54.93%, 59.98%, 62.47% for Rosenbrock, Beale and Axis Parallel Hyper Ellipsoid functions respectively. In case of Booth function, saving was not obtained for 10 particles but saving of 37.8% was obtained when 8 particles have been selected out of 50.

## 4.3 ADAPTIVE SOCIAL ACCELERATION CONSTANT BASED PSO (ASACPSO)

Adaptive Social Acceleration Constant based PSO has been developed using the best value of Social Acceleration Constant. Social acceleration constant has the ability to concentrate the search around a promising area to refine a candidate solution. It also determines the tendency of velocity update. Large value of Social Acceleration Constant directs the particle towards the neighbor group's best. Social acceleration constant Cg regulates the maximum step size in the direction of the global best particle. ASACPSO has been implemented on seven mathematical benchmark functions and its performance has been compared with Basic Particle Swarm Optimization (BPSO).

In basic PSO, $C_p$ and $C_g$ are fixed to 1.0, whereas in ASACPSO, $C_g$ is searched adaptively and is named as adaptive social acceleration constant $C_{sg}$. The adaptive social acceleration based constant $C_{sg}$ has been formulated using following three functions:

I. Linearly decreasing function

$$C_{sg1} = C_{gmax} - (k*(C_{gmax} - C_{gmin})) / IT_{max}) \quad (4.5)$$

II. Exponential of linearly decreasing function

$$C_{sg2} = \exp((-1)* (C_{gmax} - (k*(C_{gmax} - C_{gmin}))/ IT_{max})) \quad (4.6)$$

III. Exponential decreasing function consisting of exponential of constant multiplied by linearly decreasing function.

$$C_{sg3} = \exp ((-0.34)* (C_{gmax} - (k*(C_{gmax} - C_{gmin})) / IT_{max})) \quad (4.7)$$

$IT_{max}$ Maximum number of Iterations

k Current Iteration

$C_{gmax}$ Maximum value of Social acceleration constant

$C_{gmin}$ Minimum value of Social acceleration constant

$C_{sg}$ Adaptive social acceleration constant

$C_{sg1}$ Social acceleration constant based on linearly decreasing function

$C_{sg2}$ Social acceleration constant based on exponential of linearly decreasing function

$C_{sg3}$ Social acceleration constant based on exponentially decreasing function consisting of exponential of constant multiplied by linearly decreasing function.

Algorithms based on $C_{sg1}$, $C_{sg2}$, and $C_{sg3}$ have been named as Adaptive Linearly decreasing PSO (ALDPSO), Adaptive Exponential Linearly Decreasing PSO (AELDPSO-I) and Exponential decreasing function consisting of constant multiplied by linearly decreasing function (AELDPSO-II) respectively.All the three algorithms have been implemented to minimize the Rosenbrock function. The best value of Adaptive–Social Acceleration Constant $C_{sg}$ is searched by applying equation (4.5), (4.6) and (4.7) after second, third, fourth and up to seventh Iteration. When these equations are applied after seventh Iteration, there was no improvement in the results. The results of ALDPSO, AELDPSO - I and AELDPSO - II are shown in

Tables 4.12, 4.13 and 4.14 respectively. The best value of $C_{sg}$ is selected based on minimum number of Kount required to minimize the function. ASACPSO Algorithm is developed using the best value of $C_{sg}$.

The parameters for all the three algorithms are fixed as follows:

$C_{gmax}$ = 1.5; $C_{gmin}$ = 0.0; P = Population Size = 40

$IT_{max} =$ Maximum number of Iterations =1000;

Results for Rosenbrock Function using ALDPSO are shown in Table 4.12.

**TABLE 4.12**
**Results of Rosenbrock function using ALDPSO**

| Condition for implementation of $C_{sg1}$ (1) | Results (2) | k (3) | Kount (4) | $C_{sg1}$ (5) |
|---|---|---|---|---|
| k > 2 | Success | 45 | 1840 | 0.955 |
| k > 3 | Success | 46 | 1880 | 0.954 |
| k > 4 | Success | 42 | 1720 | 0.958 |
| k > 5 | **Success** | **40** | **1640** | **0.960** |
| k > 6 | Success | 48 | 1960 | 0.952 |
| k > 7 | Success | 44 | 1800 | 0.956 |

Column (1) of Table 4.12 shows the number of Iterations after which $C_{sg1}$ **(**defined by equation (4.5) has been implemented. Column (2) shows the result of Rosenbrock function for defined condition of column (1). Columns (3) and (4) represent number of Iterations and number of Kount required respectively to optimize the function.

Column (5) shows value of $C_{sg1}$ searched for Rosenbrock function. When linear decreasing function $C_{sg1}$ implemented after second Iteration Kount required to optimize the function is 1840 and an adaptive social acceleration constant value is 0.955 as shown in row (2) of Table 4.12. It is observed from column (4) of Table 4.12 minimum Kount is obtained when adaptive $C_{sg2}$ has been searched after five Iterations and its value is 0.96. It has been highlighted in the Table 4.12. When this strategy is applied after eighth Iteration, no improvement has been observed in results.

Results for Mathematical Rosenbrock function using AELDPSO-I is shown in Table 4.13. Column (1) of Table 4.13 shows the number of Iterations after which $C_{sg2}$, (defined by equation (4.6) has been implemented. Column (2) shows the result of Rosenbrock function for defined condition of column (1). Columns (3) and (4) represents number of Iterations and number of Kount respectively required to optimize the function. Column (5) shows value of $C_{sg2}$ searched for Rosenbrock function. By observing column (4) of Table 4.13 it is found that minimum Kount is obtained when adaptive $C_{sg2}$ is searched after seventh Iteration.

**TABLE 4.13**
**Results of Rosenbrock function using AELDPSO-I**

| **Condition for implementation of $C_{sg2}$ (1)** | **Results (2)** | **k (3)** | **Kount (4)** | **$C_{sg2}$ (5)** |
|---|---|---|---|---|
| k > 2 | Failure | 1000 | 40040 | 0 |
| k > 3 | Success | 68 | 2760 | 0.3428 |
| k > 4 | Success | 54 | 2200 | 0.348 |
| k > 5 | Success | 62 | 2520 | 0.345 |
| k > 6 | Success | 64 | 2600 | 0.3443 |
| **k > 7** | Success | **44** | **1800** | **0.956** |

Iterations and its value is 0.956. The row corresponding to this has been highlighted. When this strategy is applied after eighth Iteration, no improvement has been observed in results.

Results for Mathematical benchmark Rosenbrock function using AELDPSO-II are shown in Table 4.14.

**TABLE 4.14**
**Results of Rosenbrock function for AELDPSO - II**

| **Condition for implementation of $C_{sg3}$ (1)** | **Result (2)** | **k (3)** | **Kount (4)** | **$C_{sg3}$ (5)** |
|---|---|---|---|---|
| k > 2 | Success | 33 | 1360 | 0.688 |
| k > 3 | Success | 33 | 1360 | 0.79 |
| k > 4 | Success | 36 | 1480 | 0.688 |
| k > 5 | Success | **30** | **1240** | **0.788** |
| k > 6 | Success | 32 | 1320 | 0.793 |
| k > 7 | Success | 40 | 1640 | 0.7154 |

Column (1) of Table 4.14 shows the number of Iterations after which $C_{sg3}$ (defined by equation (4.7) has been implemented. Column (2) shows the result of Rosenbrock function for defined condition of column (1). Column (3) and (4) represent number of Iterations and number of Kount required respectively to optimize the function. Column (5) shows value of $C_{sg3}$ searched for Rosenbrock function. When $C_{sg3}$ is implemented after five Iterations, Kount required to optimize the function is 1240 and an adaptive social acceleration constant value is 0.788 as shown in row (5) of Table 4.14. By observing column (4) of Table 4.14, minimum Kount is obtained when adaptive $C_{sg3}$ has been searched after five Iterations and its value is 0.788. It has been highlighted in the Table 4.14. When this strategy is applied after eighth Iteration, no improvement has been observed in results.

Comparison of results of ALDPSO, AELDPSO-I and II are shown in Table 4.15.

**TABLE 4.15**
**Comparison of Results of ALDPSO, AELDPSO-I and II**

| | **ALDPSO** | | **AELDPSO-I** | | **AELDPSO-II** | |
|---|---|---|---|---|---|---|
| **Condition for implementation of $C_{sg}$ (1)** | **$C_{sg1}$ (2)** | **Kount (3)** | **$C_{sg2}$ (4)** | **Kount (5)** | **$C_{sg3}$ (6)** | **Kount (7)** |
| k > 2 | 0.955 | 1840 | 0 | 40040 | 0.688 | 1360 |
| k > 3 | 0.954 | 1880 | 0.3428 | 2760 | 0.79 | 1360 |
| k > 4 | 0.958 | 1720 | 0.348 | 2200 | 0.688 | 1480 |
| **k > 5** | **0.96** | **1640** | **0.345** | **2520** | **0.788** | **1240** |
| k > 6 | 0.952 | 1960 | 0.3443 | 2600 | 0.793 | 1320 |
| k > 7 | 0.956 | 1800 | 0.956 | 1800 | 0.7154 | 1640 |

Column (1) of Table 4.15 represent the number of Iterations after which Csg was implemented. Columns (2), (4) and (6) represent the adaptive social acceleration constants $C_{sg1}$, $C_{sg2}$ and $C_{sg3}$ achieved by ALDPSO, AELDPSO-I, and AELDPSO-II respectively for each condition of implementation shown in column 1. Columns (3), (5) and (7) represent the Kount required to minimize the function. It was observed that AELDPSO-II requires less Kount as compared to ALDPSO and AELDPSO-I for all conditions of column (1). It was further observed from Table 4.15 for case of k > 5, minimum Kount are required by AELDPSO-II. For this condition of $C_{sg}$ implementation, the Kount required by three algorithms were:

ALDPSO : 1640

AELDPSO-I : 2520

AELDPSO - II: 1240

This row has been highlighted. The value of an adaptive social acceleration constant was achieved by AELDPSO-II and was taken as social acceleration constant for PSO which leads to the development of Adaptive Social Acceleration Constant based PSO (ASACPSO). The Experimentation has been done on following seven mathematical benchmark functions using Basic PSO, AELDPSO-II and Adaptive Social Acceleration Constant based PSO (ASACPSO).

1. Rosenbrock function
2. Beale Function
3. Booth Function
4. Hyper Ellipsoid Function
5. Ackley function
6. Schwefel function
7. Three Hump Function

Following parameters are considered for basic PSO (BPSO).

$r_p = C_g = r_g = C_p = 1$;     W = 0.6;     P= Population size = 40

These parameters have been selected merely for comparison of the results obtained by the proposed algorithms. Initially Basic PSO (BPSO) has been applied for 40 particles. Then the best social acceleration constant was searched using AELDPSO-II. This best value of social constant worked as $C_{sg}$ for ASACPSO algorithm. All the three algorithms BPSO, AELDPSO-II and ASACPSO Algorithm have been implemented on seven mathematical benchmark functions and their results have been compared in Table 4.16 to Table 4.22.

### 4.3.1 Rosenbrock Function

Mathematically, it is defined in equation number (4.1). Table 4.16 shows the results of Rosenbrock function using BPSO, AELDPSO-II and Adaptive Social Acceleration constant based PSO (ASACPSO).

**TABLE 4.16**
**Results for Rosenbrock function**

| PSOs (1) | $x_1$ (2) | $x_2$ (3) | f (4) | Cg/Csg (5) | Kount (6) | k (7) |
|---|---|---|---|---|---|---|
| **BPSO** | 0.99 | 0.99 | 1.9e-9 | 1 | 1600 | 39 |
| **AELDPSO-II** | 0.99 | 0.99 | 5.9e-10 | 0.788 | 1240 | 30 |
| **ASACPSO** | 0.99 | 0.99 | 1.79e-8 | 0.788 | 1320 | 32 |

Column (1) of Table 4.16 shows the type of PSO being implemented on Rosenbrock function. Columns (2) and (3) represent two variables of Rosenbrock function. Column (4) represents the corresponding function value. Column (5) represents Cg for BPSO and Csg for AELDPSO-II and ASACPSO algorithm. Columns (6) and (7) represents Kount and Iteration required respectively for optimizing the function.

### 4.3.2 Beale Function

Mathematically Beale function is defined as given in equation (4.2). Table 4.17 shows the results of Beale function for BPSO, AELDPSO-II and Adaptive Social Acceleration Constant based PSO (ASACPSO).

**TABLE 4.17**
**Results for Beale function**

| PSOs (1) | $x_1$ (2) | $x_2$ (3) | f (4) | $C_{sg}/C_{sg1}$ (5) | Kount (6) | k (7) |
|---|---|---|---|---|---|---|
| **BPSO** | 2.99 | 0.5 | 5.4e-10 | 1 | 1800 | 44 |
| **AELDPSO-II** | 3 | 0.5 | 9.42e-10 | 0.8071 | 1440 | 35 |
| **ASACPSO** | 3 | 0.49 | 1.1e-8 | 0.8071 | 1000 | 24 |

### 4.3.3 Booth Function

Mathematically, it is defined as given in equation (4.3). Table 4.18 shows the results of Booth function for BPSO, AELDPSO-II and Adaptive Social Acceleration constant based PSO (ASACPSO).

**TABLE 4.18**
**Results for Booth function**

| PSOs (1) | $x_1$ (2) | $x_2$ (3) | f (4) | Cg/Csg (5) | Kount (6) | k (7) |
|---|---|---|---|---|---|---|
| **BPSO** | 1 | 3 | 2.06e-13 | 1 | 2000 | 49 |
| **AELDPSO-II** | 1 | 2.99 | 5.9e-10 | 0.7358 | 1320 | 32 |
| **ASACPSO** | 0.99 | 2.99 | 6.32e-8 | 0.7962 | 1280 | 31 |

### 4.3.4 Axis Parallel Hyper Ellipsoid Function

Mathematically, it is defined as given in equation (4.4). Table 4.19 shows the results of Axis Parallel Hyper Ellipsoid function for BPSO, AELDPSO-II and Adaptive Social Acceleration constant based PSO (ASACPSO).

**TABLE 4.19**
**Results for Axis Parallel Hyper Ellipsoid function**

| PSOs (1) | $x_1$ (2) | $x_2$ (3) | f (4) | Cg/Csg (5) | Kount (6) | k (7) |
|---|---|---|---|---|---|---|
| **BPSO** | -1.165e-5 | -7.33e-9 | 3.8e-11 | 1 | 1880 | 46 |
| **AELDPSO-II** | -2.5e-6 | -6.12e-6 | 8.15e-12 | 0.669 | 1680 | 41 |
| **ASACPSO** | -1.3e-5 | -6.3e-9 | 1.8e-19 | 0.669 | 1160 | 28 |

### 4.3.5 Ackley Function

Mathematically, it is defined as

$$f(x_1, x_2) = -20 * e^{(-0.2\sqrt{(0.5(x_1^2)+x_2^2)})} \quad (4.8)$$

$f(0,0) = 0$ Range $-5.0 \le x_i \le 5.0$

Table 4.20 show the results of Ackley function for BPSO, AELDPSO-II and Adaptive Social Acceleration Constant based PSO (ASACPSO).

**TABLE 4.20**
**Results for Ackley function**

| PSOs (1) | $x_1$ (2) | $x_2$ (3) | f (4) | Cg/Csg (5) | Kount (6) | k (7) |
|---|---|---|---|---|---|---|
| BPSO | -1.64e-11 | -8.15e-12 | 5.204e-11 | 1 | 3920 | 97 |
| AELDPSO-II | -1.40e-9 | 1.286e-9 | 5.3902e-9 | 0.725 | 2360 | 58 |
| ASACPSO | -1.64e-16 | -2.692e-16 | 1.88e-15 | 0.725 | 2240 | 55 |

### 4.3.6 Schwefel Function

Mathematically, it is defined as

$$f(x) = 418.9829 - \mathbf{x_1} * \sin(\sqrt{\mathrm{abs}}(\mathbf{x_1})) \quad (4.13)$$

$f(x) = 0$; at $x = (420.9687)$

Range  $-500.0 \le x_i \le 500.0$

Table 4.21 show the results of  Schwefel function for BPSO, AELDPSO-II, Adaptive Social Acceleration constant based PSO (ASACPSO).

**TABLE 4.21**
**Results for Schwefel Function**

| PSOs (1) | $x_1$ (2) | f (3) | Cg/Csg (4) | Kount (5) | k (6) |
|---|---|---|---|---|---|
| BPSO | 420.968 | 1.222e-5 | 1 | 3600 | 89 |
| AELDPSO | 420.965 | 1.4392e-5 | 0.7242 | 40 | 51 |
| ASACPSO | 420.963 | 1.27e-5 | 0.7242 | 40 | 53 |

## 4.3.7 Three Hump Camel Function

Mathematically, it is defined as

$$f(x_1, x_2) = 2*x_1^2 - 1.05*x_1^4 + x_1^6/6 + x_1 * x_2 + x_2^2 \tag{4.14}$$

$f(0,0) = 0$

Table 4.22 shows the results of Three Hump Camel function for BPSO, AELDPSO-II and Adaptive Social Acceleration Constant based PSO (ASACPSO).

**TABLE 4.22**
**Results for Three Hump Camel function**

| PSOs (1) | $x_1$ (2) | $x_2$ (3) | f (4) | Cg/Csg (5) | Kount (6) | k (7) |
|---|---|---|---|---|---|---|
| BPSO | -1.8062e-6 | 2.20e-6 | 7.40e-12 | 1 | 1800 | 44 |
| AELDPSO | -2.236e-6 | 1.42e-5 | 1.82e-10 | 0.719 | 1320 | 32 |
| ASACPSO | -2.1779e-6 | -4.07e-7 | 1.27e-5 | 0.719 | 1000 | 24 |

Comparison of Kount for all the functions optimized using BPSO, AELDPSO-II and ASACPSO algorithms are shown in Table 4.23.

**TABLE 4.23**
**Comparison of Kount for mathematical functions**

| S.No. (1) | Functions (2) | BPSO (Kount) (3) | AELDPSO-II (Kount) (4) | ASACPSO (Kount) (5) |
|---|---|---|---|---|
| 1 | Rosenbrock | 1600 | 1240 | 1320 |
| 2 | Beale | 1800 | 1440 | 1000 |
| 3 | Booth | 2000 | 1320 | 1280 |
| 4 | Axis Parallel Hyper Ellipsoid | 1880 | 1680 | 1160 |
| 5 | Ackley Function | 3920 | 2360 | 2240 |
| 6 | Schwefel function | 3600 | 40 | 40 |
| 7 | Three Hump Camel function | 1800 | 1320 | 1000 |

It is observed that minimum Kount are required for ASACPSO Algorithm except in case of Rosenbrock function, where Kount required for AELDPSO-II is minimum.

Table 4.24 shows the % saving in Kount for ASACPSO Algorithm and BPSO:

**TABLE 4.24**
**Results of BPSO & ASACPSO ALGORITHM and % saving in Kount**

| S. No. (1) | Functions (2) | BPSO (3) | ASACPSO (4) | % Saving in Kount (5) |
|---|---|---|---|---|
| 1 | Rosenbrock | 1600 | 1320 | 17.5 |
| 2 | Beale | 1800 | 1000 | 44.5 |
| 3 | Booth | 2000 | 1280 | 36 |
| 4 | Axis parallel Hyper Ellipsoid | 1880 | 1160 | 38.29 |
| 5 | Ackley | 3920 | 2240 | 42.85 |
| 6 | Schwefel | 3600 | 40 | 97.8 |
| 7 | Three Hump Camel function | 1800 | 1000 | 44.5 |

It has been observed that every function converged for lesser Iterations and lesser Kount for ASACPSO Algorithm in comparison to basic PSO (BPSO). Minimum saving in

Kount for Rosenbrock function and maximum saving in Kount for Schwefel function has been achieved as shown in Table 4.24. A saving of 44.5% has been achieved in Beale and Three Hump Camel functions. The saving has been calculated using following formula: Saving in Kount % = [100*(Kount for 40 particles (with $C_g$=1) – Kount for 40 particles with $C_{sg3}$] / Kount for 40 particles(with $C_g$=1).Comparison of Kount for different functions for BPSO and ASACPSO are shown in Fig. 4.4.

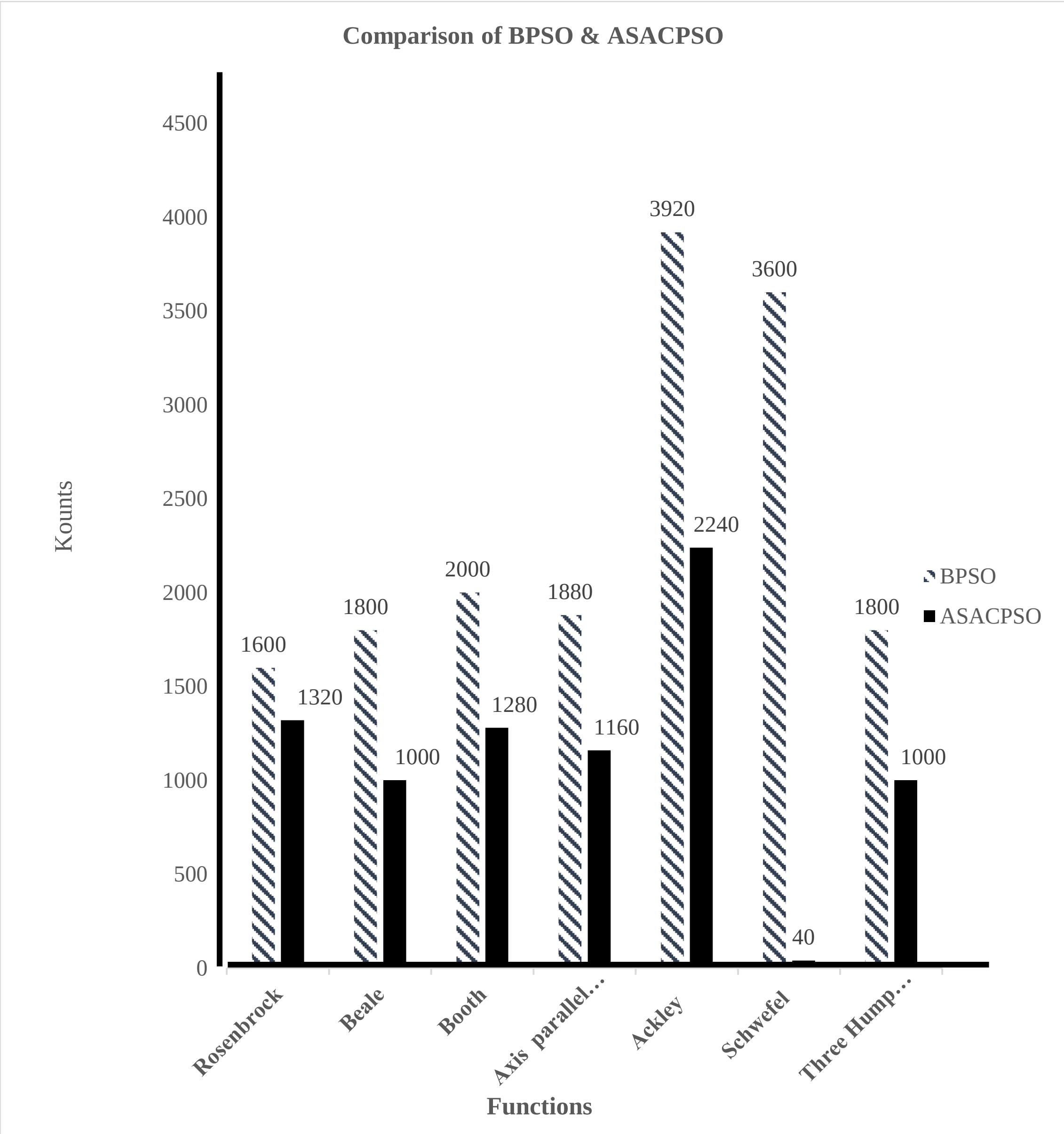



**Fig. 4.3** Comparison of Kount for different functions using BPSO & ASACPSO Algorithms.

## 4.4 ECONOMIC LOAD DISPATCH (ELD)

Thermal power plants (coal, natural gas, oil and nuclear) are responsible for roughly 80% of global electricity production. Commercial power stations are usually constructed on a large scale and designed for continuous operation. The direct cost of electric energy produced by a thermal power station is the result of cost of fuel, capital cost for the plant, operator labor, maintenance, and such factors as ash handling and disposal. The sizes of the electric power system are increasing rapidly to meet the energy requirement. Therefore, the number of power plants are connected in parallel to supply the system load by an interconnection of the power system. In the grid system, it becomes necessary to operate the plant units more economically. Economic Load Dispatch (ELD) problem allocates loads to plants at a minimum cost while satisfying the equality and inequality constraints [5, 30, 92].

Mathematically, ELD problem is defined as:

Minimize

$$F_C = \sum_{i=1}^{NG} F[C_i(P_{gi})] \tag{4.15}$$

Where cost of generation is defined as:

$$C_i(P_{gi}) = \sum_{i=1}^{NG} (a_i P_{gi}^2 + b_i P_{gi} + c_i) \tag{4.16}$$

The system transmission losses are defined as:

$$F_L = \sum_{i=1}^{NG} \sum_{j=1}^{NG} P_{gi} B_{ij} P_{gj} \tag{4.17}$$

Subject to the constraints

Equality constraint

$$\sum_{i=1}^{NG} P_{gi} = P_D + F_L \tag{4.18}$$

Inequality constraint

$$P_{gimin} \leq P_{gi} \leq P_{gimax} \qquad i = 1,2 \dots NG \qquad (4.19)$$

To consider the equality constraint of the problem, the function has been modified by inclusion of a parameter K. The objective function becomes as follows:

$$F = F_C + K\,(P_D + F_L - P_{gi}) \qquad (4.20)$$

Where parameter K is fixed at 1000 for all three IEEE 5, 14, and 30 bus systems. Different values of K were considered and it was observed that ELD problem converged when it was fixed to 1000 for IEEE 5, 14 and 30 bus systems. The optimum solution is achieved when the change in function value during successive Iterations is less than the limit specified which is T = $1 * 10^{-6}$ and the absolute value of difference between generation, demand and losses is less than $1 * 10^{-6}$.

### 4.4.1 Application of BPSO and Improved Particle Swarm Optimization based on Initial Selection of Particles (IPSO IS) to ELD problem

The Experimentation has been done on the following Systems

**1**. IEEE 5 Bus System

2. IEEE 14 Bus System

3. IEEE 30 Bus System

## 4.5 COMPUTATIONAL PROCEDURE

The Computational procedures for BPSO and IPSO IS have been discussed below for Economic Load Dispatch Problem:

## (a) BPSO

The sequence for the solution of Economic Load Dispatch problem using Basic Particle Swarm Optimization (BPSO) algorithm is explained as follows:

1. Fix the number of particles "q" as population size and set the number of maximum Iterations $IT_{max}$ and tolerance value T = $1 * 10^{-6}$ .
2. Fix the cost coefficients, loss coefficients, load demand and generator limits of all the generators.
3. Set function evaluation count, Kount = 0.
4. Generate initial random positions and velocities for all particles.
5. Calculate the losses for each particle using (4.17).
6. Calculate the cost of generation using (4.15).
7. At $0^{th}$ Iteration the personal best position is same as the initial random positions and minimum value of pbest is the gbest value.
8. Increase the count value 'Kount' by 1 using Kount = Kount + 1.
9. Calculate the velocity of each particle using (3.21).
10. Check if velocity is within the limits. If not, fix the velocity to the limit violated.
11. Calculate the new positions of the particles by evaluating (3.22).
12. Calculate ELD for the new positions generated at $11^{th}$ step.
13. Update $x_{pbest}$ and $x_{gbest}$ values by comparing ELD function values.
14. Check if both the stopping criteria are satisfied, if not, then go to step 8, else go to 15.
15. Display the optimize values for cost of generation, system transmission losses and numbers of Kount.

## (b) IPSO IS

Steps required for IPSO IS for ELD problem are as follows:

1. Initialize the particles less than Pmin. Pmin is minimum number of particles for which ELD is optimized by BPSO.

2. Select the best particles according to minimum function value.
3. Pmin is used for selecting the initial population "q" for IPSO IS. Where q ≥ Pmin.
4. Particles are selected from q, where P is selected for which P ≤ Pmin, and for which function optimized.
5. Perform ELD with selected particles P.

## 4.6 COMPUTATIONAL RESULTS AND DISCUSSION

Improved Particle Swarm Optimization based on Initial Selection of Particles (IPSO IS) has been implemented for Economic Load Dispatch of IEEE 5 Bus, IEEE 14 Bus and IEEE 30 Bus Systems.

### 4.6.1 IEEE 5 Bus System

Following parameters are considered for basic PSO (BPSO) and Improved PSO (IPSO IS) for IEEE 5,14 and 30 bus systems.

$r_p = C_g = r_g = C_p = 1$, W = 0.6, $IT_{max} = 1000$ , $T = 1 * 10^{-6}$

Initially Basic PSO (BPSO) has been applied to solve ELD of IEEE 5 Bus System by varying number of particles from 10 to 100 in step of 10. The results are shown in Table 4.25.

**TABLE 4.25**
**Results of ELD using BPSO for IEEE 5 Bus System**

| S. No. (1) | q (2) | Cost (\$/hr.) (3) | Loss ( $F_L$ ) (MW) (4) | k (5) | Kount (6) | Remarks (7) |
|---|---|---|---|---|---|---|
| 1 | 10 | 761.155 | 5.2037 | 1000 | 20020 | Unsuccessful |
| 2 | 20 | 761.135 | 5.2252 | 114 | 4600 | Successful |
| 3 | 30 | 761.135 | 5.2238 | 114 | 6900 | Successful |
| 4 | 40 | 761.353 | 5.2262 | 131 | 10560 | Successful |
| 5 | 50 | 761.353 | 5.2255 | 125 | 12600 | Successful |
| 6 | 60 | 761.352 | 5.2246 | 108 | 13080 | Successful |
| 7 | 70 | 761.383 | 5.2159 | 137 | 19320 | Successful |
| 8 | 80 | 761.135 | 5.2200 | 88 | 14240 | Successful |
| 9 | 90 | 761.135 | 5.2200 | 89 | 16200 | Successful |
| 10 | 100 | 761.135 | 5.2200 | 88 | 17800 | Successful |
| 11 | 8 | | | 1000 | | Unsuccessful |

The second column of Table 4.25 shows the number of particles. The third and fourth columns show the cost of generation and transmission losses of the system. The fifth and sixth columns show the number of Iterations (k) and Kount required to optimize the function. Seventh column shows the remarks for function convergence.

**Kount** in this case is number of function evaluations, i.e. how many times a function has been evaluated. This is taken as the measure of computational time. The measure of computational time in seconds or minutes will depend on the technology, i.e. the type of computer being used. Therefore, the time has been measured in terms of function evaluations. The results of Table 4.25 show that the Kount increases as the number of particles increase. Table 4.25 shows that the Basic PSO could not optimize the function for q = 10. The minimum value of particles for which function get optimized is 20, so Pmin = 20.

An Improved PSO (IPSO IS) based on initial selection of particles has been applied to IEEE 5 Bus System. In IPSO IS, it is experimented to choose better particles "P" from the initially generated "q" particles. Better particles have been obtained based on the function value. In improved selection based PSO better particles "P ≤ Pmin" are selected from the initially generated "q ≥ Pmin" particles . In IPSO IS "q" are the initial particles based on "q ≥ Pmin" and selected particles are "P" based on "P ≤ Pmin" respectively shown in column 2 and 3 of Table 4.26. The best results in terms of minimum Kount for each P and corresponding q for which function optimized have been shown in Table 4.26.

**TABLE 4.26**
**Results of ELD using IPSO IS for "q ≤ Pmin"**

| S. No. (1) | q (2) | P (3) | k (4) | Cost($F_C$) (5) | Loss ($F_L$) (6) | Kount(S) (7) | Kount(N) (8) |
|---|---|---|---|---|---|---|---|
| 1 | 20 | 8 | 108 | 761.13 | 5.127 | 1756 | 4600 |

In Table 4.26 Column (2) and (3) represent the initial particles and selected Particles. Column (4) shows the Iteration required to optimize the function. Column (5) and (6) show the cost of function and system Loss. Kount (S) and Kount(N) in column (7) and column (8) respectively represent Kount required to optimize the function using IPSO IS and BPSO. It is observed that Kount (S) is much less than Kount (N). It is also observed that for P = 8, ELD could not be optimized using BPSO for IEEE 5 bus system . Whereas using IPSO IS, the said function could be optimized.

### 4.6.2 IEEE 14 Bus System

Initially BPSO has been implemented for ELD of IEEE 14 Bus System by varying number of particles from 10 to 100 in step of 10. The results are shown in Table 4.27.

**TABLE 4.27**
**Results of ELD using BPSO for IEEE 14 Bus System**

| S. No. (1) | q (2) | Cost ($F_C$) (3) | Loss ($F_L$) (4) | k (5) | Kount (6) | Remarks (7) |
|---|---|---|---|---|---|---|
| 1 | 10 | x | x | 1000 | x | Unsuccessful |
| 2 | 20 | x | x | 1000 | x | Unsuccessful |
| 3 | 30 | x | x | 1000 | x | Unsuccessful |
| 4 | 40 | 1143.853 | 10.27687 | 133 | 5400 | Successful |
| 5 | 50 | 1143.853 | 10.27311 | 103 | 5250 | Successful |
| 6 | 60 | 1144.037 | 10.07383 | 171 | 10380 | Successful |
| 7 | 70 | 1143.854 | 10.28487 | 132 | 9380 | Successful |
| 8 | 80 | 1143.853 | 10.27293 | 100 | 8160 | Successful |
| 9 | 90 | 1143.854 | 10.27294 | 92 | 8460 | Successful |
| 10 | 100 | 1143.854 | 10.27844 | 137 | 13900 | Successful |

It is observed from Table 4.27 that ELD for IEEE 14 bus system could not be performed for 10 particles using BPSO. Here we observed that minimum value of P i.e. Pmin for which IEEE 14 Bus function could be optimized is 40. Now IPSO IS has been implemented and results are shown in Table 4.28.

**TABLE 4.28**
**Results of ELD using IPSO IS for "q ≥ Pmin" and " P ≤ pmin"**

| S. No. (1) | q (2) | P (3) | Cost ($F_C$) (4) | Loss ($F_L$) (5) | k (6) | Kount(S) (7) | Kount(N) (8) |
|---|---|---|---|---|---|---|---|
| 1 | 40 | 10 | 1145.263 | 9.407 | 162 | 1670 | 5400 |
| 2 | 60 | 10 | 1146.776 | 8.897 | 139 | 1460 | 13080 |
| 3 | 70 | 10 | 1147.434 | 8.772 | 114 | 1220 | 19320 |
| 4 | 80 | 10 | 1147.673 | 8.72 | 147 | 1560 | 14240 |
| 5 | 40 | 30 | 1143.864 | 10.183 | 151 | 4600 | 5400 |
| 6 | 60 | 20 | 1144.582 | 9.559 | 153 | 3140 | 13080 |
| 7 | 70 | 20 | 1144.625 | 9.535 | 157 | 3230 | 19320 |
| 9 | 90 | 20 | 1144.571 | 9.554 | 156 | 3230 | 16200 |

It is observed from results of Table 4.28 that ELD for IEEE 14 bus system could be performed successfully by 10 particles which is less than Pmin.

## 4.6.3 IEEE 30 Bus System

Initially BPSO has been implemented for ELD of IEEE 30 Bus System by varying number of particles from 5 to 100 in step of 10. The results are shown in Table 4.29.

**TABLE 4.29**
**Results of ELD using BPSO for IEEE 30 Bus System**

| S. No. (1) | P (2) | Cost ($F_C$) (3) | Loss ($F_L$) (4) | k (5) | Kount (N) (6) | Remarks (7) |
|---|---|---|---|---|---|---|
| **1** | 5 | x | x | 1000 | - | x |
| **2** | 10 | x | x | 1000 | - | x |
| **3** | 20 | 1256.200 | 12.41 | 133 | 5360 | Successful |
| **5** | 30 | 1256.508 | 12.86 | 162 | 9780 | Successful |
| **4** | 40 | 1256.387 | 12.35 | 149 | 12000 | Successful |
| **5** | 50 | 1256.471 | 12.88 | 165 | 16600 | Successful |
| **6** | 60 | 1256.200 | 12.40 | 136 | 16440 | Successful |
| **7** | 70 | 1256.192 | 12.37 | 126 | 17780 | Successful |
| **8** | 80 | 1256.192 | 12.37 | 134 | 21600 | Successful |
| **9** | 90 | 1256.192 | 12.37 | 106 | 19260 | Successful |
| **10** | 100 | 1256.230 | 12.42 | 152 | 30600 | Successful |

Here we observed that minimum value of Pmin for which ELD could be performed for IEEE 30 Bus system is 20. Now IPSO IS is implemented on IEEE 30 bus systems. Results are shown in Table 4.30.

**TABLE 4.30**
**ELD using IPSO IS for "q ≥ Pmin" and "P ≤ Pmin"**
**(IEEE 30 Bus System)**

| S. No. (1) | q (2) | P (3) | Cost ($F_C$) (4) | Loss ($F_L$) (5) | k (6) | Kount(S) (7) | Kount(N) (8) |
|---|---|---|---|---|---|---|---|
| 1 | 40 | 10 | 1256.688 | 11.74 | 151 | 3070 | 12000 |
| 2 | 60 | 10 | 1257.102 | 11.51 | 154 | 3150 | 16440 |
| 3 | 80 | 10 | 1257.896 | 11.33 | 152 | 3130 | 21600 |
| 4 | 90 | 10 | 1259.369 | 10.87 | 136 | 2820 | 19260 |
| 5 | 50 | 8 | 1257.929 | 11.21 | 148 | 2426 | 16600 |

The results of Table 4.29 show that for q = 10 the function could not be optimized using BPSO. Whereas using IPSO IS, the said function could be optimized for 10 and 8 particles i.e. for particles "q ≥ Pmin" as shown in Table 4.30.

Minimum particle size Pmin for which ELD could be performed successfully for IEEE 5, 14 and 30 bus systems by BPSO is shown in Table 4.31.

**TABLE 4.31**
**Pmin for ELD using BPSO**

| S.No | SYSTEM | Pmin | k | Kount |
|---|---|---|---|---|
| **1** | **IEEE 5 BUS** | 20 | 114 | 4600 |
| **2** | **IEEE 14 BUS** | 40 | 133 | 5400 |
| **3** | **IEEE 30 BUS** | 20 | 133 | 5360 |

Various combinations of 'P' and 'q' have been tried for all the systems. Table 4.32 shows the comparison of Kount for IEEE 5, 14 and 30 bus systems for performing ELD using BPSO and IPSO IS.

**TABLE 4.32**
**Comparison of Kount for BPSO and IPSO IS**
**(IEEE 5, 14 and 30 Bus Systems)**

| S. No. (1) | SYSTEM (2) | q (3) | P < Pmin (4) | Kount (S) (5) | Kount (N) (6) | % Saving in Kount (7) |
|---|---|---|---|---|---|---|
| 1 | IEEE 5 BUS | 20 | 8 | 1756 | 4600 | 61.82% |
| 2 | IEEE 14 BUS | 40 | 10 | 1670 | 5400 | 69.07% |
| 3 | IEEE 30 BUS | 50 | 8 | 2426 | 16600 | 85.38% |

Table 4.32 shows the effect of Improved PSO (IPSO IS) based on initial selection of particles. It is observed that, for every system, ELD could be carried out for particles i.e. Pmin. In order to reduce the Kount, the particles have been selected out of double or more than double of this size i.e. for IEEE 5 bus system, ELD using Basic PSO (BPSO) required 20 minimum particles to converge, whereas by using Improved PSO (IPSO IS) and selecting 8 particles from 20 initial particles gave a saving of 61.82% in terms of Kount. For IEEE 14 bus system, ELD could be performed for P < Pmin i.e. by selecting 10 out of 40 particles which resulted in saving of 69.07%. Similarly, a saving of 85.38% in Kount is achieved for IEEE 30 bus system.

The saving has been calculated using the formula given below:

Saving in Kount % = 100 * (Kount (N) – Kount (S)) / Kount (N).

It has been observed that the IEEE 5, 14 and 30 bus systems ELD could be performed successfully for small particles size of 8, 10 and 8 respectively by IPSO IS as shown in Table 4.32.

## 4.7 ELD USING ADAPTIVE SOCIAL ACCELERATION CONSTANT BASED PARTICLE SWARM OPTIMIZATION BASED PSO (ASACPSO)

Adaptive Social Acceleration constant based Particle Swarm Optimization has been explained in section 4.3 and implemented on seven benchmark mathematical functions successfully. In this section, ASACPSO has been implemented for ELD of IEEE 5, 14 and 30 bus systems. PSOs based on $C_{sg1}$, $C_{sg2}$, $C_{sg3}$ have been named as Adaptive Linearly Decreasing PSO (ALDPSO), Adaptive Exponential Linearly Decreasing PSO (AELDPSO-I) and Adaptive Exponential Linearly Decreasing PSO-II (AELDPSO-II) respectively. All the three algorithms have been implemented to solve Economic Load Dispatch problem of IEEE 5, 14, and 30 bus systems. The best value of Adaptive–Social Acceleration Constant $C_{sg}$ for each algorithm is searched by applying Eq. (4.5), (4.6) and (4.7) after second, third, fourth and up to seventh Iteration. The convergence was not achieved when $C_{sg}$ was implemented after 7th Iteration. The results of ALDPSO, AELDPSO-I and AELDPSO-II for IEEE 5 bus system are shown in Tables 4.33, 4.34, and 4.35 respectively. The best value of $C_{sg}$ is selected based on minimum number of Kount required to perform ELD. Adaptive Social Acceleration Constant based PSO (ASACPSO) is then developed using the best value of $C_{sg}$.

### 4.7.1 Computational procedure

The basic PSO (BPSO) and proposed PSO algorithms–ALDPSO, AELDPSO-I, AELDPSO-II, ASACPSO have been implemented for Economic Load Dispatch of IEEE 5-bus, 14-bus and 30-bus systems. For all the algorithms the parameters were fixed as:

q = 40; $r_p = C_g = r_g = C_p = 1$; W=0.6; $C_{gmax}$ =1.5; $C_{gmin}$= 0.4; $IT_{max} = 1000$

The sequence for the solution of Economic Load Dispatch problem using ASACPSO algorithm are explained as follows:

1. Initialize the population between the limits of power.
2. Initialize the velocities of particles between the limits.
3. Fix the number of maximum Iterations $IT_{max}$, tolerance value T and penalty parameter

K. For optimum solution the change in the function value during successive Iterations must be less than the limit specified which is T = $10^{-6}$.

4. To meet the constraint the absolute value of difference between generation, demand and losses must be less than T = $1 * 10^{-6}$.
5. Set Iteration count k = 0 and Kount = 0.
6. Fix the Cost coefficients, Loss coefficients, load demand and calculate the Cost of generation using equation (4.16) and Loss using equation (4.17) for each particle.
7. Evaluate function value for each particle using (4.20).
8. At zeroth Iteration, the personal best positions are same as the initial random positions. Global best value is the lowest function value.
9. Increase the Iteration value ‘k’ and Kount by 1.
10. Calculate the velocity of each particle by replacing $C_g$ by $C_{sg}$ in (3.21).
11. Check if velocity is within the limits. If not fix the velocity to the limit violated.
12. Update the new positions of the particles using (3.22).
13. Perform ELD using (4.20) for the new position of particles.
14. Update $x_{pbest}$ and $x_{gbest}$, values.
15. Check if both the stopping criteria are satisfied, then go to 16. If not, then go to 9.
16. Display cost of generation, transmission losses, Iterations and numbers of Kount required to optimize the function.

### 4.7.2 Computational Results and Discussion

The Results of ELD using ALDPSO, AELDPSO-I and AELDPSO-II for IEEE 5 bus system are shown in Tables 4.33, 4.34 and 4.35 respectively.

**TABLE 4.33**
**Results of ELD using ALDPSO**
**(IEEE 5 Bus System)**

| **Condition for Implementation of $C_{sg1}$ (1)** | **$C_{sg1}$ (2)** | **$F_C$ ($/hr) (3)** | **$F_L$ (MW) (4)** | **Kount (5)** | **k (6)** | **$P_1$ (MW) (7)** | **$P_2$ (MW) (8)** |
|---|---|---|---|---|---|---|---|
| k > 2 | 1.37 | 761.63 | 5.08 | 9080 | 113 | 90.08 | 74.99 |
| k > 3 | 1.37 | 762.29 | 5.06 | 8920 | 111 | 87.19 | 77.87 |
| k > 4 | **1. 384** | **761.14** | **5.09** | **8440** | **105** | **91.18** | **73.91** |
| k > 5 | x | | | | | | |
| k > 6 | 1.363 | 761.27 | 5.1 | 9960 | 124 | 92.34 | 72.76 |

Column (1) of Table 4.33 represents condition for implementation of $C_{sg1}$. Column (2) shows the searched value of $C_{sg1}$. Column (3) and (4) shows the Cost of generation and system transmission losses of IEEE 5 Bus Systems. Column (5) and (6) represents the number of function evaluation (Kount) and number of Iterations required to optimize the function. Column (7) and (8) shows the output of generator (1) and (2) respectively. $C_{sg1}$ is implemented after 2nd to 6th Iteration. When $C_{sg1}$ is implemented after $5^{th}$ Iteration then function is not optimized. Minimum cost is obtained in minimum number of Kounts when $C_{sg1}$ is implemented after 4th Iteration. This row has been highlighted in Table 4.33.

Results of AELDPSO-I, searched by $C_{sg2}$ is shown in Table 4.34. $C_{sg2}$ is implemented after 2nd to 7th Iteration. It is observed that for this algorithm the results are obtained only for the case when $C_{sg2}$ is implemented after $4^{th}$ Iteration. Therefore, this algorithm has not been implemented on IEEE 14 and 30 Bus systems.

**TABLE 4.34**
**Results of ELD using AELDPSO-I**
**(IEEE 5 Bus System)**

| **Condition for implementation of $C_{sg2}$ (1)** | **$C_{sg2}$ (2)** | **$F_C$ ($/h) (3)** | **$F_L$ (MW) (4)** | **Kount (5)** | **k (6)** | **$P_1$ (7)** | **$P_2$ (8)** |
|---|---|---|---|---|---|---|---|
| k > 2 | x | | | | | | |
| k > 3 | x | | | | | | |
| k > 4 | 0.6703 | 762.57 | 5.06 | 8480 | 105 | 86.20 | 78.85 |
| k > 5 | x | | | | | | |
| k > 6 | x | | | | | | |
| k > 7 | x | | | | | | |

Table 4.35 shows the results of AELDPSO-II.

**TABLE 4.35**
**Results of ELD using AELDPSO-II**
**(IEEE 5 Bus System)**

| **Condition for implementation of $C_{sg3}$ (1)** | **$C_{sg3}$ (2)** | **$F_C$ ($/h) (3)** | **$F_L$ (MW) (4)** | **Kount (5)** | **k (6)** | **$P_1$ (7)** | **$P_2$ (8)** |
|---|---|---|---|---|---|---|---|
| k > 2 | 0.617 | 762.96 | 5.52 | 6040 | 75 | 109.65 | 55.87 |
| k > 3 | 0.616 | 761.03 | 5.06 | 5800 | 72 | 99.59 | 65.38. |
| k > 4 | | | x | x | x | x | |
| k > 5 | 0.615 | 762.16 | 5.06 | 5480 | 68 | 87.70 | 77.36 |
| k > 6 | **0.616** | **761.50** | **5.08** | **5560** | **69** | **90.83** | **74.25** |
| k > 7 | x | x | 5.10 | x | x | | |

Table 4.35 shows the results of ELD using AELDPSO-II. The best result is obtained when $C_{sg3}$ has been implemented for k > 6. After 5$^{th}$ Iteration, k > 5 we get minimum Kount for function optimization but cost of generation is higher in comparison to that obtained for k > 6, therefore, results of k > 6 have been considered.

From the results of Table 4.35 the best value of $C_{sg3}$ has been found to 0.616. Keeping this value of $C_{sg3}$ in PSO algorithm results in development of ASACPSO.

ASACPSO algorithm is implemented on IEEE 5 Bus System. The results are shown in Table 4.36.

**TABLE 4.36**
**Results of ELD using ASACPSO**
**(IEEE 5 Bus System)**

| $F_C$ (\$/h) | $F_L$(MW) | Kount | k | $C_{sg}$ | $P_1$ | $P_2$ |
|---|---|---|---|---|---|---|
| 761.37 | 5.09 | 5480 | 68 | 0.616 | 91.65 | 73.44 |

Results of BPSO, AELDPSO-II, and ASACPSO for IEEE 5 bus system are compared in Table 4.37.

**TABLE 4.37**
**Results of ELD using BPSO, AELDPSO-II and ASACPSO**
**(IEEE 5 Bus System)**

| S. No. (1) | PSOs (2) | $F_C$ (\$/h) (3) | $F_L$ (MW) (4) | Kount (5) | %Saving in Kount (6) |
|---|---|---|---|---|---|
| 1 | BPSO | 761.43 | 5.093 | 8360 | NA |
| 2 | AELDPSO-II | 761.03 | 5.06 | 5800 | 30.62 |
| 3 | ASACPSO | 761.37 | 5.09 | 5480 | 34.44 |

Table 4.37 shows the results of BPSO, AELDPSO-II and ASACPSO for IEEE 5 bus system. It is observed that ASACPSO converges faster than BPSO and AELDPSO-II and results in saving of 34.44% in Kount as compared to BPSO. Figures 4.4 and 4.5 show the comparison in terms of Kount and cost of generation respectively as obtained by BPSO, AELDPSO-II and ASACPSO for IEEE 5- bus system.

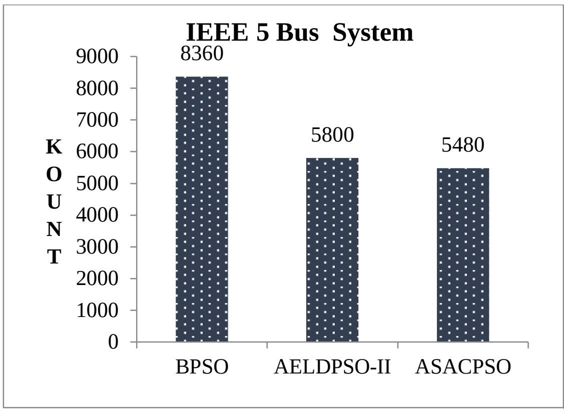



**Fig. 4.4** Comparison in terms of Kount

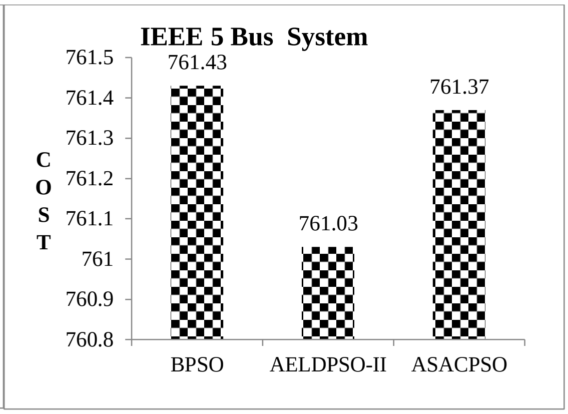



**Fig. 4.5** Comparison in terms of ($F_C$) Cost

It is clearly seen from Fig. 4.4 that both the PSO algorithms – AELDPSO-II and ASACPSO require less number of Kounts than BPSO. Fig. 4.5 shows that cost of generation is also less for both the algorithms compared to BPSO.

Economic Load Dispatch has also been carried out for IEEE 14 bus and IEEE 30 bus system using BPSO, AELDPSO-II and ASACPSO algorithms. Results of IEEE 14 and IEEE 30 bus systems are shown in Tables 4.38 and 4.39 respectively.

**TABLE 4.38**
**RESULTS of ELD using BPSO, AELDPSO-II and ASACPSO**
**(IEEE 14 Bus System)**

| PSOs | $F_C$ (\$/hr.) | $F_L$ (MW) | Kount | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|---|---|---|
| **BPSO** | 1145.138 | 7.938 | 4800 | 126.89 | 78.99 | 60.86 |
| **AELDPSO-II** | 1143.313 | 7.834 | 3080 | 131.03 | 74.75 | 61.04 |
| **ASACPSO** | 1136.062 | 8.6115 | 3520 | 151.58 | 70.35 | 45.67 |

Fig. 4.6 and Fig. 4.7 show the comparison in terms of Kount and cost of generation respectively for BPSO, AELDPSO-II and ASACPSO respectively for IEEE 14 bus system.

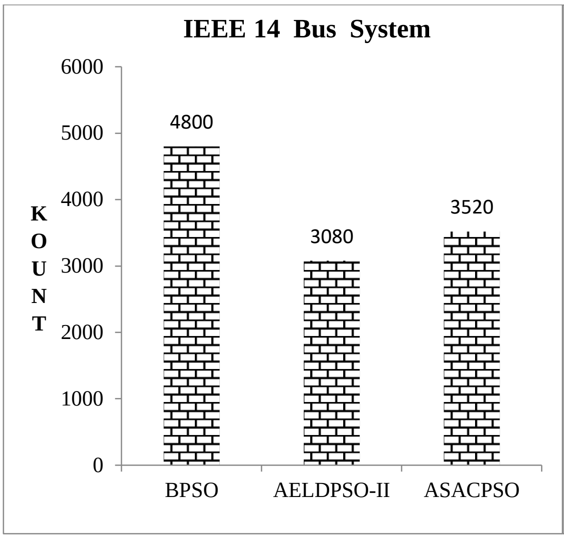



**Fig. 4.6** Comparison in terms of Kount

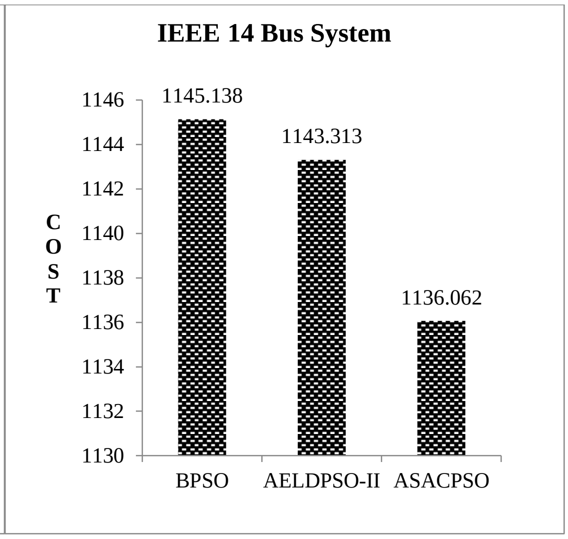



**Fig.4.7** Comparison of Cost of Generation ($F_C$)

## TABLE 4. 39
## Results of ELD using BPSO, AELDPSO-II and ASACPSO (IEEE 30 Bus System)

| PSOs | $F_C$ (\$/hr.) | $F_L$ (MW) | Kount | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|---|---|---|
| **BPSO** | 1266.97 | 11.05 | 5240 | 154.3 | 49.26 | 90.85 |
| **AELDPSO-II** | 1258.99 | 10.94 | 3040 | 140.92 | 83.36 | 70.06 |
| **ASACPSO** | 1258.66 | 10.93 | 3040 | 140.91 | 83.36 | 70.06 |

Fig.4.8 and 4.9 show the comparison in terms of Kount and cost of generation respectively as obtained by BPSO, AELDPSO-II and ASACPSO for IEEE 30 bus system.

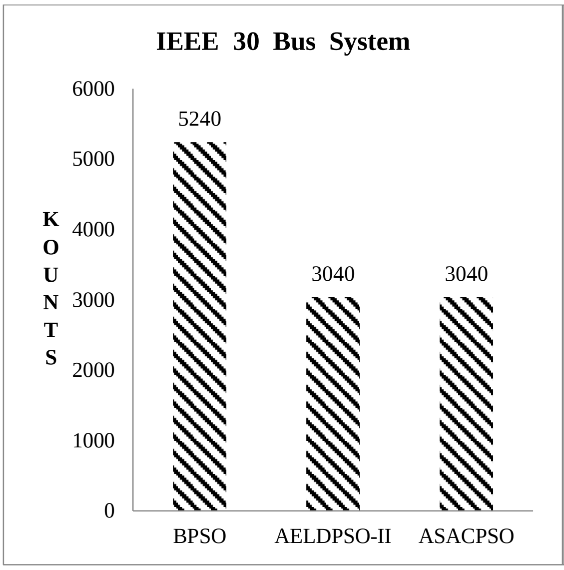



**Fig.4.8** Comparison in terms of Kount

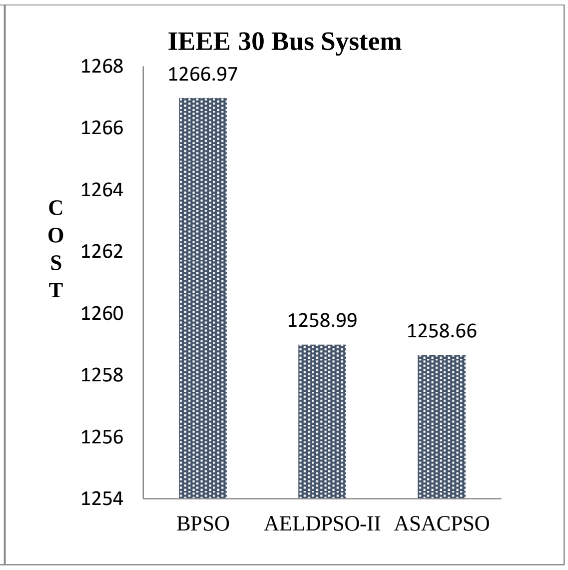



**Fig. 4.9** Comparison in terms of ($F_C$) Cost

Table 4.40 shows the percentage saving in Kount obtained by AELDPSO-II and ASACPSO for IEEE 5, 14 and 30 bus systems.

**TABLE 4.40**
**Saving in Kount for IEEE 5, 14 and 30 BUS Systems**

| Systems | AELDPSO-II | ASACPSO |
|---|---|---|
| **IEEE 5 Bus System** | 30.62% | 34.44% |
| **IEEE 14 Bus System** | 35.83% | 26.66% |
| **IEEE 30 Bus System** | 41.98% | 41.98% |

It is observed from Table 4.40 that saving in Kount for IEEE 5, 14 and 30 bus systems by AELDPSO-II are 30.62 %, 35.83 % and 41.98% respectively and 34.4 %, 26.66 % and 41.98%. respectively using ASACPSO.

The results of all the PSO algorithms have been compared with lambda Iteration method and are shown in Table 4.41 for IEEE 5, 14 and 30 bus systems.

**TABLE 4.41**
**Comparison of Results of ELD by Lambda Iteration Method and PSO Algorithm**

| S. No. | METHOD/ ALGORITHM | IEEE 5 Bus | | IEEE 14 Bus | | IEEE 30 Bus | |
|---|---|---|---|---|---|---|---|
| | | Cost ( $F_C$) \$/hr. | Loss ( $F_L$) MW | Cost ( $F_C$) \$/hr. | Loss ( $F_L$) MW | Cost ( $F_C$) \$/hr. | Loss ( $F_L$) MW |
| 1 | **LAMBDA** | **761.16** | **5.18** | **1139.00** | **9.18** | **1256.20** | **12.28** |
| 2 | **PSO** | 761.43 | 5.09 | 1145.14 | 7.94 | 1266.97 | 11.05 |
| 3 | **AELDPSO-II** | 761.03 | 5.06 | 1143.31 | 7.83 | 1258.99 | 10.94 |
| 4 | **ASACPSO** | **761.37** | **5.09** | **1136.06** | **8.61** | **1258.66** | **10.93** |

It is observed that the results of IEEE-14 bus system are found to be encouraging as both cost and loss determined by ASACPSO are less than that determined by lambda- Iteration method. In case of 5-bus system, a small increase of 0.21 \$/hr. is observed for cost whereas the loss is decreased by 0.09 MW. Similarly, in case of IEEE 30-bus system, the cost of generation has increased by 2.46 \$/hr. whereas loss is decreased by 1.35 MW compared to Lambda Iteration method. Therefore, it is concluded that the results of ASACPSO are comparable to that obtained by Lambda-Iteration method.

## 4.8 CONCLUSIONS

Two types of improved PSO algorithms, Improved Particle Swarm Optimization based on Initial Selection of particles (IPSO IS) and Adaptive Social Acceleration Constant based Particle Swarm Optimization (ASACPSO) have been developed and implemented to optimize mathematical benchmark functions and Economic Load dispatch problem for IEEE 5, 14 and 30 bus systems. Conclusions for IPSO IS and ASACPSO for optimization of benchmark function and ELD problem are explained in section 4.8.1 and 4.8.2 respectively.

### 4.8.1 IPSO IS

The minimum number of particles for each mathematical function by BPSO has been determined. An improved PSO based on Initial Selection of Particles (IPSO IS) has been developed by selecting a better population of particles from the initial randomly generated particles based on function value. IPSO IS has been tested on four mathematical benchmark functions. Its performance has been compared with BPSO and has been found to give better results and faster convergence. Saving in Kount for "q" = 50 and "P" = 10 is 54.93%, 59.98%, 62.47% for Rosenbrock, Beale and Axis Parallel Hyper Ellipsoid functions respectively. However, maximum saving of 82.25% for Rosenbrock function has been obtained for "q" = 50 and "P" = 5 and maximum saving of 72.67% has been obtained for Axis Parallel Hyper Ellipsoid function for "q" = 40 and "P" = 5. In case of Booth function saving is not obtained for 10 particles but saving of 37.8% is obtained when 8 particles have been selected out of 50. In case of Booth function saving is not obtained for 10 particles but saving of 37.8% is obtained when 8 particles have been selected out of 50.

The minimum number of particles for IEEE 5, 14, and 30 bus systems has been determined by IPSO IS. Selection based PSO (IPSO IS) has been developed by selecting a better population of particles from the initial randomly generated particles based on function value. Its performance has been compared with BPSO and has been found to give better results and faster convergence. Saving in Kount for IEEE 5, 14, and 30 bus systems are 61.28%, 69.07%%, 85.38% respectively.

### 4.8.2 ASACPSO

ALDPSO, AELDPSO-I, AELDPSO-II have been developed and applied to Rosenbrock Function. It was observed from the results of Rosenbrock function that AELDPSO-II minimized the function in less Kount in comparison to ALDPSO and AELDPSO-I. ASACPSO has been developed using best value of Social Acceleration Constant obtained by AELDPSO-II, ASACPSO was applied to seven mathematical benchmark functions. It was observed that AELDPSO-II and ASACPSO converge faster in comparison to basic PSO. It has also been observed that saving in Kount is obtained for all the functions. Maximum saving of 97.6% in Kount is obtained for Schwefel function. However, for Rosenbrock function minimum saving of 17.5% Kount is achieved.

The AELDPSO, AELDPSO-I, AELDPSO–II and ASACPSO algorithms have been developed by searching the best Adaptive Social Acceleration Constant for IEEE 5, 14 and 30 bus systems. The performance of ASACPSO has been compared with BPSO and AELDPSO-II . ASACPSO has been found to give better results and converge faster than BPSO and AELDPSO-II. Saving in Kount for IEEE 5, 14 and 30 bus systems are 30.62%, 35.83% and 41.98% respectively for AELDPSO-II and 34.44%, 26.66% and 41.98% respectively for ASACPSO. The results of ASACPSO has been found to be better for

14 bus system and comparable for 5 and 30 bus systems as compared to lambda Iteration method.

## Research Publications

[1] N. K. Jain, Uma Nangia, Jyoti Jain, "An improved PSO based on initial selection of particles", IEEE first International Conference on Power Electronics, Intelligent Control and Energy, (ICPEICES 2016), Delhi Technological University (DTU), Delhi, (4-6 July, 2016).

[2] N. K. Jain, Uma Nangia, Jyoti Jain "An improved PSO based on initial selection of particles (ISBPSO) for Economic Load Dispatch" IEEE first International Conference on Power Electronics, Intelligent Control and Energy, (ICPEICES 2016), Delhi Technological University (DTU), Delhi, (4-6 July, 2016).

[3] N. K. Jain, Uma Nangia, Jyoti Jain, "Economic Load Dispatch using Adaptive Social Acceleration Constant based Particle Swarm Optimization", Journal of The Institution of Engineers India: Series B, Springer, 99 (5), pp 431 - 439, October 2018.

# CHAPTER 5
# SPLIT PHASE ECONOMIC LOAD DISPATCH USING PSO

## 5.1 INTRODUCTION

In this chapter, a new algorithm named as Split Phase Economic Load Dispatch (ELD) using Particle Swarm Optimization (PSO) has been developed. It has been named as Split Phase Economic Load Dispatch Algorithm (SPELDA). Economic Load Dispatch is a constrained optimization problem. This algorithm works in two phases. In the first phase, a population of points is randomly generated. The feasible points which satisfy the equality as well as inequality constraints of the Economic Load Dispatch problem are selected after each iteration of the algorithm. At this stage, the cost of generation is not included in the effective function to be minimized. These points are copied to the External set and then the algorithm switches over to the second phase. By doing so, a better population of points is created. All these points lie in the feasible region of the problem i.e. ELD. In the second phase, the points of External set become the initial points for Economic Load Dispatch of IEEE 5, 14 and 30 bus systems. In this phase the objective function to be minimized includes cost of generation along with constraints. A penalty factor is associated with constraints. This facilitated to restrict the points to move out of the feasible region.

In both the phases, the velocities of the points are modified by the use of clamping factors which slows down the movement of individual particles/points. The results of Split Phase Economic Load Dispatch Algorithm (SPELDA) have been compared with basic particle swarm optimization [172] and lambda iteration [92] method. This strategy of split phase gives much faster convergence for ELD as compared to Basic PSO. Analysis of results further shows that the cost of generation obtained by SPELDA is lesser compared to those obtained by using Basic PSO and Lambda iteration method. Economic Load Dispatch (ELD) is the process of allocating the required load and losses among the available generation units such that the cost of generation is minimized. The ELD problem is formulated as a nonlinear constrained optimization problem with both equality and inequality constraints. Economic Load Dispatch problem minimizes the total fuel cost of all committed plants while meeting the demand and losses.

## 5.2 PROBLEM STATEMENT

The objective of economic load dispatch problem is to minimize the cost of generation.

Mathematically, the problem is defined as:

Minimize

$$F_C = \sum_{i=1}^{NG} a_i P_{gi}^2 + b_i P_{gi} + c_i \tag{5.1}$$

Subject to

Inequality constraint

$$P_{gimin} \leq P_{gi} \leq P_{gimax} \quad i=1,2..NG \tag{5.2}$$

Equality constraint

$$f = \sum_{i=1}^{NG} P_{gi} - P_D - F_L \tag{5.3}$$

f is equality constraint, $P_D$ and $F_L$ are demand and losses for system. Transmission Losses are calculated using the following equation

$$F_L = \sum_{i=1}^{NG} \sum_{j=1}^{NG} P_{gi} B_{ij} P_{gj} \tag{5.4}$$

## 5.3 SPLIT PHASE ECONOMIC LOAD DISPATCH ALGORITHM (SPELDA)

In this chapter, a new algorithm Split Phase Economic Load Dispatch-algorithm (SPELDA) has been developed and applied for ELD of IEEE 5,14 and 30 bus systems. This algorithm performs ELD using PSO in two phases. In the first phase, it identifies the feasible points i.e. the points which satisfy the constraints of the problem and in second phase, it performs ELD by BPSO using these feasible points as initial points. Further to prevent premature convergence, the position of the particles in both the phases has been updated by the use of clamping factors Cf1 and Cf2. This algorithm results in lesser cost of generation compared to that obtained by BPSO and lambda iteration method with lower computational effort.

### 5.3.1 First Phase for Selection Mechanism

In the first phase, the algorithm identifies all the feasible points that satisfy equality and inequality constraints of ELD problem. The inequality constraints defined by equation (5.2) have been considered by generating the points between lower and upper limits of generator. The function to be minimized during this phase of the algorithm consists of only the equality constraints and is given by equation (5.3). The aim, here, remains to drag the particles well inside the feasible region. The moment a particle enters the feasible region, it is identified and isolated from the main population. These feasible points are stored in External set and are not allowed to participate in future iteration of BPSO. In this phase the position of points is updated by

$$x_{ij}^{k+1} = x_{ij}^{k} + Cf1 * V_{ij}^{k+1}$$

$i=1, 2, 3 \ldots NG, \quad j=1, 2, 3 \ldots q$ (5.5) where Cf1 is the clamping factor for first phase of algorithm. This facilitates a part of population to enter into the feasible region faster. This process stops after required number of such points are identified. In this phase, ten number of points lying in the feasible region are identified. Flow chart of First Phase of SPELDA is shown in Fig.5.1.

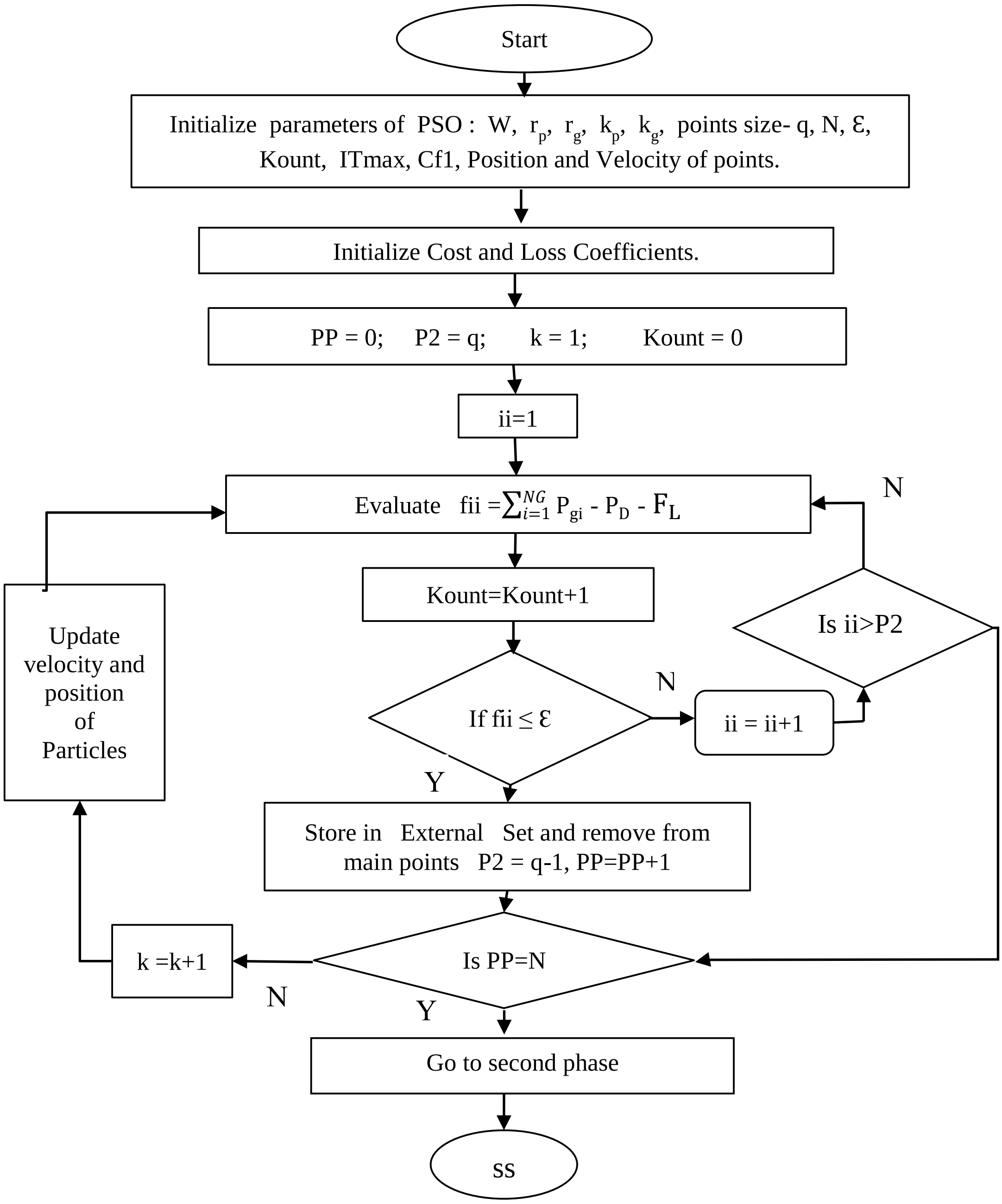



**Fig.5.1** Flow Chart for Split Phase Economic Load Dispatch Algorithm (First Phase)

### 5.3.2 Second Phase of Economic Load Dispatch

In the second phase of proposed algorithm, the points selected in first phase which are stored in the External set become the initial points for performing Economic Load Dispatch of IEEE 5,14 and 30 bus systems using BPSO. Cost of generation is given by equation (5.1). The flow chart of second phase is shown in Fig 5.2. The objective function to be minimized is now changed to

$$F = F_C + k_p * f \tag{5.6}$$

Where $k_p$ is the penalty factor. This restricts the points / particles to jump out of the feasible region. The value of $k_p$ should be sufficiently high. Here, $k_p$ = 1000 has been considered.

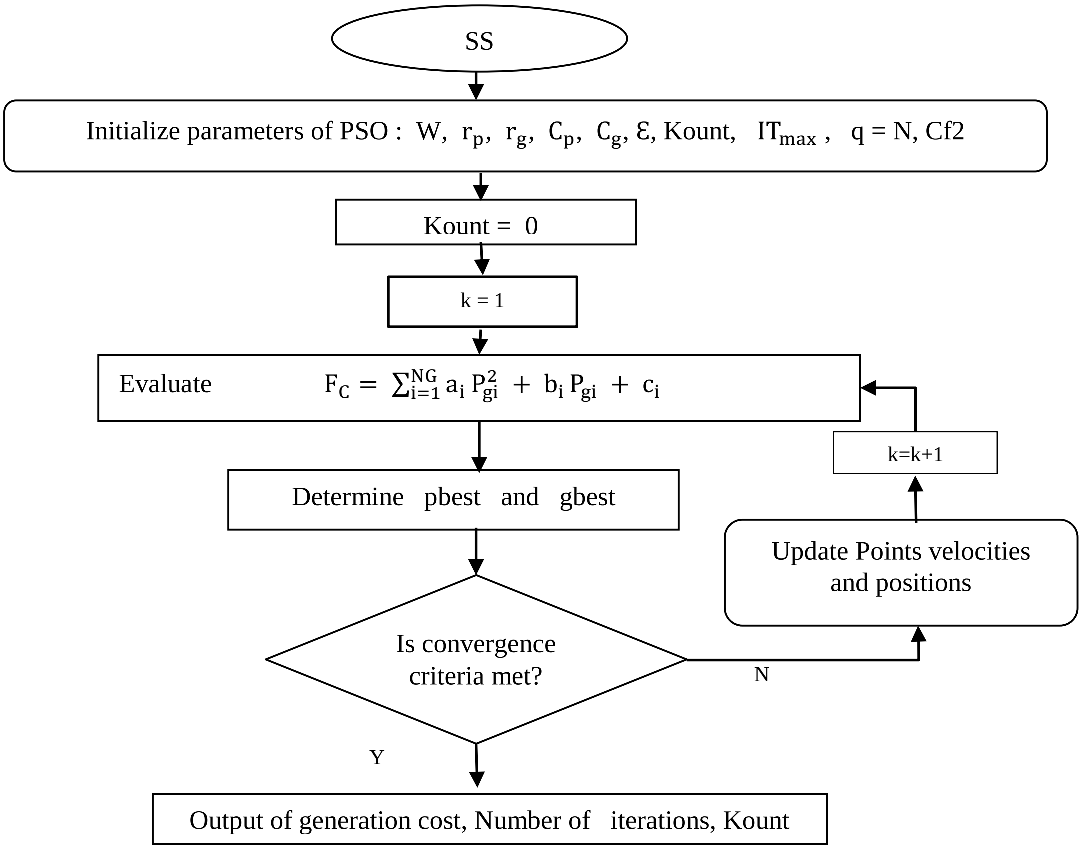



**Fig. 5.2** Flow Chart for Split Phase Economic Load Dispatch Algorithm (Second Phase)

Further, to prevent premature convergence, velocity is modified by clamping factor Cf2 to update the position of particles / points.

$$x_{ij}^{k+1} = x_{ij}^{k} + Cf2 * v_{ij}^{k+1}$$

$$i = 1, 2, 3\ldots\ NG, \quad j = 1, 2, 3\ldots.q \qquad (5.7)$$

The steps for SPELDA for ELD problem are:

1. Initialize Parameters of PSO-W, $C_p$, $C_g$, $r_p$, $r_g$, k (iteration number), ITmax, ε, q (population size, or Initial points), N (Number of points in the External set) and function evaluations (Kount). Initialize k=0 and Kount = 0. Initialize cost characteristics of generators. Generate points between $P_{gimin}$ and $P_{gimax}$. In this problem, value of W, $C_p$, $C_g$, $r_p$, $r_g$, ITmax, ε, q and N are 0.6, 1,1,1,1, 2000, $1 * 10^{-6}$, 20 and 10 respectively have been selected. Generate random velocity and position for the initial points.

2. Calculate function (fitness) value for all points. The function to be optimized at present corresponds to the equality constraint defined by equation (5.3). Increment Kount by q.

3. Determine pbest and gbest.

4. Determine point number corresponding to gbest.

5. For all points, check, if function value f < Ɛ. If yes, store point number and its coordinates in the External set and remove this from main points.

6. Increase the iteration by one. k = k+1.

7. Calculate velocity of all points for next iteration.

8. Update position of all points using (5.5).

9. Check generator constraints. Fix the generation to the limit violated.

10. Calculate function f for each point. Increment Kount by one for each particle/point.

11. Increment iteration count by one i.e. k = k+1.

12. Update $x_{pbest}$ and $x_{gbest}$ values.

13. For all points check if function value f < Ɛ. If yes, store point number, its coordinates in the External set and remove from main points.

14. Check if the number of points in the External set exceeds the specified value. If yes, go to 15. Else, go to 7.

15. Perform Economic Load Dispatch by BPSO to Minimize F.

    $F = F_C + k_p * f$ Using points of External set as the initial points.

16. Determine $x_{pbest}$ and $x_{gbest}$.

17. Check if the stopping criterion is met. If yes, go to 20. Else, go to 18.

18. Update velocity using equation (3.21) and position using equation (5.7).

19. Increment k and Kount. Go to 15.

20. Output the cost of generation, no of iterations, number of function evaluations i.e. Kount.

## 5.4 COMPUTATIONAL PROCEDURE

The Split Phase Economic Load Dispatch Algorithm (SPELDA) which works in two phases has been implemented for Economic Load Dispatch of IEEE 5, 14 and 30 bus systems. In the first phase ten feasible points are identified and stored in the External set.

In the second phase ELD is performed using BPSO and the points of External set become the initial points. In the first phase, position of points is updated by equation (5.5). The value of Cf1 and Cf2 has been searched for minimum Kount by varying Cf1and Cf2 from 0.1 to 0.9. The convergence in minimum Kount was obtained for Cf1 = 0.6 for IEEE 5 bus system and Cf1= 0.452 for IEEE 14 bus and 30 bus systems.

In the second phase, the objective function f (equation (5.3)) is replaced by F= $F_C + k_p$ f equation (5.6) . Faster convergence for all the systems was obtained for Cf2=0.4. Second phase ends when the convergence criteria are met.

## 5.5 COMPUTATIONAL RESULTS AND DISCUSSION

Split Phase Economic Load Dispatch Algorithm (SPELDA) has been implemented on IEEE 5 bus, 14 bus and 30 bus systems for ELD. Table 5.1 shows the results of ELD using Basic Particle Swarm Optimization (BPSO) for following values of parameters: Initial Number of Particles = q = 20, $IT_{max} = 3000$, $C_p, C_g, r_p, r_g = 1$ and W = 0.6.

**TABLE 5.1**
**Results of ELD for BPSO**

| S. No.<br>(1) | IEEE Bus System<br>(2) | Cost $/hr.<br>(3) | Kount<br>(4) | No. of iterations (k)<br>(5) |
|---|---|---|---|---|
| 1 | 5 | 761.6296 | 4060 | 101 |
| 2 | 14 | 1155.58.96 | 5100 | 127 |
| 3 | 30 | 1248.06 | 4100 | 102 |

Column (2) of Table 5.1 shows the IEEE bus systems, Columns (3), (4) and (5) represent cost of the generation, Kount and iterations required to optimize the function respectively.

Table 5.2 represents the results of ELD obtained by SPELDA for IEEE 5 bus, 14 bus and 30 bus systems for fixed value of Parameters of SPELDA algorithm.

Initial Points = q = 20; N = Points in the External set = 10;

$C_p, C_g, r_p, r_g = 1$; W = 0.6; and Cf2 = 0.4.

Columns (2), (3), (4), (5) and (6) of Table 5.2 represent IEEE bus systems, cost of generation, clamping factor for first phase, Kount and iteration required to optimize the ELD problem respectively.

**TABLE 5.2**
**Results of SPELDA (q = 20, N = 10, Cf2 = 0.4)**

| **S. No. (1)** | **IEEE Bus Systems (2)** | **$F_C$ (\$/hr.) (3)** | **Cf1 (4)** | **Kount (5)** | **No of iterations (k) (6)** |
|---|---|---|---|---|---|
| 1 | 5 | 761.02 | 0.6 | 2262 | 87 |
| 2 | 14 | 1137.15 | 0.452 | 2444 | 87 |
| 3 | 30 | 1246.53 | 0.452 | 2592 | 93 |

Table 5.3 shows the comparison of BPSO and SPELDA in terms of Kount required to perform the Economic Load Dispatch for IEEE 5, 14 and 30 bus systems. It is also shown in Fig.5.3.

**TABLE 5.3**
**Comparison of Kount for BPSO and SPELDA**

| **S.No. (1)** | **IEEE Bus Systems (2)** | **KOUNT** | | **% Saving in Kount (5)** |
|---|---|---|---|---|
| | | **BPSO (3)** | **SPELDA (4)** | |
| **1** | **5** | 4060 | 2262 | **44.29%** |
| **2** | **14** | 5100 | 2444 | **52.08%** |
| **3** | **30** | 4100 | 2592 | **36.78%** |

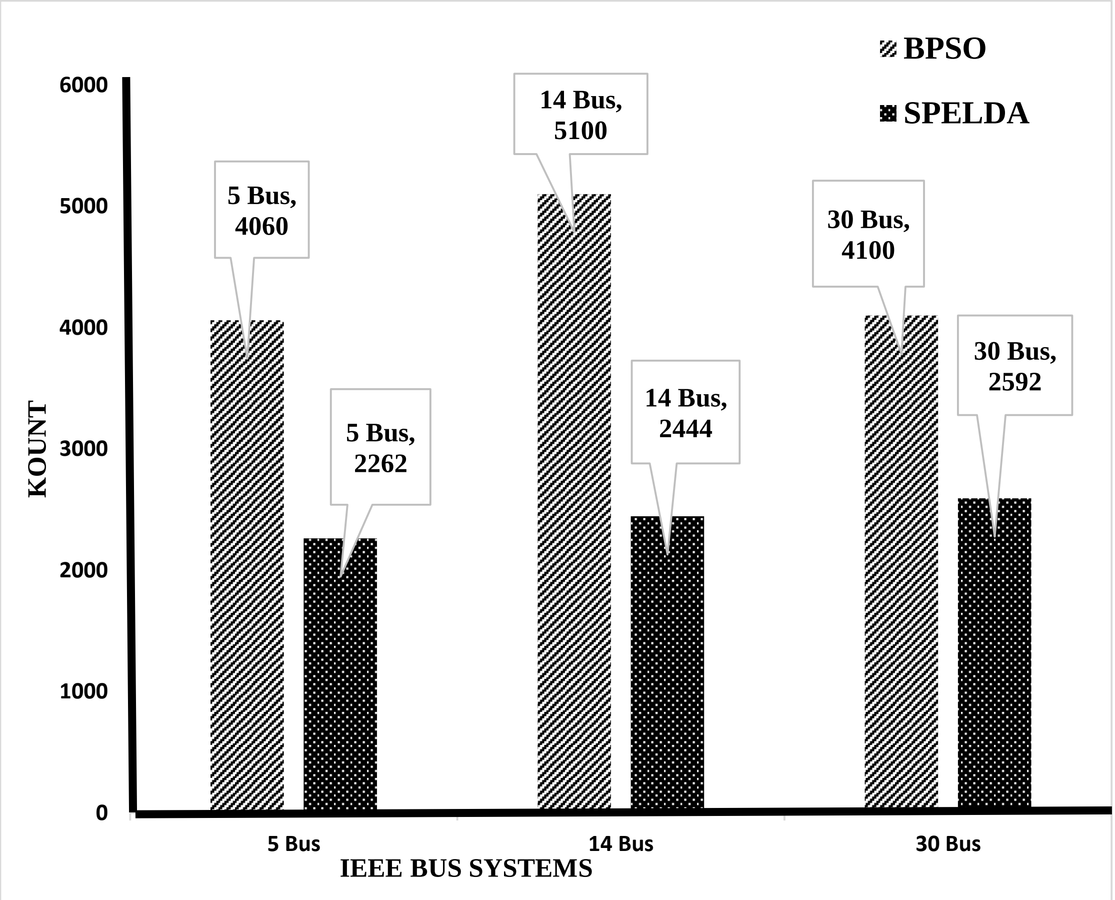



**Fig. 5.3** Comparison of Kount for BPSO and SPELDA

Table 5.4 Shows the comparison of cost of Generation in $/hr. for Economic Load Dispatch of IEEE 5, 14 and 30 bus systems. Columns (3), (4) and (5) show the results of ELD using BPSO, Lambda Iteration and SPELDA respectively.

**TABLE 5.4**
**Comparison of Cost of Generation for ELD**

| S. No. (1) | IEEE Bus Systems (2) | Methods / Algorithm | | |
|---|---|---|---|---|
| | | BPSO Cost ($/hr.) (3) | Lambda iteration Cost ($/hr.) (4) | SPELDA Cost ($/hr.) (5) |
| 1 | IEEE 5 bus | 761.62 | 761.16 | **761.02** |
| 2 | IEEE 14 bus | 1155.58 | 1139 | **1137.15** |
| 3 | IEEE 30 bus | 1248.06 | 1256 | **1246.53** |

These results of ELD using BPSO, Lambda Iteration and SPELDA have been compared and graphically represented in Fig. 5.4, 5.5 and 5.6 for IEEE 5 Bus, IEEE 14 Bus and IEEE 30 Bus Systems respectively.

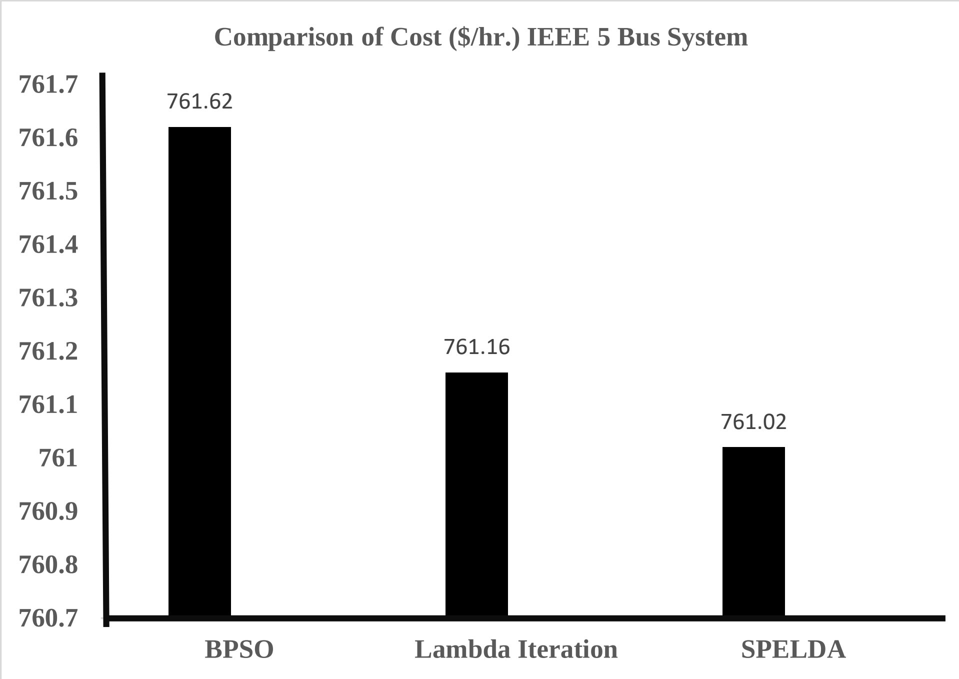



**Fig. 5.4** Comparison of Cost ($/hr.) for IEEE 5 Bus System

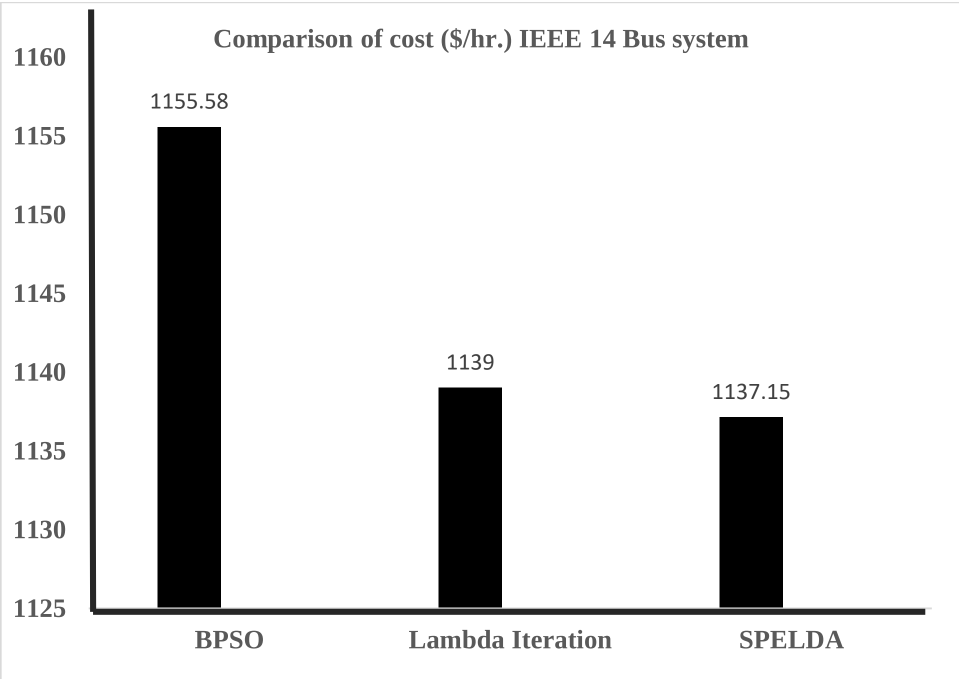



**Fig.5.5** Comparison of Cost $/hr. for IEEE 14 Bus System

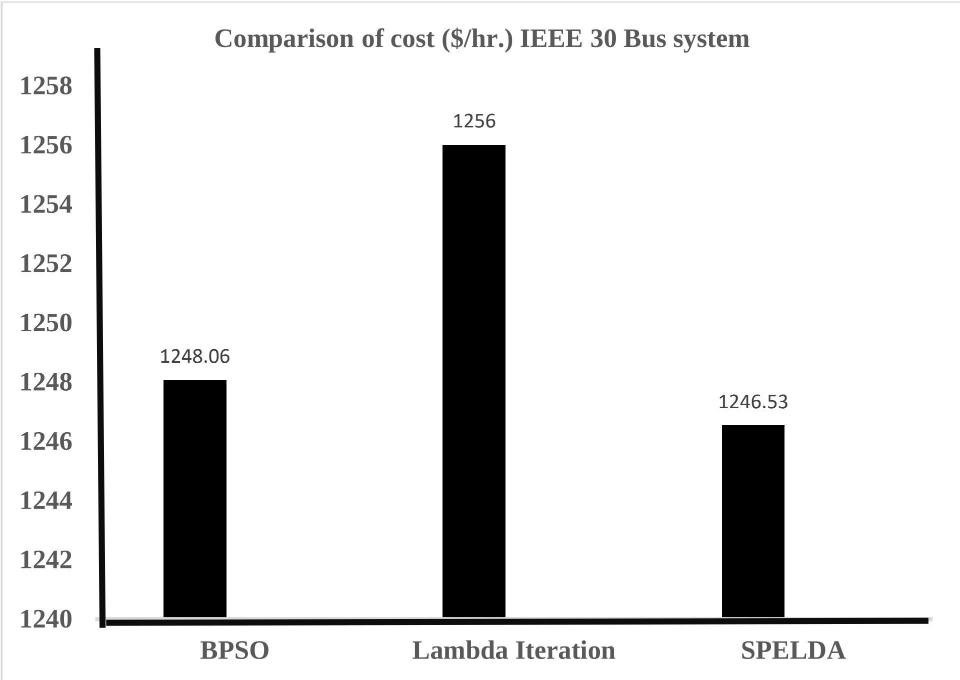



**Fig. 5.6** Comparison of Cost for IEEE 30 Bus System

From the results of Table 5.3, it is observed that a saving of 44.29%, 52.08% and 36.78% in Kount is obtained for ELD of IEEE 5 bus, 14 bus and 30 bus systems respectively.

It is further observed from the results of ELD of IEEE 5 bus, 14 bus and 30 bus system represented in Table 5.4 and Fig. 5.4, 5.5 and 5.6 that cost of generation obtained by SPELDA is lesser than that obtained by BPSO and lambda iteration method for all three systems.

## 5.6 CONCLUSIONS

In this Chapter, a new algorithm - Split Phase Economic Load Dispatch Algorithm (SPELDA) for Economic Load dispatch using basic Particle Swarm optimization (BPSO) has been developed. It works in two phases. In the first phase the feasible points are identified which become the initial points for performing ELD in the second phase. It has been successfully implemented on IEEE 5, 14 and 30 bus systems. The results of SPELDA have been compared with basic PSO (BPSO) and Lambda iteration method. The new algorithm SPELDA is found to perform the best from computational effort as well as from quality of result points of views. SPELDA resulted in faster convergence as 44.29%,52.08% and 36.78% saving in Kount is obtained for IEEE 5, 14 and 30 bus systems respectively. Also the cost of generation is 761.023$/hr., 1137.55 $/hr. and 1246.55 $/hr. for IEEE 5, 14 and 30 bus systems which is much less than that obtained by Basic PSO (BPSO) and Lambda iteration method.

## Research Publication:

N. K. Jain, Uma Nangia, Jyoti Jain, "Split Phase Economic Load Dispatch Algorithm". Journal of the Institution of Engineers India : Series B, Springer. (communicated to IEIB, Jan. 2019).

# CHAPTER 6
# MULTIOBJECTIVE ECONOMIC LOAD DISPATCH

## 6.1 INTRODUCTION

In general, a large scale power system possesses multiple objectives to be achieved. The optimal Power System operation is achieved when various objectives of Power System: cost of generation, system transmission losses, environmental emissions and security etc. are simultaneously attained. Single objective optimization techniques cannot handle such problems because of conflicting nature of these objectives. Single objective optimization techniques give optimal solution in respect of an objective function under consideration. The way out, therefore, lies in the multiobjective approach [3, 6] to problem solving.

Multiobjective economic load dispatch problem has been formulated using weighting method [1, 4, 21] and modified form of constrained method [1,176]. The noninferior set has been generated for IEEE 5, 14 and 30 bus systems using PSO and GA technique. The noninferior set obtained has been displayed in 3-D space considering all three objectives and in 2 - D space considering all combinations of two objectives for IEEE 5, 14 and 30 bus systems. In the present research work, Weighting Method and Constraint method have been implemented for solving Multiobjective Economic Load Dispatch (MELD) problem. Target point is obtained by Maximization of minimum Relative Attainment [113,176] and Fuzzy logic system [187, 45, 52, 93].

In this Chapter, three important objectives of Power System- cost of generation, system transmission losses and environmental emissions have been considered.

## 6.2 FORMULATION OF GENERAL MULTIOBJECTIVE PROGAMMING PROBLEM

The general multiobjective optimization [29] problem with N decision variables, m constraints and h objectives is

Minimize

$$Z\,(x_1, x_2, x_3, \ldots\ldots\ldots\ldots, x_N) = [Z_1(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N);$$
$$Z_2\,(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N);$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ ;$$
$$Z_h(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N)]; \qquad (6.1)$$

Subject to

$$g_i\,(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N) \leq 0 \qquad i=1, 2, \ldots\ldots\ldots, m \qquad (6.2)$$

$$x_j \geq 0 \qquad j=1, 2, \ldots\ldots\ldots, N \qquad (6.3)$$

$Z(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N)$ is the multiobjective function and $Z_1(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N)$, $Z_2(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N)$, $Z_h(x_1, x_2, x_3 \ldots\ldots\ldots\ldots, x_N)$ are the h individual objective functions. In the Multiobjective function Z the various individual objectives have just been written, but it does not imply any kind of operation say multiplication, addition or anything whatsoever in general. In particular, Z can be designed to incorporate $Z_1$, $Z_2$, $Z_{3\ldots.}$, $Z_h$ depending upon the approach.

In economic load dispatch, cost of generation is considered as the objective function to be minimized. In optimum reactive power generation, transmission losses can be considered as the objective function to be minimized. This is because, in a decoupled sense, transmission losses are mainly dependent on voltage magnitudes, which are dependent on the reactive powers. Therefore, optimization of reactive power dispatch can be considered as minimization of system active power losses. This will also improve the voltage profile and would result in the reduction of cost of installing extra equipment for VAR generation and voltage adjustment [7].

In the present work, oxides of nitrogen emission are taken as the index for environmental pollution. The amount of $NO_x$ emission is given as a function of generator output. Multiobjective economic load dispatch studies have been carried out on IEEE 5, 14 and 30 bus systems in 2-D and 3-D space. The data of 5, 14 and 30 bus system is given in Appendix I. In 2-D space, two objectives i.e. cost of generation ($F_C$) and system transmission losses ($F_L$) have been considered. In 3-D space, in addition to the above-mentioned two objectives, environmental emission ($F_E$) is also considered.

The ideal situation where one would like to operate the Power Systems is one where all the objectives i.e. cost of generation, system transmission losses ($F_L$) and environmental emission ($F_E$) are minimum. Such a point is called the ***Ideal point.*** It is represented by ($F_{Cmin,}F_{Lmin}$) in 2-D space, whereas in 3-D space it is represented by ($F_{Cmin,}F_{Lmin,}F_{Emin}$). However, such a point is not feasible. If it was, then there would not be any conflict among the objectives.

Therefore, while considering Multiobjective Economic load dispatch problem, a strategy has to be adopted by the Power Systems analyst or operator to achieve optimum values as per his satisfaction level and requirements. The operating point so obtained is called Target Point (TP) or the best – compromise solution.

## 6.3 NONINFERIORITY

In single objective problems, an optimal solution is obtained which gives the best value of the objective function under consideration. However, this notion of optimality must be dropped for multiobjective problems because a solution, which minimizes one objective, will not, in general, minimize any of other objectives. What is optimal in terms of one of the h objectives is usually nonoptimal for the other h-1 objectives. Through single objective optimization, we get a solution, which is no doubt an optimal solution, but whether it is the best or not, is to be decided by the electric utility. If it is, there is no problem. However, if it is not, which may be the case many a times (generally is the case) as utilities would not be willing to make a decision in favor of adopting the solution (optimum solution). Further the Power Systems analyst has nothing else (any other solution) to present before the utilities to facilitate their decision making process. The process of decision-making gets stuck up. It is because the utilities may be satisfied to achieve the best value of one objective, but simultaneously utilities may be utterly dissatisfied over the values of certain other objectives which may be intolerably bad. Utilities may be interested in achieving better values of some other objectives even at the cost of the objective, which the power systems analyst has optimized through single objective optimization method. But, these single objective optimization techniques cannot give any solution other than the optimum solution. Therefore, multiobjective approach should be adopted for problem solving and there is no optimal solution for a multiobjective problem. A new concept called noninferiority serves the similar purpose.

A feasible solution to a multiobjective programming problem is noninferior if there exists no other feasible solution that will yield an improvement in one objective without causing degradation in at least one of the other objectives. A given noninferior solution may or may not be acceptable to the decision maker. However, it is important to note that, it is one of these noninferior solutions for which decision maker looks for.

### 6.3.1 Graphical Explanation of Noninferiority

Let us explain this definition graphically. An arbitrary collection of feasible alternatives for a two objective minimization problem is shown in Fig. 6.1. Curve 1 forms the boundary of the feasible region. The definition of noninferiority can be used to find noninferior solutions in Fig. 6.1. All the feasible solutions above curve 1 are inferior because they yield more of both $Z_1$ ($F_C$) and $Z_2$ ($F_L$).

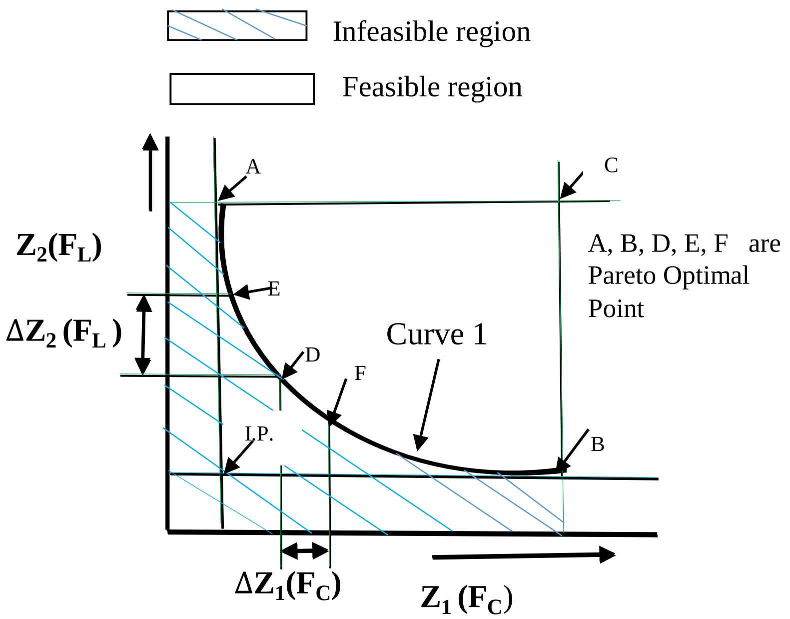



**Fig.6.1** Graphical Explanation of Noninferiority

Consider, an exterior point C in Fig. 6.1, which is feasible but inferior. Alternative A gives lesser of $Z_1(F_C)$ than does C without increasing the amount of $Z_2(F_L)$. Alternative B gives lesser amount of $Z_2$ ($F_L$) without increasing the amount of $Z_1$ ($F_C$). For these reasons, point C represents a feasible but inferior solution. Consider point D on curve 1. Suppose it is desired to achieve lesser value of $Z_1$ ($F_C$) than the value at point D. Since it is not desirable to move to the left of curve AB as even though it gives lesser value of $Z_1$ ($F_C$), yet it lies in the infeasible region. Therefore, it is desirable to move upward only along the curve 1 to have lesser value of $Z_1$ ($F_C$). Let us say, we get point E. At this point, we get lesser value of $Z_1$ ($F_C$) but there is some increase in $Z_2$ ($F_L$). In other words, in order to gain on $Z_1$ ($F_C$) we have to sacrifice $\Delta Z_2$ ($F_L$) units of $Z_2$ ($F_L$). Similarly, in moving from D to F, we have to sacrifice $\Delta Z_1$ ($\Delta F_C$) units of $Z_1$ ($F_C$) to gain on $Z_2$ ($F_L$). Therefore, we can say that points D, E and F are noninferior. Mathematically, a solution x is noninferior for a minimization problem if there exists no feasible y such that

$$Z_K(y) \leq Z_K(x) \qquad K=1, 2\ldots\ldots\ldots h \qquad (6.4)$$

and $$Z_K(y) < Z_K(x) \quad \text{for at least one} \quad K=1, 2\ldots\ldots\ldots h \qquad (6.5)$$

It means a solution to a multiobjective minimization problem is said to be noninferior, if it is not possible to improve upon one of the objectives without deteriorating in at least one of these objectives.

## 6.4 CLASSIFICATION OF MULTIOBJECTIVE METHODS

There are several approaches to multiobjective programming and planning. These methods are categorized based on the role of the power system analyst in the planning process. The role of the analyst is determined by the information flows in the planning process. There are two types of information flows.

**1. Bottom - up (From Power System Analyst to decision maker (Power System Operator))**

The analyst to decision maker or bottom up flow contains results about the noninferior set – noninferior alternatives and tradeoffs among the objectives.

**2 Top - down (From decision maker (Power System Operator) to Analyst)**

The decision maker to analyst or top - down flow occurs when decision maker (Power system operators) explicitly articulate preferences so that a best - compromise solution is identified. On the basis of information flows, the multiobjective methods are categorized into Generating Techniques and Techniques that incorporate preferences.

Generating techniques emphasize the development of information about a multiobjective problem that is presented to a decision-maker (Power System Operator) that allows the range of choice and tradeoffs among objectives to be well understood. The information flow is of bottom-up variety. Power System analyst apply a generating technique to find an exact representation or an approximation of the noninferior set. The results are then presented to the decision-maker (Power System Operator) either graphically or in a tabular form, which, based on this information, selects a best-compromise solution. There are several generating techniques:

- Weighting method
- Constraint method
- Noninferior Set Estimation (NISE) Method
- Multiobjective Simplex method

In the present research work, Weighting Method and Constraint method have been implemented for solving Multiobjective Economic Load Dispatch (MELD) problem.

## 6.5 FORMULATION OF MELD PROBLEM

Three aspects of the multiobjective economic load dispatch in 3-D space are:

(1) To minimize the cost of generation

(2) To minimize the system transmission losses

(3) To minimize the environmental emission

### 6.5.1 Weighted Sum Method

The objective function (F) to be minimized is formulated as the weighted sum of objectives.

The MELD problem in 3-D space is

Minimize

$$F=[F_C, F_L, F_E] \tag{6.6}$$

Where

$$F = W_1*F_C+ W_2*F_L+ W_3*F_E + k_p * (P_D+F_L-\sum_{i=1}^{NG} P_{gi}) \tag{6.7}$$

$$F_C = \sum_{i=1}^{NG} a_i P_{gi}^2+b_i P_{gi} + c_i \tag{6.8}$$

$$F_L = \sum_{i=1}^{NG} \sum_{j=1}^{NG} P_{gi} B_{ij} P_{gj} \tag{6.9}$$

$$F_E = \sum_{i=1}^{NG} d_i P_{gi}^2+e_i P_{gi} + f_i \tag{6.10}$$

Subject to the constraints

Equality constraint

$$\sum_{i=1}^{NG} P_{gi} = P_D + F_L \tag{6.11}$$

Inequality constraint

$$P_{gimin} \le P_{gi} \le P_{gimax} \qquad i = 1, 2... NG \tag{6.12}$$

## 6.6 COMPUTATIONAL PROCEDURE

The steps for generation of noninferior set for Multiobjective Economic Load Dispatch (MELD) problem using Particle Swarm Optimization technique are as follows:

1. Set the parameters of PSO:

$IT_{max}$ = 1000, q = 40, w = 0.6,

$r_p = C_g = r_g = C_p = 1$

2. Set the cost, loss and emission coefficients. Also set the load demand and generator limits of all the generators.

3. Assign the value of $W_1$, $W_2$, $W_3$ (weights) attached to $F_C$, $F_L$, $F_E$.

4. Fix the value of penalty parameter $k_p = 1000$.

5. Set iteration count k = 0 and Kount = 0.

6. Generate initial random position and velocity for all particles.

7. Calculate the MELD function using (6.7) for each particle.

8. Determine the personal best and global best positions.

9. Increase Kount by q i.e. Kount=Kount + q.

10. Increase the iteration count k by 1 i.e. k = k+1

11. Calculate the velocity of each particle using (3.21).

12. Check if velocity is within the limits. If not fix the velocity to the limit violated.

13. Calculate the new positions of the particles by evaluating (3.22).

14. Calculate function value using (6.7) for each particle.

15. Update $x_{pbest}$ and $x_{gbest}$ values.

16. Check if both the stopping criteria are satisfied, if not then go to step 9, else go to 17.

17. Output the optimized values for cost of generation, system transmission losses and environmental emission i.e. the noninferior set.

Flow chart for Multiobjective Economic Load Dispatch (MELD) problem is shown in Fig. 6.2.

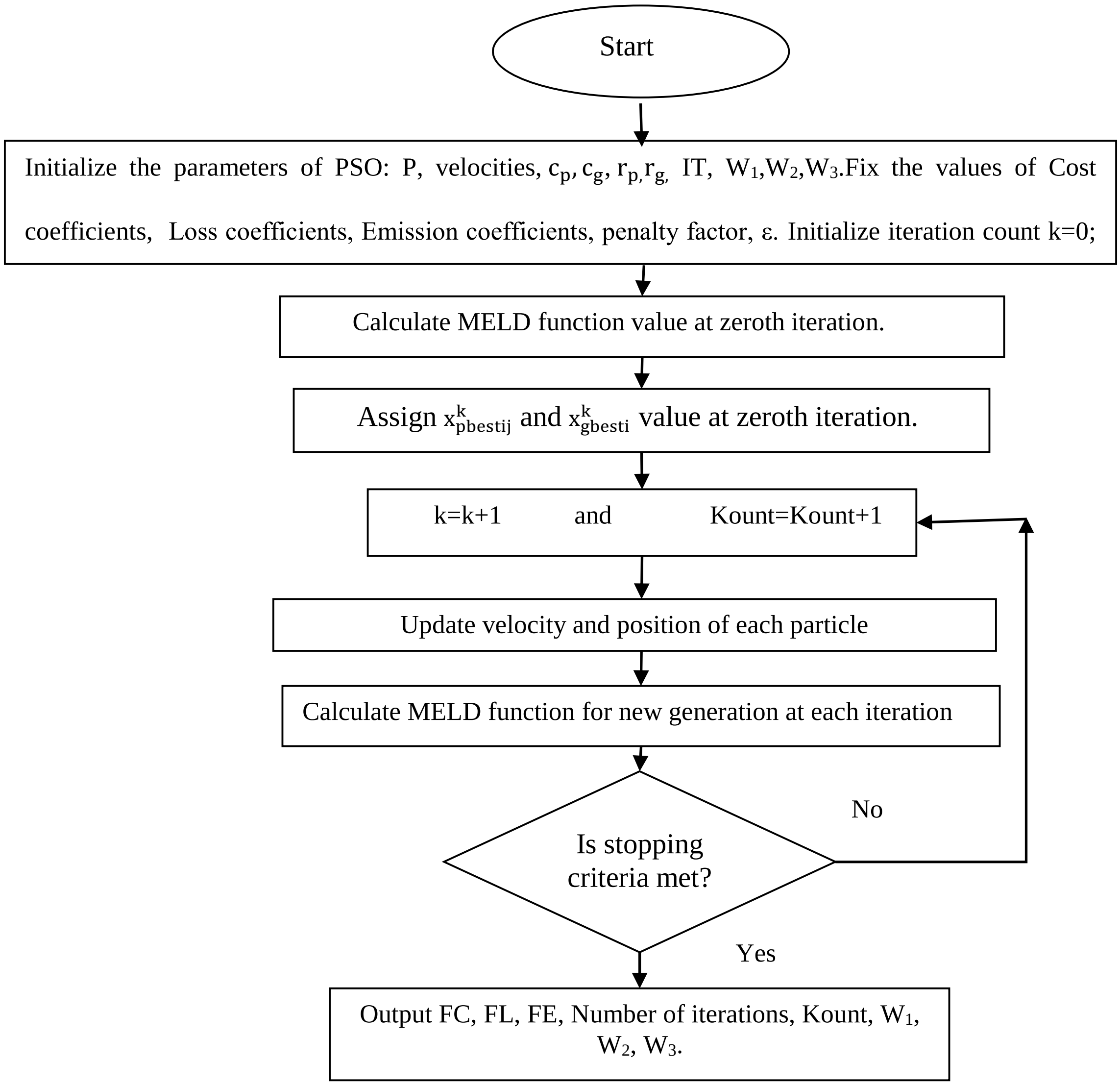



**Fig. 6.2** Flow Chart for MELD

Target point has been achieved by Maximization of Minimum Relative Attainments method and Fuzzy logic system.

### 6.6.1 Maximal Effectiveness Principle

The Maximal Effectiveness principle makes use of the concept of an Ideal solution. This method assumes that all the objectives are being minimized and seeks the solution which maximizes the minimum relative attainment, $\tau_i$ (x) by any objective $Z_i(x)$ of its ideal reference value $Z_{imin}$ relative to its worst feasible value $Z_i^{worst}$.

$\tau_i\ (x) = (Z_i(x) - Z_i^{worst})\ /(Z_{imin} - Z_i^{worst})$ $\quad$ i=1,2,3…h $\quad$ (6.13)

where h = No. of objectives

This algorithm to determine the Target Point is given below:

1) Define the function as weighted sum of all the objectives.
2) Generate the noninferior set by Weighting method.
3) Take the first element of the noninferior set. Put M=1.
4) Calculate the relative attainment of all the objectives using equation (6.13).
5) Calculate the sum of Minimum relative attainments of all the objectives corresponding to the element of noninferior set.
6) If M > 1, find the sum which is maximum. Else, increase M by 1.
7) Check if all the elements of noninferior set have been considered.
8) If yes, output the objectives corresponding to maximum minimum relative attainment and stop; else go to step 4.

## 6.7 COMPUTATIONAL RESULTS AND DISCUSSION

Tables 6.1, 6.2 and 6.3 show the noninferior set of Multiobjective Economic Load Dispatch (MELD) using PSO and corresponding minimum relative attainments for IEEE 5 bus, IEEE 14 bus and IEEE 30 bus system respectively in 3-D space. Columns (2), (3) and (4) of Tables 6.1, 6.2 and 6.3 show the weights $W_1$, $W_2$ and $W_3$ respectively. Columns (5), (6) and (7) show the noninferior set. Columns (8), (9) and (10) show the minimum relative attainments of cost of generation ($\tau_C$), transmission losses ($\tau_L$) and environmental emissions ($\tau_E$) respectively. The column (11) shows the sum of minimum relative attainments of all the objectives. Column (12) shows the Kount (function evaluation) required to minimize the function and column (13) shows the number of iterations required to complete the optimization.

**TABLE 6.1**
**Results of MELD and Minimum Relative Attainments in 3-D Space**
**(IEEE 5 Bus System)**

| S. No. (1) | $W_1$ (2) | $W_2$ (3) | $W_3$ (4) | $F_C$ ($/hr) (5) | $F_L$ (MW) (6) | $F_E$ (kg/hr) (7) | $\tau_C$ (8) | $\tau_L$ (9) | $\tau_E$ (10) | $\sum\tau$ (11) | Kount (12) | k (13) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 0 | 0 | 760.960 | 5.1834 | 129.646 | 1.00 | 0.00 | 0.00 | 1.00 | 9760 | 243 |
| 2 | 100 | 100 | 0 | 760.978 | 5.1599 | 128.310 | 1.00 | 0.19 | 0.15 | 1.34 | 9480 | 236 |
| 3 | 100 | 200 | 0 | 760.999 | 5.1507 | 127.770 | 0.99 | 0.26 | 0.21 | 1.46 | 9440 | 235 |
| 4 | 100 | 420 | 0 | 761.064 | 5.1333 | 126.730 | 0.98 | 0.4 | 0.33 | 1.71 | 14120 | 352 |
| 5 | 100 | 0 | 10 | 761.255 | 5.1075 | 125.070 | 0.94 | 0.61 | 0.51 | 2.07 | 21280 | 531 |
| **6** | **100** | **0** | **20** | **761.643** | **5.0822** | **123.230** | **0.84** | **0.82** | **0.72** | **2.37** | **9320** | **232** |
| 7 | 100 | 3500 | 0 | 761.913 | 5.0725 | 122.380 | 0.62 | 0.89 | 0.81 | 2.33 | 9400 | 234 |
| 8 | 100 | 4600 | 0 | 762.039 | 5.0692 | 122.064 | 0.47 | 0.92 | 0.85 | 2.24 | 9320 | 232 |
| 9 | 118 | 2800 | 39 | 762.060 | 5.0685 | 121.992 | 0.4 | 0.93 | 0.86 | 2.18 | 9400 | 234 |
| 10 | 100 | 5000 | 0 | 762.085 | 5.0681 | 121.954 | 0.39 | 0.93 | 0.86 | 2.18 | 9640 | 240 |
| 11 | 100 | 0 | 30 | 762.230 | 5.0650 | 121.613 | 0.37 | 0.95 | 0.90 | 2.23 | 9200 | 230 |
| 12 | 100 | 5800 | 0 | 762.252 | 5.0640 | 121.583 | 0.29 | 0.96 | 0.90 | 2.16 | 9320 | 232 |
| 13 | 100 | 6000 | 0 | 762.262 | 5.0647 | 121.562 | 0.28 | 0.96 | 0.90 | 2.14 | 9320 | 232 |
| 14 | 160 | 200 | 77 | 762.270 | 5.0640 | 121.527 | 0.27 | 0.96 | 0.91 | 2.14 | 9360 | 233 |
| 15 | 160 | 40 | 80 | 762.310 | 5.0600 | 121.452 | 0.27 | 0.99 | 0.92 | 2.18 | 9440 | 235 |
| 16 | 160 | 75 | 83 | 762.320 | 5.0637 | 121.430 | 0.25 | 0.96 | 0.92 | 2.13 | 9480 | 236 |
| 17 | 180 | 180 | 95 | 762.369 | 5.0630 | 121.350 | 0.24 | 0.97 | 0.93 | 2.14 | 9520 | 237 |
| 18 | 160 | 70 | 85 | 762.440 | 5.0620 | 121.200 | 0.21 | 0.98 | 0.94 | 2.14 | 9440 | 235 |
| 19 | 0 | 100 | 0 | 762.742 | 5.0594 | 120.720 | 0.17 | 1 | 1.00 | 2.17 | 9200 | 229 |
| 20 | 0 | 100 | 20 | 762.753 | 5.0593 | 120.700 | 0.01 | 1 | 1.00 | 2.01 | 9520 | 237 |

Following observations are made from Table 6.1.

$F_{Cmax}$ = 762.753 $/hr., $F_{Lmax}$ = 5.18346 MW $F_{Emax}$ = 129.646 Kg/hr.

Ideal point for IEEE 5 bus system is:

$F_{Cmin}$= 760.960 $/hr., $F_{Lmin}$= 5.05934 MW, $F_{Emin}$=120.700 Kg/hr.

The Target Point is one for which sum of minimum relative attainments is maximum. It is seen at S. No.6 of Table 6.1 and is highlighted. Target Point for IEEE 5 bus system in 3-D space is:

**$F_C$ = 761.643 $/hr., $F_L$ = 5.08239 MW, $F_E$ = 123.230 kg/hr.**

# TABLE 6.2
## Results of MELD and Minimum Relative Attainments in 3-D Space
## (IEEE 14 Bus System)

| **S. No. (1)** | **$W_1$ (2)** | **$W_2$ (3)** | **$W_3$ (4)** | **$F_C$ ($/hr) (5)** | **$F_L$ (MW) (6)** | **$F_E$ (kg/hr) (7)** | **$\tau_C$ (8)** | **$\tau_L$ (9)** | **$\tau_E$ (10)** | **$\sum\tau$ (11)** | **Kount (12)** | **k (13)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1135.670 | 8.8214 | 613.2896 | 1.000 | 0.000 | 0.002 | 1.00 | 7080 | 88 |
| 2 | 1 | 0 | 0 | 1135.736 | 8.8143 | 613.6401 | 0.998 | 0.004 | 0.000 | 1.00 | 8040 | 100 |
| 3 | 1 | 5 | 0 | 1136.704 | 8.4788 | 592.5560 | 0.983 | 0.231 | 0.174 | 1.38 | 7000 | 87 |
| 4 | 1 | 10 | 0 | 1139.498 | 8.1192 | 569.3447 | 0.938 | 0.475 | 0.366 | 1.78 | 6760 | 84 |
| 5 | 1 | 20 | 0 | 1142.426 | 7.9063 | 554.1978 | 0.892 | 0.619 | 0.492 | 2.00 | 7480 | 93 |
| 6 | 1 | 30 | 0 | 1144.029 | 7.8340 | 549.1883 | 0.866 | 0.667 | 0.533 | 2.06 | 8040 | 100 |
| 7 | 1 | 10 | 1 | 1145.619 | 7.7277 | 538.7998 | 0.841 | 0.739 | 0.619 | 2.20 | 7960 | 99 |
| 8 | 1 | 20 | 1 | 1146.091 | 7.7225 | 539.9504 | 0.833 | 0.743 | 0.610 | 2.18 | 7640 | 95 |
| 9 | 1 | 5 | 1 | 1147.347 | 7.6986 | 538.4211 | 0.813 | 0.759 | 0.622 | 2.19 | 8520 | 106 |
| 10 | 1 | 30 | 1 | 1149.108 | 7.6105 | 530.8281 | 0.785 | 0.819 | 0.685 | 2.28 | 7800 | 97 |
| 11 | 1 | 1 | 1 | 1149.436 | 7.6256 | 532.2451 | 0.780 | 0.808 | 0.673 | 2.26 | 8360 | 104 |
| **12** | **1** | **20** | **1** | **1162.251** | **7.3495** | **503.0716** | **0.575** | **0.995** | **0.915** | **2.48** | **7640** | **95** |
| 13 | 1 | 40 | 1 | 1169.323 | 7.3503 | 503.0457 | 0.462 | 0.995 | 0.915 | 2.37 | 7880 | 98 |
| 14 | 1 | 50 | 1 | 1169.884 | 7.3430 | 502.4701 | 0.453 | 1.000 | 0.920 | 2.37 | 7240 | 90 |
| 15 | 1 | 30 | 5 | 1187.431 | 7.3584 | 494.2004 | 0.173 | 0.989 | 0.988 | 2.15 | 7160 | 89 |
| 16 | 1 | 50 | 5 | 1187.72 | 7.3494 | 494.1329 | 0.168 | 0.995 | 0.989 | 2.15 | 6440 | 80 |
| 17 | 1 | 40 | 5 | 1187.932 | 7.3491 | 494.0940 | 0.165 | 0.995 | 0.989 | 2.15 | 7560 | 94 |
| 18 | 1 | 20 | 5 | 1188.146 | 7.3653 | 494.0481 | 0.161 | 0.984 | 0.989 | 2.13 | 6440 | 80 |
| 19 | 1 | 5 | 5 | 1188.269 | 7.3660 | 494.0222 | 0.159 | 0.984 | 0.990 | 2.13 | 7640 | 95 |
| 20 | 1 | 10 | 5 | 1188.543 | 7.3659 | 493.9609 | 0.155 | 0.984 | 0.990 | 2.13 | 7560 | 94 |
| 21 | 1 | 1 | 5 | 1188.676 | 7.3705 | 493.9427 | 0.153 | 0.981 | 0.990 | 2.12 | 7000 | 87 |
| 22 | 1 | 30 | 10 | 1192.469 | 7.3692 | 493.3742 | 0.092 | 0.982 | 0.995 | 2.07 | 7080 | 88 |
| 23 | 1 | 5 | 10 | 1193.375 | 7.3827 | 493.2310 | 0.078 | 0.973 | 0.996 | 2.04 | 6680 | 83 |
| 24 | 1 | 0 | 10 | 1193.416 | 7.3837 | 493.2232 | 0.077 | 0.972 | 0.996 | 2.04 | 6760 | 84 |
| 25 | 1 | 1 | 10 | 1193.416 | 7.3837 | 493.2232 | 0.077 | 0.972 | 0.996 | 2.04 | 6600 | 82 |
| 26 | 1 | 10 | 10 | 1193.851 | 7.3797 | 493.2192 | 0.070 | 0.975 | 0.996 | 2.04 | 7640 | 95 |
| 27 | 1 | 5 | 12 | 1194.751 | 7.3949 | 493.0908 | 0.056 | 0.964 | 0.997 | 2.01 | 7400 | 92 |
| 28 | 1 | 5 | 15 | 1195.998 | 7.4210 | 492.9527 | 0.036 | 0.947 | 0.999 | 1.98 | 6680 | 83 |
| 29 | 1 | 1 | 1 | 1196.013 | 7.4214 | 492.9515 | 0.036 | 0.946 | 0.999 | 1.98 | 8360 | 104 |
| 30 | 1 | 30 | 20 | 1196.061 | 7.4042 | 493.0000 | 0.035 | 0.958 | 0.998 | 1.99 | 7720 | 96 |
| 31 | 0 | 0 | 1 | 1196.543 | 7.4033 | 493.0023 | 0.027 | 0.959 | 0.998 | 1.98 | 6760 | 84 |
| 32 | 1 | 10 | 20 | 1196.55 | 7.4126 | 492.9514 | 0.027 | 0.952 | 0.999 | 1.97 | 5960 | 74 |
| 33 | 1 | 5 | 20 | 1197.19 | 7.4288 | 492.8858 | 0.017 | 0.941 | 0.999 | 1.95 | 7480 | 93 |
| 34 | 1 | 0 | 20 | 1197.45 | 7.4241 | 492.8929 | 0.013 | 0.945 | 0.999 | 1.95 | 7080 | 88 |
| 35 | 1 | 0 | 30 | 1198.268 | 7.4487 | 492.8321 | 0.000 | 0.928 | 1.000 | 1.92 | 6280 | 78 |

Following observations are made from Table 6.2.

$F_{Cmax}$=1198.29 $/hr., $F_{Lmax}$= 8.8214 MW, $F_{Emax}$ = 613.6401Kg/hr.

$F_{Cmin}$=1135.67 $/hr., $F_{Lmin}$=7.3429 MW, $F_{Emin}$ = 492.8321Kg/hr.

Ideal Point for 3-D space is represented by: **1135.67 $/hr., 7.3429 MW, 492.8321Kg/hr.**

It is seen that S.No.12 of Table 6.2 is the Target Point for IEEE 14 bus system in 3-D space for which **$F_C$=1162.251 $/hr., $F_L$= 7.3495 MW, $F_E$ = 503.0716 kg/hr.**

**TABLE 6.3**
**Results of MELD and Minimum Relative Attainments in 3-D Space**
**(IEEE 30 Bus System)**

| S. No. (1) | $W_1$ (2) | $W_2$ (3) | $W_3$ (4) | $F_C$ ($/hr) (5) | $F_L$ (MW) (6) | $F_E$ (kg/hr) (7) | $\tau_C$ (8) | $\tau_L$ (9) | $\tau_E$ (10) | $\sum\tau$ (11) | Kount (12) | k (13) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 20 | 1 | 1 | 1244.73 | 10.26 | 639.38 | 1.00 | 0.00 | 0.00 | 1.00 | 5160 | 64 |
| 2 | 10 | 1 | 1 | 1245.16 | 10.08 | 627.42 | 0.99 | 0.13 | 0.10 | 1.22 | 6840 | 85 |
| 3 | 10 | 0.5 | 1 | 1246.08 | 9.86 | 614.30 | 0.97 | 0.28 | 0.21 | 1.46 | 6680 | 83 |
| 4 | 1 | 0 | 0 | 1247.79 | 9.68 | 604.77 | 0.94 | 0.41 | 0.29 | 1.64 | 6680 | 83 |
| 5 | 1 | 1 | 0 | 1247.79 | 9.68 | 604.77 | 0.94 | 0.41 | 0.29 | 1.64 | 6840 | 85 |
| 6 | 1 | 0 | 0 | 1247.79 | 9.81 | 616.90 | 0.94 | 0.32 | 0.19 | 1.45 | 6680 | 83 |
| 7 | 1 | 0.5 | 0 | 1247.99 | 9.78 | 614.97 | 0.94 | 0.33 | 0.20 | 1.48 | 6920 | 86 |
| 8 | 1 | 10 | 0 | 1248.82 | 9.69 | 608.04 | 0.92 | 0.40 | 0.26 | 1.58 | 6840 | 85 |
| 9 | 3 | 0 | 1 | 1248.83 | 9.58 | 597.60 | 0.92 | 0.48 | 0.35 | 1.75 | 6680 | 82 |
| 10 | 2 | 0 | 1 | 1249.93 | 9.49 | 591.37 | 0.90 | 0.54 | 0.40 | 1.84 | 6920 | 86 |
| 11 | 1 | 20 | 0 | 1252.34 | 9.41 | 586.00 | 0.86 | 0.59 | 0.44 | 1.90 | 6760 | 84 |
| 12 | 10 | 0 | 3 | 1253.15 | 9.33 | 579.60 | 0.84 | 0.65 | 0.50 | 1.99 | 6680 | 83 |
| 13 | 12 | 0 | 4 | 1255.85 | 9.17 | 565.44 | 0.79 | 0.76 | 0.61 | 2.17 | 6680 | 83 |
| 14 | 18 | 0 | 9 | 1261.16 | 8.99 | 550.98 | 0.69 | 0.88 | 0.74 | 2.31 | 5960 | 74 |
| 15 | 16 | 0 | 9 | 1262.39 | 8.98 | 549.40 | 0.67 | 0.89 | 0.75 | 2.31 | 6840 | 85 |
| **16** | **0** | **0** | **1** | **1267.51** | **8.88** | **540.37** | **0.58** | **0.96** | **0.82** | **2.36** | **7160** | **89** |
| 17 | 1 | 0.5 | 1 | 1269.22 | 8.90 | 539.16 | 0.54 | 0.95 | 0.83 | 2.33 | 6920 | 86 |
| 18 | 1 | 200 | 0 | 1283.01 | 8.89 | 527.08 | 0.29 | 0.96 | 0.93 | 2.18 | 7160 | 89 |
| 19 | 1 | 200 | 1 | 1283.43 | 8.92 | 527.77 | 0.28 | 0.94 | 0.93 | 2.14 | 7160 | 89 |
| 20 | 1 | 0.5 | 20 | 1287.81 | 8.87 | 523.49 | 0.20 | 0.97 | 0.96 | 2.13 | 6920 | 86 |
| 21 | 1 | 200 | 10 | 1293.34 | 8.84 | 520.38 | 0.10 | 0.99 | 0.99 | 2.08 | 7160 | 89 |
| 22 | 1 | 0.5 | 30 | 1295.12 | 8.83 | 519.82 | 0.06 | 1.00 | 0.99 | 2.05 | 7160 | 89 |
| 23 | 1 | 0 | 40 | 1298.48 | 8.82 | 519.13 | 0.00 | 1.00 | 1.00 | 2.00 | 6920 | 86 |

Following observations are made from Table 6.3. It is seen that S. No. 16 of Table 6.3 is the Target Point for IEEE 30 bus system in 3-D space for which

**$F_C$=1267.51 $/hr., $F_L$= 8.88 MW, $F_E$ = 540.37 kg/hr.**

Noninferior set can also be displayed in 3-D space in terms of percentage distance from origin, where origin represents I.P. Columns (2), (3) and (4) of Tables 6.4, 6.5 and 6.6 show the weights $W_1$, $W_2$ and $W_3$ respectively. Columns (5), (6) and (7) show the $F_C$, $F_L$, $F_E$ . Columns (8), (9) and (10) of Tables 6.4, 6.5 and 6.6 show the percentage distance of $F_C$, $F_L$ and $F_E$ from the I.P. for IEEE 5, 14 and 30 bus system respectively.

Percentage distance from origin along x, y, and z-axis has been calculated using Following equations:

% Distance for $F_C$ from origin along x axis = $((F_C - F_{Cmin}) / (F_{Cmax} - F_{Cmin}))*100$

% Distance for $F_L$ from origin along y axis = $((F_L - F_{Lmin}) / (F_{Lmax} - F_{Lmin}))*100$

% Distance for $F_E$ from origin along z axis = $((F_E - F_{Emin}) / (F_{Emax} - F_{Emin}))*100$

Tables 6.4, 6.5 and 6.6 show the Noninferior set for IEEE 5,14 and 30 bus systems in terms of percentage distance from origin.

**TABLE 6.4**
**Noninferior Set of IEEE-5 Bus System in 3-D Space**

| S. No. (1) | $W_1$ (2) | $W_2$ (3) | $W_3$ (4) | $F_C$ ($/hr.) (5) | $F_L$ (MW) (6) | $F_E$ (Kg/hr.) (7) | % Distance of $F_C$ from origin along x- axis (8) | % Distance of $F_L$ from origin along y-axis (9) | % Distance of $F_E$ from origin along z-axis (10) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 0 | 0 | 760.9600 | 5.1835 | 129.6400 | 0.0000 | 100.0000 | 100.0000 |
| 2 | 100 | 100 | 0 | 760.9780 | 5.1599 | 128.3100 | 1.0039 | 81.0425 | 85.0688 |
| 3 | 100 | 200 | 0 | 760.9985 | 5.1507 | 127.7700 | 2.1472 | 73.6062 | 79.0324 |
| 4 | 100 | 420 | 0 | 761.0640 | 5.1334 | 126.7300 | 5.8003 | 59.6544 | 67.4067 |
| 5 | 100 | 0 | 10 | 761.2550 | 5.1075 | 125.0700 | 16.4529 | 38.8253 | 48.8503 |
| **6** | **100** | **0** | **20** | **761.6432** | **5.0823** | **123.2300** | **38.1037** | **18.4902** | **28.2817** |
| 7 | 100 | 3500 | 0 | 761.9130 | 5.0725 | 122.3800 | 53.1511 | 10.6026 | 18.7800 |
| 8 | 100 | 4600 | 0 | 762.0386 | 5.0692 | 122.0644 | 60.1562 | 7.9762 | 15.2520 |
| 9 | 118 | 2800 | 39 | 762.0600 | 5.0685 | 121.9920 | 61.3497 | 7.3800 | 14.4427 |
| 10 | 100 | 5000 | 0 | 762.0845 | 5.0682 | 121.9538 | 62.7161 | 7.1221 | 14.0157 |
| 11 | 100 | 0 | 30 | 762.2300 | 5.0650 | 121.6134 | 70.8310 | 4.5601 | 10.2105 |
| 12 | 100 | 5800 | 0 | 762.2519 | 5.0640 | 121.5830 | 72.0524 | 3.7560 | 9.8707 |
| 13 | 100 | 6000 | 0 | 762.2622 | 5.0647 | 121.5619 | 72.6269 | 4.3418 | 9.6348 |
| 14 | 160 | 200 | 77 | 762.2700 | 5.0640 | 121.5270 | 73.0619 | 3.7544 | 9.2447 |
| 15 | 160 | 40 | 80 | 762.3100 | 5.0600 | 121.4515 | 75.2928 | 0.5317 | 8.4007 |
| 16 | 160 | 75 | 83 | 762.3200 | 5.0637 | 121.4300 | 75.8505 | 3.5361 | 8.1603 |
| 17 | 180 | 180 | 95 | 762.3692 | 5.0631 | 121.3503 | 78.5945 | 3.0132 | 7.2694 |
| 18 | 160 | 70 | 85 | 762.4400 | 5.0621 | 121.2000 | 82.5432 | 2.2188 | 5.5893 |
| 19 | 0 | 100 | 0 | 762.7420 | 5.0594 | 120.7200 | 99.3865 | 0.0483 | 0.2236 |
| 20 | 0 | 100 | 20 | 762.7530 | 5.0593 | 120.7000 | 100.0000 | 0.0000 | 0.0000 |

## TABLE 6.5

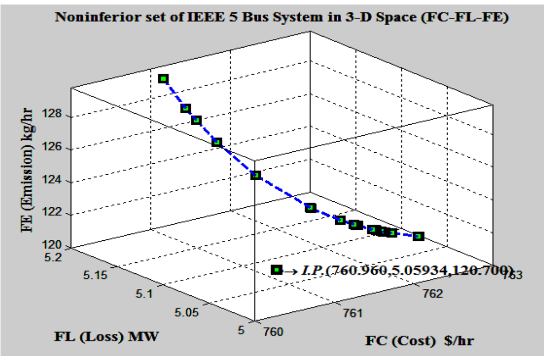
## Noninferior Set of IEEE-14 Bus System in 3-D Space

| S. No. (1) | $W_1$ (2) | $W_2$ (3) | $W_3$ (4) | $F_C$ ($/hr.) (5) | $F_L$ (MW) (6) | $F_E$ (Kg/hr.) (7) | % Distance of $F_C$ from origin along x-axis (8) | % Distance of $F_L$ from origin along y-axis (9) | % Distance of $F_E$ from origin along z-axis (10) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1135.6700 | 8.8214 | 613.2896 | 0.0000 | 100.0000 | 99.7099 |
| 2 | 1 | 0 | 0 | 1135.7360 | 8.8143 | 613.6401 | 0.1054 | 99.5205 | 100.0000 |
| 3 | 1 | 5 | 0 | 1136.7040 | 8.4788 | 592.5560 | 1.6518 | 76.8269 | 82.5474 |
| 4 | 1 | 10 | 0 | 1139.4980 | 8.1192 | 569.3447 | 6.1152 | 52.5016 | 63.3341 |
| 5 | 1 | 20 | 0 | 1142.4260 | 7.9063 | 554.1978 | 10.7927 | 38.1022 | 50.7961 |
| 6 | 1 | 30 | 0 | 1144.0290 | 7.8340 | 549.1883 | 13.3535 | 33.2112 | 46.6494 |
| 7 | 1 | 10 | 1 | 1145.6190 | 7.7277 | 538.7998 | 15.8935 | 26.0236 | 38.0502 |
| 8 | 1 | 20 | 1 | 1146.0910 | 7.7225 | 539.9504 | 16.6475 | 25.6696 | 39.0026 |
| 9 | 1 | 5 | 1 | 1147.3470 | 7.6986 | 538.4211 | 18.6540 | 24.0531 | 37.7367 |
| 10 | 1 | 30 | 1 | 1149.1080 | 7.6105 | 530.8281 | 21.4671 | 18.0959 | 31.4516 |
| 11 | 1 | 1 | 1 | 1149.1900 | 7.6295 | 532.5697 | 21.5981 | 19.3797 | 32.8932 |
| **12** | **1** | **20** | **1** | **1162.2510** | **7.3495** | **503.0716** | **42.4630** | **0.4390** | **8.4758** |
| 13 | 1 | 40 | 1 | 1169.3230 | 7.3503 | 503.0457 | 53.7605 | 0.4990 | 8.4544 |
| 14 | 1 | 50 | 1 | 1169.8840 | 7.3430 | 502.4701 | 54.6567 | 0.0000 | 7.9779 |
| 15 | 1 | 30 | 5 | 1187.4310 | 7.3584 | 494.2004 | 82.6880 | 1.0445 | 1.1326 |
| 16 | 1 | 50 | 5 | 1187.7200 | 7.3494 | 494.1329 | 83.1496 | 0.4378 | 1.0768 |
| 17 | 1 | 40 | 5 | 1187.9320 | 7.3491 | 494.0940 | 83.4883 | 0.4127 | 1.0446 |
| 18 | 1 | 20 | 5 | 1188.1460 | 7.3653 | 494.0481 | 83.8302 | 1.5079 | 1.0066 |
| 19 | 1 | 5 | 5 | 1188.2690 | 7.3660 | 494.0222 | 84.0267 | 1.5593 | 0.9851 |
| 20 | 1 | 10 | 5 | 1188.5430 | 7.3659 | 493.9609 | 84.4644 | 1.5532 | 0.9344 |
| 21 | 1 | 1 | 5 | 1188.6760 | 7.3705 | 493.9427 | 84.6768 | 1.8614 | 0.9193 |
| 22 | 1 | 30 | 10 | 1192.4690 | 7.3692 | 493.3742 | 90.7361 | 1.7732 | 0.4487 |
| 23 | 1 | 5 | 10 | 1193.3750 | 7.3827 | 493.2310 | 92.1835 | 2.6844 | 0.3302 |
| 24 | 1 | 0 | 10 | 1193.4160 | 7.3837 | 493.2232 | 92.2490 | 2.7546 | 0.3237 |
| 25 | 1 | 1 | 10 | 1193.4160 | 7.3837 | 493.2232 | 92.2490 | 2.7546 | 0.3237 |
| 26 | 1 | 10 | 10 | 1193.8510 | 7.3797 | 493.2192 | 92.9439 | 2.4873 | 0.3204 |
| 27 | 1 | 5 | 12 | 1194.7510 | 7.3949 | 493.0908 | 94.3816 | 3.5096 | 0.2141 |
| 28 | 1 | 5 | 15 | 1195.9980 | 7.4210 | 492.9527 | 96.3737 | 5.2813 | 0.0998 |
| 29 | 1 | 1 | 1 | 1196.0130 | 7.4214 | 492.9515 | 96.3977 | 5.3062 | 0.0988 |
| 30 | 1 | 30 | 20 | 1196.0610 | 7.4042 | 493.0000 | 96.4743 | 4.1410 | 0.1390 |
| 31 | 0 | 0 | 1 | 1196.5430 | 7.4033 | 493.0023 | 97.2443 | 4.0823 | 0.1409 |
| 32 | 1 | 10 | 20 | 1196.5500 | 7.4126 | 492.9514 | 97.2555 | 4.7110 | 0.0988 |
| 33 | 1 | 5 | 20 | 1197.1900 | 7.4288 | 492.8858 | 98.2779 | 5.8089 | 0.0445 |
| 34 | 1 | 0 | 20 | 1197.4500 | 7.4241 | 492.8929 | 98.6933 | 5.4870 | 0.0503 |
| 35 | 1 | 0 | 30 | 1198.2680 | 7.4487 | 492.8321 | 100.0000 | 7.1490 | 0.0000 |

## TABLE 6.6
## Noninferior Set of IEEE - 30 Bus System in 3-D Space

| S. No. (1) | $W_1$ (2) | $W_2$ (3) | $W_3$ (4) | $F_C$ ($/hr.) (5) | $F_L$ (MW) (6) | $F_E$ (Kg/hr.) (7) | % Distance for $F_C$ From origin along x-axis (8) | % Distance for $F_L$ from origin along y-axis (9) | % Distance for $F_E$ from origin along z-axis (10) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 20 | 1 | 1 | 1244.729 | 10.2626 | 639.3834 | 0.0000 | 100.0000 | 100.0000 |
| 2 | 10 | 1 | 1 | 1245.158 | 10.0756 | 627.4210 | 0.7981 | 87.0015 | 90.0525 |
| 3 | 10 | 0.50 | 1 | 1246.079 | 9.8644 | 614.3042 | 2.5117 | 72.3173 | 79.1452 |
| 4 | 1 | 0 | 0 | 1247.792 | 9.6774 | 604.7701 | 5.6989 | 59.3166 | 71.2170 |
| 5 | 1 | 1 | 0 | 1247.792 | 9.6774 | 604.7701 | 5.6989 | 59.3166 | 71.2170 |
| 6 | 1 | 0 | 0 | 1247.794 | 9.8082 | 616.8960 | 5.7026 | 68.4079 | 81.3004 |
| 7 | 1 | 0.50 | 0 | 1247.994 | 9.7817 | 614.9666 | 6.0747 | 66.5691 | 79.6960 |
| 8 | 1 | 10 | 0 | 1248.823 | 9.6880 | 608.0366 | 7.6171 | 60.0500 | 73.9333 |
| 9 | 3 | 0 | 1 | 1248.830 | 9.5769 | 597.5952 | 7.6301 | 52.3277 | 65.2507 |
| 10 | 2 | 0 | 1 | 1249.925 | 9.4915 | 591.3659 | 9.6675 | 46.3874 | 60.0707 |
| 11 | 1 | 20 | 0 | 1252.344 | 9.4086 | 585.9975 | 14.1682 | 40.6222 | 55.6066 |
| 12 | 10 | 0 | 3 | 1253.146 | 9.3337 | 579.5993 | 15.6604 | 35.4172 | 50.2861 |
| 13 | 12 | 0 | 4 | 1255.852 | 9.1662 | 565.4351 | 20.6951 | 23.7699 | 38.5078 |
| 14 | 18 | 0 | 9 | 1261.155 | 8.9907 | 550.9842 | 30.5617 | 11.5659 | 26.4910 |
| 15 | 16 | 0 | 9 | 1262.390 | 8.9823 | 549.4000 | 32.8595 | 10.9879 | 25.1742 |
| **16** | **0** | **0** | **1** | **1267.511** | **8.8756** | **540.3710** | **42.3874** | **3.5658** | **17.6661** |
| 17 | 1 | 0.5 | 1 | 1269.219 | 8.8970 | 539.1601 | 45.565334 | 5.054424 | 16.6586 |
| 18 | 1 | 200 | 0 | 1283.008 | 8.8865 | 527.0773 | 71.220719 | 4.323464 | 6.6111 |
| 19 | 1 | 200 | 1 | 1283.425 | 8.9171 | 527.7670 | 71.996577 | 6.449179 | 7.1846 |
| 20 | 1 | 0.5 | 20 | 1287.813 | 8.8725 | 523.4930 | 80.160753 | 3.347274 | 3.6305 |
| 21 | 1 | 200 | 10 | 1293.343 | 8.8387 | 520.3835 | 90.449700 | 0.997396 | 1.0448 |
| 22 | 1 | 0.5 | 30 | 1295.116 | 8.8274 | 519.8162 | 93.748488 | 0.216862 | 0.5731 |
| 23 | 1 | 0 | 40 | 1298.476 | 8.8243 | 519.1279 | 100.0000 | 0.0000 | 0.00075 |

For IEEE 5 bus system, the noninferior set for all the three objectives has been displayed in 3-D space in Fig.6.3 and for various combinations of two objectives: $F_C$-$F_L$, $F_C$-$F_E$, and $F_L$-$F_E$ have been displayed in Fig. 6.4, 6.5 and 6.6 respectively in 2-D space.



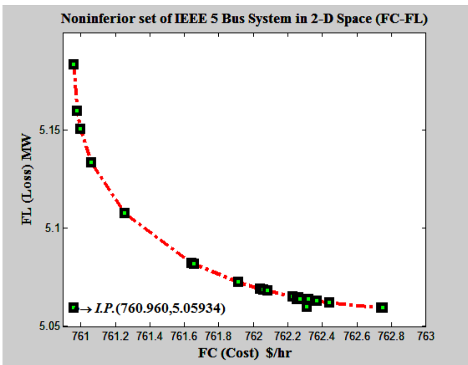
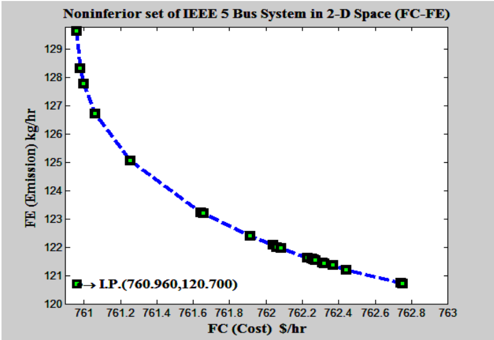




**Fig.6.3** Noninferior set of IEEE 5 bus system in 3-D

**Fig.6.4** Noninferior set of IEEE 5 bus system



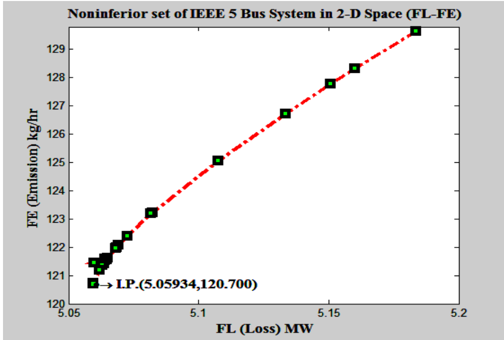




**Fig.6.5** Noninferior set of IEEE 5 bus system

**Fig.6.6** Noninferior set of IEEE 5 bus system

For IEEE 5 bus system, Fig.6.3 shows the variation of cost of generation ($F_C$) on x-axis, transmission losses ($F_L$) on y-axis and environmental emission ($F_E$) on z-axis. The 3-D curve shows the behaviour of $F_C$, $F_L$ and $F_E$ in 3-D space. It is observed from Table 6.1 and Fig. 6.3 that when cost of generation increases, transmission losses as well as environmental emissions decrease. Thus, it is concluded that cost of generation is conflicting with both the objectives- transmission losses and environmental emissions, whereas environmental emissions and transmission losses are supportive in behaviour. The behaviour of all the three objectives can be explicitly observed in 2-D space from Fig.6.4, 6.5 and 6.6.

Fig.6.4 represents the variation of cost of generation ($F_C$) and transmission losses ($F_L$) in 2-D space. The curve shows that when cost of generation increases then transmission losses decrease. It means $F_C$ and $F_L$ are conflicting in behaviour.

Fig.6.5 represents the variation of cost of generation ($F_C$) and Emission ($F_E$) in 2-D space. The curve shows that when cost of generation increases the emission decreases i.e. $F_C$ and $F_E$ are conflicting.

Fig.6.6 represents the variation of transmission losses ($F_L$) and emission ($F_E$) in 2-D space. The curve shows that when the transmission losses decreases, emission also decreases i.e. $F_L$ and $F_E$ are supportive in behaviour.

For IEEE 14 bus system, the noninferior set for all the three objectives has been displayed in 3-D space in Fig.6.7 and for various combinations of two objectives: $F_C$-$F_L$, $F_C$-$F_E$ and $F_L$-$F_E$ have been displayed in Fig.6.8, 6.9 and 6.10 respectively in 2-D space.

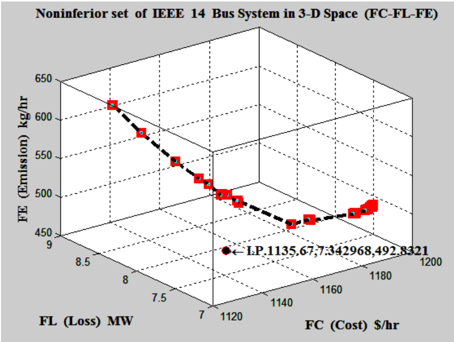



**Fig.6.7** Noninferior set of IEEE14 bus system in 3-D

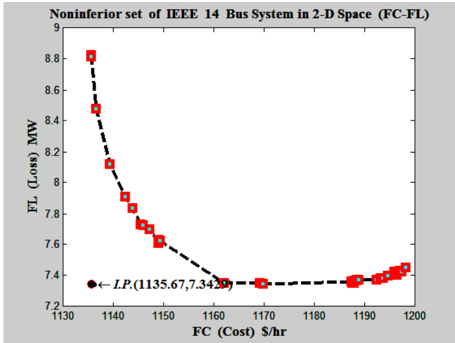



**Fig.6.8** Noninferior set of IEEE 14 bus system

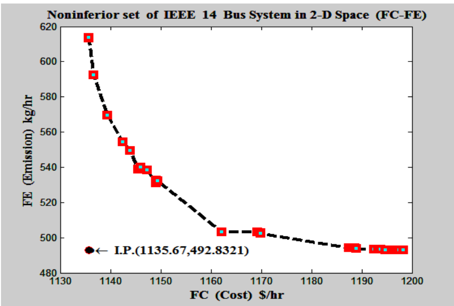



**Fig.6.9** Noninferior set of IEEE 14 bus system

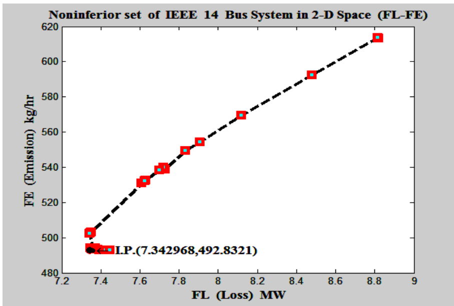



**Fig.6.10** Noninferior set of IEEE 14 bus system

For IEEE 30 bus system, the noninferior set for all the three objectives has been displayed in 3 - D space in Fig.6.11 and for various combinations of two objectives: $F_C$-$F_L$, $F_C$-$F_E$, and $F_L$-$F_E$ have been displayed in Fig.6.12, 6.13 and 6.14 respectively in 2-D space.

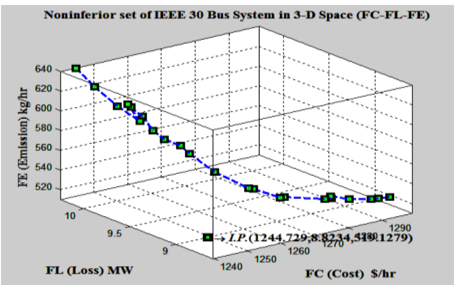



**Fig.6.11** Noninferior set of IEEE 30 Bus system

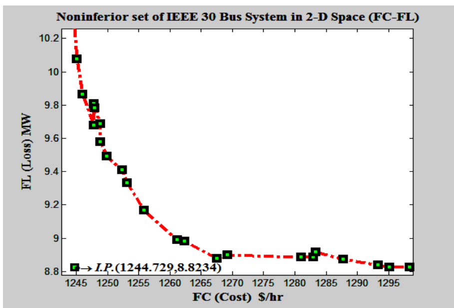
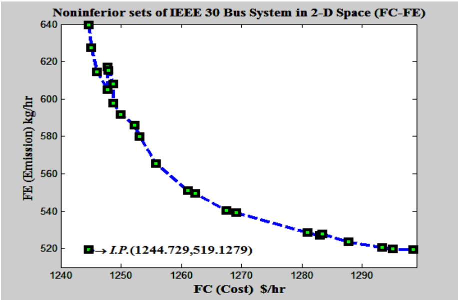



**Fig.6.12** Noninferior set of IEEE 30 Bus system



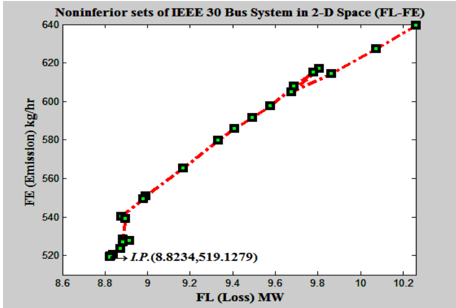




**Fig.6.13** Noninferior set of IEEE 30 Bus system **Fig.6.14** Noninferior set of IEEE 30 Bus system

Fig.6.11shows the variation of cost of generation ($F_C$) on x-axis, transmission losses ($F_L$) on y-axis and environmental emission ($F_E$) on z-axis. The 3-D curve shows the behaviour of $F_C$, $F_L$ and $F_E$ in 3-D space.

It is observed from 3-D curve of all the systems that cost of generation is conflicting with both the objectives- transmission losses and environmental emission, whereas environmental emission and transmission losses are supportive in behaviour. The behaviour of all the three objectives can be explicitly observed in 2-D space from Fig.6.12, 6.13 and 6.14. The 2-D curves show that when cost of generation increases the transmission losses decreases i.e. $F_C$ & $F_L$ are conflicting in behaviour, $F_C$ & $F_E$ are conflicting and $F_L$ & $F_E$ are supportive in behaviour. Target point using the Minimum Relative Attainments for Noninferior set of IEEE 5 bus, IEEE 14 bus and IEEE 30 bus systems is shown in Fig.6.15, 6.16 and 6.17 respectively.

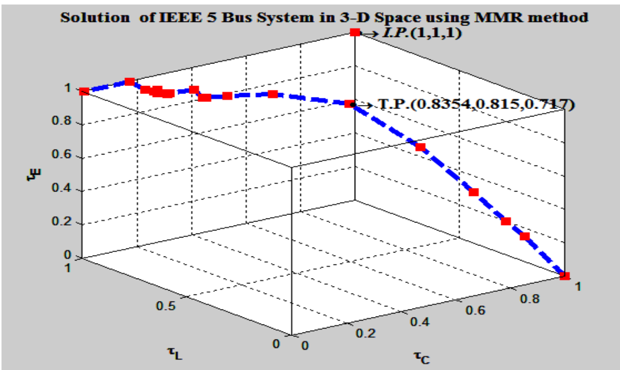



**Fig.6.15** Minimum Relative Attainments of Noninferior Set of IEEE 5 Bus System

Fig. 6.15 represents the variation of minimum relative attainments for cost of generation ($\tau_C$), transmission losses ($\tau_L$), and environmental emissions ($\tau_E$) in 3-D space. The minimum relative attainment of an objective increases with decrease in the value of that objective. I.P. represents the Ideal Point ($F_{Cmin}$, $F_{Lmin}$, $F_{Emin}$) where the relative attainment of each objective is maximum and is equal to 1. In this Fig. T.P. represent the target point. At T.P. the relative attainments $\tau_C, \tau_L, \tau_E$ are 0.84, 0.82, 0.717183 respectively and $F_C$, $F_L$, $F_E$ are **761.643 $/hr., 5.082 MW and 123.230 Kg/hr.** respectively. Fig.6.16 shows the minimum relative attainments of Noninferior set of IEEE 14 bus system.

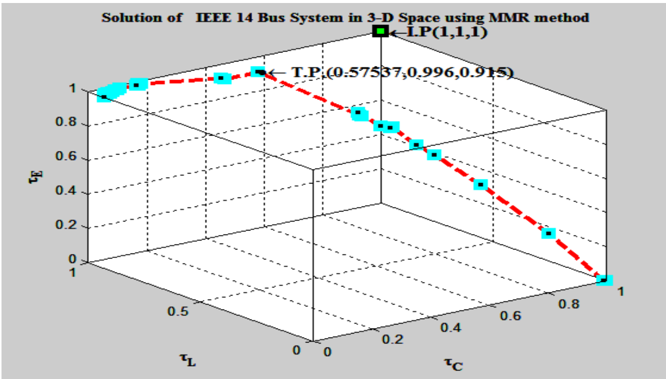



**Fig.6.16** Minimum Relative Attainments of Noninferior Set of IEEE 14 Bus System

At T.P. (Target Point) the relative attainments $\tau_C, \tau_L, \tau_E$ are **0.57, 0.99, 0.91** respectively and $F_C$, $F_L$, $F_E$ are **1162.25 $/hr., 7.349 MW and 503.0716 Kg/hr.** respectively.Fig.6.17 shows the minimum relative attainments of noninferior set of IEEE 30 bus system.

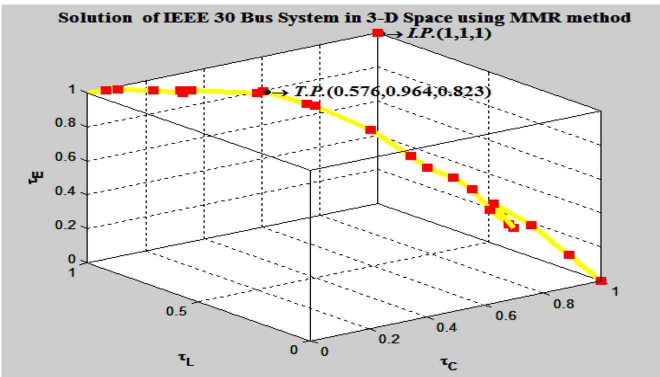



**Fig.6.17** Minimum Relative Attainments of Noninferior Set of IEEE 30 Bus System

At T.P. the relative attainments $\tau_C, \tau_L, \tau_E$ are **0.576, 0.9643, 0.8233** respectively and $F_C$, $F_L$, $F_E$ are **1267.511 $/hr., 8.8756 MW, and 540.376 Kg/hr.** respectively. Noninferior set can also be displayed in 3-D space in terms of percentage distance from origin, where origin represents I.P. Column (8), (9) and (10) of Tables (6.4), (6.5) and (6) show the percentage distance of $F_C$, $F_L$ and $F_E$ from the I.P. for IEEE 5,14 and 30 bus system respectively. The same has also been displayed in Fig. 6.18, 6.19 and 6.20. Fig. 6.18 show the Noninferior set in terms of percentage distance of each objective from the origin (I.P.) for IEEE 5 bus system.

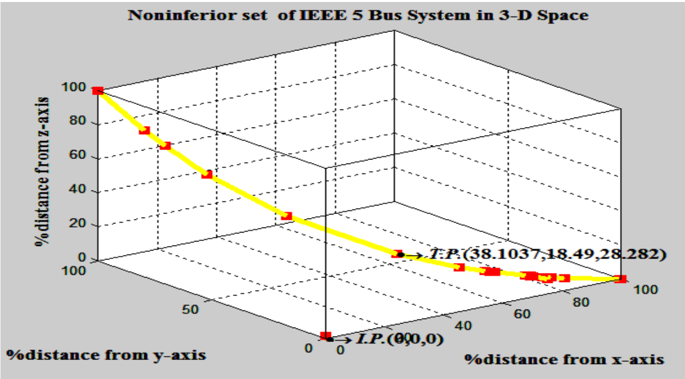



**Fig. 6.18** Noninferior set of IEEE 5 bus system in 3-D Space

Fig. 6.19 shows the Noninferior set in terms of percentage distance of each objective from the origin (I.P.) for IEEE 14 bus system.

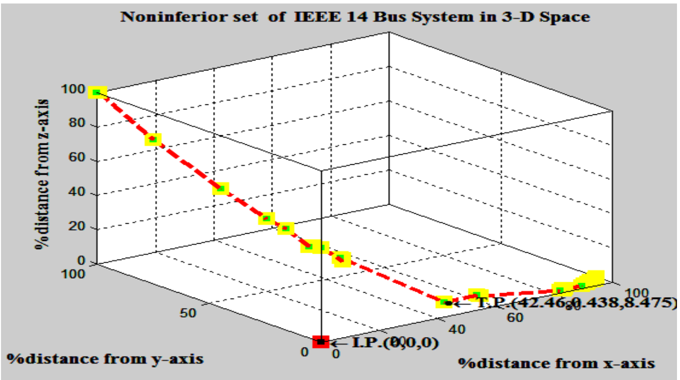



**Fig. 6.19 Noninferior** set of IEEE 14 bus system in 3-D Space

For IEEE 30 bus system, Fig. 6.20 shows the Noninferior set in terms of percentage distance of each objective from the origin (I.P.).

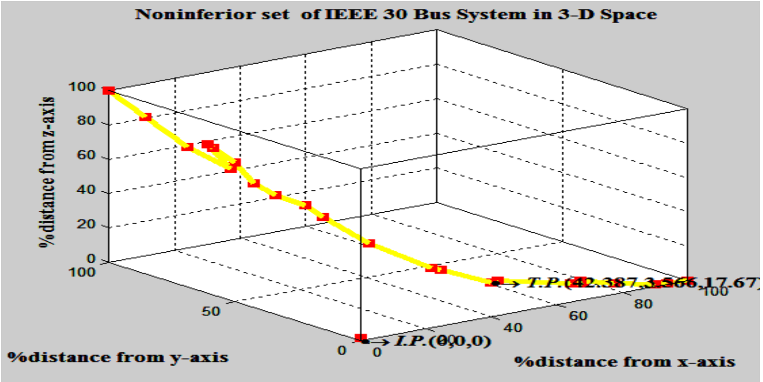



**Fig. 6.20** Noninferior set of IEEE 30 bus system in 3D space

Table 6.7 shows comparison of Target Points achieved for GA [176] and PSO algorithms for IEEE 5 bus, IEEE 14 bus and IEEE 30 bus systems.

**TABLE 6.7**
**Target Points achieved by GA and PSO algorithm**

| | IEEE 5 BUS | | IEEE 14BUS | | IEEE 30 BUS | |
|---|---|---|---|---|---|---|
| | **PSO** | **GA** | **PSO** | **GA** | **PSO** | **GA** |
| $F_C$ ($/hr.) | 761.643 | 761.688 | 1162.251 | 1163.28 | 1267.511 | 1273.65 |
| $F_L$ (MW) | 5.0822 | 5.080 | 7.3494 | 7.28 | 8.8756 | 9.44 |
| $F_E$ (Kg/hr.) | 123.2300 | 123 | 503.0716 | 508 | 540.3717 | 538 |

It is observed from Table 6.7 that PSO algorithm and GA give comparable results in terms of cost of generation for IEEE 5 bus, 14 bus and 30 bus systems.

## 6.6.2 Target Point Using Fuzzy Logic System

Fuzzy Set theory has been used to determine the best compromise solution or Target point from the noninferior set. The power system operator has fuzzy goals for each objective function. The fuzzy sets defined by equations called membership functions. The membership value (0, 1) indicates the degree to which an objective is satisfied. The higher the membership value the greater the satisfaction with a solution. The multiple objectives are modelled using fuzzy sets. The membership function for each objective is to be strictly monotonically decreasing and continuous function.

It is defined as

$\mu_{Fi} = 1 \qquad F_i \le F_{imin}$

$\mu_{Fi} = (F_{imax} - F_i)/(F_{imax} - F_{imin}) \qquad F_{imin} \le F_i \le F_{imax}$

$\mu_{Fi} = 0 \qquad F_i \ge F_{imax}$

The achievement of each solution can be computed by taking sum of the membership function values for all the objectives ($\mu_{Fi}$, $i = 1,2,3 \ldots$). The Target point is achieved by normalizing the achievement of each noninferior solution over the sum of the achievements of all the objectives as follows:

$$\mu^k = \left(\sum_{i=1}^{OG} \mu_{Fi}\right) / \left(\sum_{k=1}^{No} \sum_{i=1}^{OG} \mu^k\right) \qquad (6.14)$$

OG Number of objectives in the function.

No Represents the number of solutions in the noninferior set.

The $\mu^k$ represents membership function of $k^{th}$ solution.

**TABLE 6.8**
**Results of MELD and T.P. using Fuzzy Logic System**
**(IEEE 5 Bus System)**

| S. No (1) | $F_C$ (\$/hr.) (2) | $F_L$ (MW) (3) | $F_E$ (Kg/hr.) (4) | $\mu_C$ (5) | $\mu_L$ (6) | $\mu_E$ (7) | $\sum\mu$ (8) | $\mu^k$ (9) |
|---|---|---|---|---|---|---|---|---|
| 1 | 760.960 | 5.18 | 129.646 | 1 | 0 | 0.00 | 1.00 | 0.025 |
| 2 | 760.978 | 5.16 | 128.310 | 1 | 0.19 | 0.15 | 1.34 | 0.033 |
| 3 | 760.999 | 5.15 | 127.770 | 0.99 | 0.26 | 0.21 | 1.46 | 0.036 |
| 4 | 761.064 | 5.13 | 126.730 | 0.98 | 0.4 | 0.33 | 1.71 | 0.042 |
| 5 | 761.255 | 5.11 | 125.070 | 0.94 | 0.61 | 0.51 | 2.07 | 0.051 |
| **6** | **761.643** | **5.08** | **123.230** | **0.84** | **0.82** | **0.72** | **2.37** | **0.059** |
| 7 | 761.913 | 5.07 | 122.380 | 0.62 | 0.89 | 0.81 | 2.33 | 0.058 |
| 8 | 762.039 | 5.07 | 122.064 | 0.47 | 0.92 | 0.85 | 2.24 | 0.056 |
| 9 | 762.060 | 5.07 | 121.992 | 0.4 | 0.93 | 0.86 | 2.18 | 0.054 |
| 10 | 762.085 | 5.07 | 121.954 | 0.39 | 0.93 | 0.86 | 2.18 | 0.054 |
| 11 | 762.230 | 5.07 | 121.613 | 0.37 | 0.95 | 0.90 | 2.23 | 0.055 |
| 12 | 762.252 | 5.06 | 121.583 | 0.29 | 0.96 | 0.90 | 2.16 | 0.053 |
| 13 | 762.262 | 5.06 | 121.562 | 0.28 | 0.96 | 0.90 | 2.14 | 0.053 |
| 14 | 762.270 | 5.06 | 121.527 | 0.27 | 0.96 | 0.91 | 2.14 | 0.053 |
| 15 | 762.310 | 5.06 | 121.452 | 0.27 | 0.99 | 0.92 | 2.18 | 0.054 |
| 16 | 762.320 | 5.06 | 121.430 | 0.25 | 0.96 | 0.92 | 2.13 | 0.053 |
| 17 | 762.369 | 5.06 | 121.350 | 0.24 | 0.97 | 0.93 | 2.14 | 0.053 |
| 18 | 762.440 | 5.06 | 121.200 | 0.21 | 0.98 | 0.94 | 2.14 | 0.053 |
| 19 | 762.742 | 5.06 | 120.720 | 0.17 | 1 | 1.00 | 2.17 | 0.054 |
| 20 | 762.753 | 5.06 | 120.700 | 0.01 | 1 | 1.00 | 2.01 | 0.050 |

The Noninferior set has been achieved using Particle Swarm Optimization Technique using weighting method. The Ideal Points for IEEE 5 bus system in 3-D space are 760.96 $/hr., 5.06 MW, 120.70 kg. / hr. However, Ideal Point is an infeasible point and cannot be achieved in practice.

The Target Point (T.P.) obtained by Fuzzy Logic System for IEEE 5 bus system is **761.643 $/hr., 5.08229 MW, 123.230 kg/hr.** respectively.

**TABLE 6.9**
**Results of MELD and T.P. using Fuzzy Logic System**
**(IEEE 14 Bus System)**

| S. No. | $F_C$ ($/hr.) (2) | $F_L$ (MW) (3) | $F_E$ (Kg/hr.) (4) | $\mu_C$ (5) | $\mu_L$ (6) | $\mu_E$ (7) | $\sum\mu$ (8) | $\mu^k$ (9) |
|---|---|---|---|---|---|---|---|---|
| 1 | 1135.670 | 8.821 | 613.290 | 1.000 | 0.000 | 0.003 | 1.003 | 0.014 |
| 2 | 1135.736 | 8.814 | 613.640 | 0.999 | 0.005 | 0.000 | 1.004 | 0.014 |
| 3 | 1136.704 | 8.479 | 592.556 | 0.983 | 0.232 | 0.175 | 1.390 | 0.020 |
| 4 | 1139.498 | 8.119 | 569.345 | 0.939 | 0.475 | 0.367 | 1.780 | 0.025 |
| 5 | 1142.426 | 7.906 | 554.198 | 0.892 | 0.619 | 0.492 | 2.003 | 0.028 |
| 6 | 1144.029 | 7.834 | 549.188 | 0.866 | 0.668 | 0.534 | 2.068 | 0.029 |
| 7 | 1145.619 | 7.728 | 538.800 | 0.841 | 0.740 | 0.619 | 2.200 | 0.031 |
| 8 | 1146.091 | 7.722 | 539.950 | 0.834 | 0.743 | 0.610 | 2.187 | 0.031 |
| 9 | 1147.347 | 7.699 | 538.421 | 0.813 | 0.759 | 0.623 | 2.196 | 0.031 |
| 10 | 1149.108 | 7.611 | 530.828 | 0.785 | 0.819 | 0.685 | 2.290 | 0.032 |
| 11 | 1149.436 | 7.626 | 532.245 | 0.780 | 0.809 | 0.674 | 2.263 | 0.032 |
| **12** | **1162.251** | **7.349** | **503.072** | **0.575** | **0.996** | **0.915** | **2.486** | **0.035** |
| 13 | 1169.323 | 7.350 | 503.046 | 0.462 | 0.995 | 0.915 | 2.373 | 0.034 |
| 14 | 1169.884 | 7.343 | 502.470 | 0.453 | 1.000 | 0.920 | 2.374 | 0.034 |
| 15 | 1187.431 | 7.358 | 494.200 | 0.173 | 0.990 | 0.989 | 2.151 | 0.030 |
| 16 | 1187.720 | 7.349 | 494.133 | 0.169 | 0.996 | 0.989 | 2.153 | 0.030 |
| 17 | 1187.932 | 7.349 | 494.094 | 0.165 | 0.996 | 0.990 | 2.151 | 0.030 |
| 18 | 1188.146 | 7.365 | 494.048 | 0.162 | 0.985 | 0.990 | 2.137 | 0.030 |
| 19 | 1188.269 | 7.366 | 494.022 | 0.160 | 0.984 | 0.990 | 2.134 | 0.030 |
| 20 | 1188.543 | 7.366 | 493.961 | 0.155 | 0.984 | 0.991 | 2.130 | 0.030 |
| 21 | 1188.676 | 7.370 | 493.943 | 0.153 | 0.981 | 0.991 | 2.125 | 0.030 |
| 22 | 1192.469 | 7.369 | 493.374 | 0.093 | 0.982 | 0.996 | 2.070 | 0.029 |
| 23 | 1193.375 | 7.383 | 493.231 | 0.078 | 0.973 | 0.997 | 2.048 | 0.029 |
| 24 | 1193.416 | 7.384 | 493.223 | 0.078 | 0.972 | 0.997 | 2.047 | 0.029 |
| 25 | 1193.416 | 7.384 | 493.223 | 0.078 | 0.972 | 0.997 | 2.047 | 0.029 |
| 26 | 1193.851 | 7.380 | 493.219 | 0.071 | 0.975 | 0.997 | 2.042 | 0.029 |
| 27 | 1194.751 | 7.395 | 493.091 | 0.056 | 0.965 | 0.998 | 2.019 | 0.029 |
| 28 | 1195.998 | 7.421 | 492.953 | 0.036 | 0.947 | 0.999 | 1.982 | 0.028 |
| 29 | 1196.013 | 7.421 | 492.952 | 0.036 | 0.947 | 0.999 | 1.982 | 0.028 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 30 | 1196.061 | 7.404 | 493.000 | 0.035 | 0.959 | 0.999 | 1.992 | 0.028 |
| 31 | 1196.543 | 7.403 | 493.002 | 0.028 | 0.959 | 0.999 | 1.985 | 0.028 |
| 32 | 1196.550 | 7.413 | 492.951 | 0.027 | 0.953 | 0.999 | 1.979 | 0.028 |
| 33 | 1197.190 | 7.429 | 492.886 | 0.017 | 0.942 | 1.000 | 1.959 | 0.028 |
| 34 | 1197.450 | 7.424 | 492.893 | 0.013 | 0.945 | 0.999 | 1.958 | 0.028 |
| 35 | 1198.268 | 7.449 | 492.832 | 0.000 | 0.929 | 1.000 | 1.929 | 0.027 |

The Ideal Points for IEEE 14 bus system in 3-D space are 1135.67 \$/hr.,7.3427 MW, 492.8321 kg. /hr.

Target point for IEEE 14 bus system is **1162.251 \$/hr. 7.349 MW, 503.716 kg/hr.**

**TABLE 6.10**
**Results of MELD and T.P. using Fuzzy Logic System**
**(IEEE 30 Bus System)**

| **S. No. (1)** | **$F_C$ (\$/hr.) (2)** | **$F_L$ (MW) (3)** | **$F_E$ (Kg/hr.) (4)** | **$\mu_C$ (5)** | **$\mu_L$ (6)** | **$\mu_E$ (7)** | **$\sum\mu$ (8)** | **$\mu^k$ (9)** |
|---|---|---|---|---|---|---|---|---|
| 1 | 1244.729 | 10.263 | 639.383 | 1.000 | 0.000 | 0.000 | 1.000 | 0.023 |
| 2 | 1245.158 | 10.076 | 627.421 | 0.992 | 0.130 | 0.099 | 1.221 | 0.028 |
| 3 | 1246.079 | 9.864 | 614.304 | 0.975 | 0.277 | 0.209 | 1.460 | 0.034 |
| 4 | 1247.792 | 9.677 | 604.770 | 0.943 | 0.407 | 0.288 | 1.638 | 0.038 |
| 5 | 1247.792 | 9.677 | 604.770 | 0.943 | 0.407 | 0.288 | 1.638 | 0.038 |
| 6 | 1247.794 | 9.808 | 616.896 | 0.943 | 0.316 | 0.187 | 1.446 | 0.034 |
| 7 | 1247.994 | 9.782 | 614.967 | 0.939 | 0.334 | 0.203 | 1.477 | 0.034 |
| 8 | 1248.823 | 9.688 | 608.037 | 0.924 | 0.399 | 0.261 | 1.584 | 0.037 |
| 9 | 1248.830 | 9.577 | 597.595 | 0.924 | 0.477 | 0.347 | 1.748 | 0.041 |
| 10 | 1249.925 | 9.492 | 591.366 | 0.903 | 0.536 | 0.399 | 1.839 | 0.043 |
| 11 | 1252.344 | 9.409 | 585.998 | 0.858 | 0.594 | 0.444 | 1.896 | 0.044 |
| 12 | 1253.146 | 9.334 | 579.599 | 0.843 | 0.646 | 0.497 | 1.986 | 0.046 |
| 13 | 1255.852 | 9.166 | 565.435 | 0.793 | 0.762 | 0.615 | 2.170 | 0.050 |
| 14 | 1261.155 | 8.991 | 550.984 | 0.694 | 0.884 | 0.735 | 2.314 | 0.054 |
| 15 | 1262.390 | 8.982 | 549.401 | 0.671 | 0.890 | 0.748 | 2.310 | 0.054 |
| **16** | **1267.511** | **8.876** | **540.372** | **0.576** | **0.964** | **0.823** | **2.364** | **0.055** |
| 17 | 1269.219 | 8.897 | 539.160 | 0.544 | 0.949 | 0.833 | 2.327 | 0.054 |
| 18 | 1283.008 | 8.887 | 527.077 | 0.288 | 0.957 | 0.934 | 2.178 | 0.051 |
| 19 | 1283.425 | 8.917 | 527.767 | 0.280 | 0.936 | 0.928 | 2.144 | 0.050 |
| 20 | 1287.813 | 8.873 | 523.493 | 0.198 | 0.967 | 0.964 | 2.129 | 0.050 |
| 21 | 1293.343 | 8.839 | 520.384 | 0.096 | 0.990 | 0.990 | 2.075 | 0.048 |
| 22 | 1295.116 | 8.827 | 519.816 | 0.063 | 0.998 | 0.994 | 2.055 | 0.048 |
| 23 | 1298.476 | 8.824 | 519.128 | 0.000 | 1.000 | 1.000 | 2.000 | 0.047 |

The Ideal Points for IEEE 30 bus systems in 3-D space is 1244.729 \$/hr., 8.8243 MW, 519.1279 kg. /hr. However, Ideal Point is an infeasible point and cannot be achieved in practice.

The Target Point has been also obtained by Fuzzy Logic System for IEEE 30 bus system is **1267.511 \$/hr., 8.8756 MW, 540. 172 kg/hr.**

## 6.8 CONSTRAINT METHOD

The constraint method is the most appealing generating technique. It operates by optimizing one objective while all others are constrained to some value.

Mathematically, a multiobjective problem with h objectives is formulated using Constraint method as:

Minimize

$$Z_t\ (x_1, x_2, x_3, \ldots\ldots\ldots\ldots, x_n) \qquad (6.15)$$

subject to

$$g_1\ (x_1, x_2, x_3, \ldots\ldots\ldots\ldots, x_n) \leq 0 \qquad i=1,2,3,\ldots.m \qquad (6.16)$$

$$x_j \geq 0 \qquad j=1,2,3\ldots..\ n \qquad (6.17)$$

$$Z_k\ (x_1, x_2, x_3, \ldots\ldots\ldots\ldots, x_n) \leq L_k \qquad k = 1,2,3\ldots.t\text{-}1,t+1\ldots h \qquad (6.18)$$

Where the $t^{th}$ objective is arbitrarily chosen for minimization. This is a single objective formulation.

### 6.8.1 FORMULATION OF MULTIOBJECTIVE ECONOMIC LOAD DISPATCH (MELD) PROBLEM

The MELD problem using the Constraint method [176] for generating noninferior solutions is formulated as:

Minimize

$$F_C = \sum_{i=1}^{NG} F(C_i(P_{gi})]$$

Subject to Equality constraint and inequality constraint of the systems defined by equations (6.11) and (6.12) respectively.

$$F_L \leq L_i$$

Where $L_i$ lies between $F_{Lmin}$ and $F_{LatFcmin}$ and is expressed as

$$F_{Limin} \leq L_i \leq F_{LatFcimin}$$

$F_{Limin}$ : The value of system transmission losses obtained by individually minimizing $F_L$.

$F_{LatFcmin}$: The value of system transmission losses obtained by individually minimizing $F_C$ .

### 6.8.2 COMPUTATIONAL PROCEDURE

The noninferior set for MELD problem has been obtained by Genetic Algorithm (GA). The cost of Generation is minimized keeping system transmission losses fixed to various values between $F_{Limin}$ and $F_{LatFcmin}$.

### 6.8.3 COMPUTATIONAL RESULTS

The noninferior set for IEEE 5, 14 and 30 bus systems are shown in Tables 6.11, 6.13,6.15 respectively. The relative attainments of the IEEE 5, 14 and 30 systems by using MMR are shown in Tables 6.12, 6.14 and 6.16 respectively.

**TABLE 6.11**
**Noninferior Set of IEEE 5 Bus System in 2-D Space**

| S. No | $F_L$ (MW) (Fixed) | k | $P_1$ (MW) | $P_3$ (MW) | $F_C$ ($/hr.) |
|---|---|---|---|---|---|
| 1 | **5.05** | 58 | 84.367 | 80.691 | **763.158** |
| 2 | 5.055 | 74 | 84.374 | 80.684 | 763.157 |
| 3 | 5.06 | 39 | 85.93 | 79.13 | 762.658 |
| 4 | 5.065 | 48 | 87.392 | 77.673 | 762.246 |
| 5 | 5.07 | 100 | 88.353 | 76.717 | 762.007 |
| 6 | 5.075 | 100 | 89.111 | 75.964 | 761.835 |
| 7 | 5.08 | 100 | 89.781 | 75.299 | 761.697 |
| 8 | 5.085 | 41 | 90.37 | 74.715 | 761.583 |
| 9 | 5.090 | 100 | 90.406 | 74.679 | 761.491 |
| 10 | 5.095 | 41 | 90.921 | 74.17 | 761.411 |
| 11 | 5.10 | 100 | 91.869 | 73.231 | 761.342 |
| 12 | 5.11 | 100 | 92.722 | 72.388 | 761.230 |
| 13 | 5.12 | 100 | 93.489 | 71.631 | 761.147 |
| 14 | 5.13 | 79 | 94.206 | 70.924 | 761.082 |
| 15 | 5.14 | 63 | 94.87 | 70.269 | 761.035 |
| 16 | 5.15 | 100 | 94.498 | 69.652 | 761.001 |
| 17 | 5.16 | 39 | 96.09 | 69.07 | 760.978 |
| 18 | 5.17 | 100 | 96.654 | 68.516 | 760.965 |
| 19 | **5.18** | 100 | 97.195 | 67.985 | **760.960** |

From the above table it is observed that:

Minimum loss is $F_{Lmin}$ = 5.05 MW

Minimum cost is $F_{Cmin}$ = 760.960 $/hr.

Value of loss at minimum cost is $F_{LatFcmin}$ = 5.18 MW

Value of cost at minimum loss is $F_{CatFLmin}$ = 763.158 $/hr.

**TABLE 6.12**
**Minimum Relative Attainments of 5-Bus System in 2-D Space**

| **S.No** | **$F_C$ ($/hr.)** | **$\tau_C$** | **$F_L$ (MW)** | **$\tau_L$** | **$\sum \tau$** |
|---|---|---|---|---|---|
| 1 | 763.158 | 0.0000 | 5.05 | 1.0000 | 1 |
| 2 | 763.157 | 0.0004 | 5.055 | 0.9615 | 0.09619 |
| 3 | 762.658 | 0.2274 | 5.06 | 0.9230 | 1.1504 |
| 4 | 762.246 | 0.4149 | 5.065 | 0.8846 | 1.2995 |
| 5 | 762.007 | 0.5236 | 5.07 | 0.8461 | 1.3697 |
| 6 | 761.835 | 0.6019 | 5.075 | 0.8076 | 1.4095 |
| 7 | 761.697 | 0.6646 | 5.08 | 0.7692 | 1.4338 |
| 8 | 761.583 | 0.7165 | 5.085 | 0.7307 | 1.4472 |
| **9** | **761.491** | **0.7584** | **5.090** | **0.6923** | **1.4507** |
| 10 | 761.411 | 0.7948 | 5.095 | 0.6538 | 1.4486 |
| 11 | 761.342 | 0.8262 | 5.10 | 0.6153 | 1.4415 |
| 12 | 761.230 | 0.8771 | 5.11 | 0.5384 | 1.4155 |
| 13 | 761.147 | 0.9149 | 5.12 | 0.4615 | 1.3764 |
| 14 | 761.082 | 0.9444 | 5.13 | 0.3846 | 1.3290 |
| 15 | 761.035 | 0.9658 | 5.14 | 0.3076 | 1.2734 |
| 16 | 761.001 | 0.9813 | 5.15 | 0.2307 | 1.2121 |
| 17 | 760.978 | 0.9918 | 5.16 | 0.1538 | 1.1456 |
| 18 | 760.965 | 0.9977 | 5.17 | 0.0769 | 1.0746 |
| 19 | 760.960 | 1.0000 | 5.18 | 0.000 | 1 |

The relative attainments of the objectives are added and the result is shown in the last column of Table 6.12. The Target Point is one for which the sum of relative attainments is maximum. It is seen that **Target Point** for 5-Bus system in 2-D space is:

$F_C$ = 761.491 $/hr.        $F_L$ = 5.09 MW

It is shown at S.No. 9 of the Table 6.12. The noninferior set has also been shown in Fig 6.21.

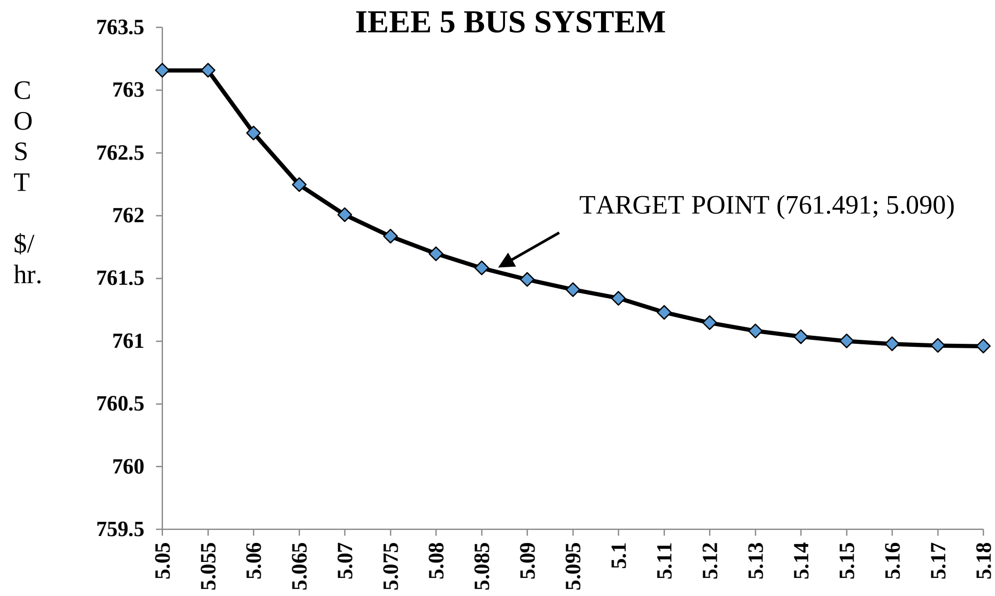



**Fig 6.21** Noninferior set of IEEE 5 bus system

## TABLE 6.13
## Noninferior Set of IEEE 14 Bus System in 2-D Space

| S. No. | $F_L$ (MW) (FIXED) | k | $P_1$ (MW) | $P_2$ (MW) | $P_6$ (MW) | $F_C$ ($/hr.) |
|---|---|---|---|---|---|---|
| 1 | 7.00 | 100 | 85.09 | 103.58 | 77.54 | 1184.64 |
| 2 | 7.10 | 100 | 85.12 | 103.56 | 77.54 | 1183.63 |
| 3 | 7.20 | 39 | 91.03 | 98.86 | 76.35 | 1176.50 |
| 4 | 7.30 | 26 | 99.67 | 88.44 | 78.17 | 1168.11 |
| 5 | 7.40 | 63 | 109.04 | 88.44 | 78.17 | 1157.98 |
| 6 | 7.50 | 65 | 114.42 | 87.89 | 64.18 | 1153.20 |
| 7 | 7.60 | 38 | 117.19 | 90.71 | 58.69 | 1150.89 |
| 8 | 7.70 | 74 | 119.12 | 96.08 | 51.53 | 1149.98 |
| 9 | 7.80 | 63 | 120.00 | 96.79 | 50.00 | 1141.63 |
| 10 | 7.90 | 62 | 120.00 | 96.80 | 50.00 | 1140.61 |
| 11 | 8.00 | 57 | 136.93 | 76.78 | 55.56 | 1140.14 |
| 12 | 8.10 | 79 | 142.11 | 64.26 | 55.70 | 1139.00 |
| 13 | 8.20 | 84 | 145.41 | 63.20 | 58.40 | 1138.62 |
| 14 | 8.30 | 36 | 145.80 | 70.46 | 51.00 | 1137.32 |
| 15 | 8.40 | 100 | 148.70 | 68.63 | 50.00 | 1136.75 |
| 16 | 8.50 | 100 | 150.00 | 67.96 | 50.00 | 1136.40 |
| 17 | 8.60 | 100 | 150.00 | 67.47 | 50.00 | 1136.10 |

**TABLE 6.14**

**Minimum Relative Attainments of 14-Bus System in 2-D Space**

| S.No | $F_C$ ($/hr) | $\tau_C$ | $F_L$ (MW) | $\tau_L$ | $\sum_\tau$ |
|---|---|---|---|---|---|
| 1 | 1184.64 | 0.0000 | 7.00 | 1.0000 | 1 |
| 2 | 1183.63 | 0.0207 | 7.10 | 0.9375 | 0.9582 |
| 3 | 1176.50 | 0.1675 | 7.20 | 0.8750 | 1.0425 |
| 4 | 1168.11 | 0.3405 | 7.30 | 0.8125 | 1.1530 |
| 5 | 1157.98 | 0.5492 | 7.40 | 0.7500 | 1.2990 |
| 6 | 1153.20 | 0.6476 | 7.50 | 0.6875 | 1.3350 |
| **7** | **1150.89** | **0.6953** | **7.60** | **0.6250** | **1.3203** |
| 8 | 1149.98 | 0.7140 | 7.70 | 0.5625 | 1.2765 |
| 9 | 1141.63 | 0.7212 | 7.80 | 0.5000 | 1.2212 |
| 10 | 1140.61 | 0.7216 | 7.90 | 0.4375 | 1.1590 |
| 11 | 1140.14 | 0.9166 | 8.00 | 0.375 | 1.2910 |
| 12 | 1139.00 | 0.9408 | 8.10 | 0.3125 | 1.2530 |
| 13 | 1138.62 | 0.9480 | 8.20 | 0.2500 | 1.1980 |
| 14 | 1137.32 | 0.9748 | 8.30 | 0.1875 | 1.1620 |
| 15 | 1136.75 | 0.9866 | 8.40 | 0.1250 | 1.1110 |
| 16 | 1136.40 | 0.9938 | 8.50 | 0.0625 | 1.0560 |
| 17 | 1136.10 | 1.0000 | 8.60 | 0.0000 | 1 |

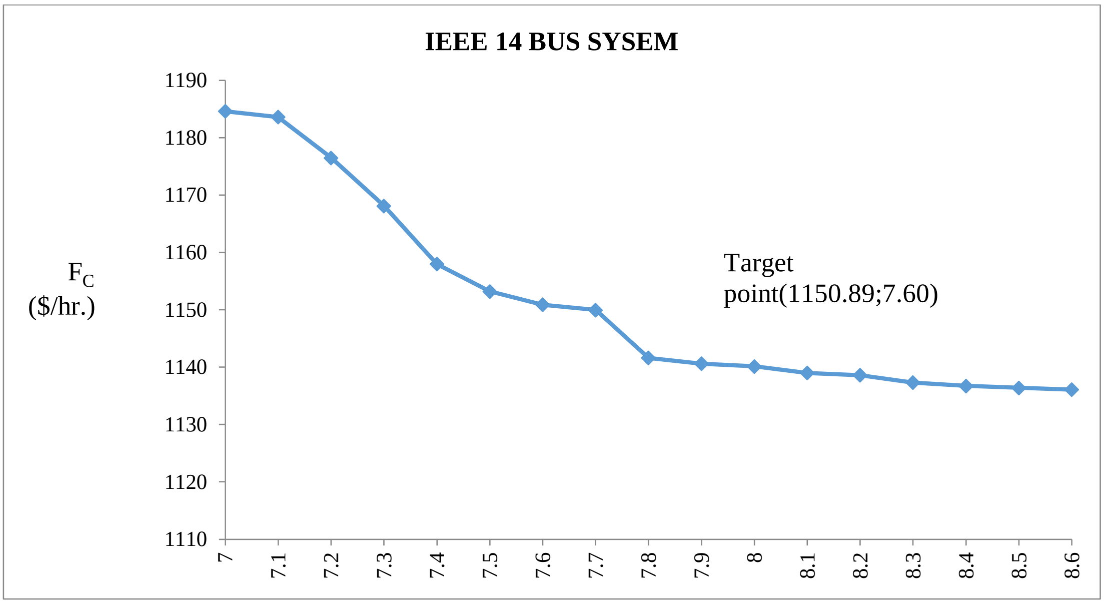
From the above Table it is observed that:

Minimum loss is $F_{Lmin}$ = 7.00 MW Minimum cost is $F_{Cmin}$ =1136.10 $/hr

Value of loss at minimum cost is $F_{LatFcmin}$ =8.60 MW

Value of cost at minimum loss is $F_{CatFLmin}$ =1184.64 $/hr.

A noninferior set of IEEE 14 bus system is also shown in Fig.6.22.




$F_L$ (MW)

**Fig 6.22** Noninferior set of IEEE 14 bus system

**TABLE 6.15**
**Noninferior Set of IEEE 30 Bus System in 2-D Space**

| S. No. | $F_L$ (MW) (FIXED) | k | $P_1$ (MW) | $P_2$ (MW) | $P_6$ (MW) | $F_C$ ($/hr.) |
|---|---|---|---|---|---|---|
| 1 | 6.90 | 64 | 50.36 | 120 | 120 | 1358.07 |
| 2 | 7.00 | 100 | 52.18 | 118.21 | 120 | 1355.07 |
| 3 | 7.10 | 100 | 57.70 | 112.79 | 120 | 1346.35 |
| 4 | 7.20 | 67 | 62.77 | 111.03 | 117.49 | 1338.74 |
| 5 | 7.30 | 68 | 67.01 | 103.68 | 119.99 | 1333.12 |
| 6 | 7.40 | 78 | 68.62 | 111.94 | 110.23 | 1326.81 |
| 7 | 7.50 | 70 | 73.43 | 105.36 | 112.10 | 1321.11 |
| 8 | 7.60 | 44 | 74.14 | 112.09 | 104.75 | 1317.80 |
| 9 | 7.70 | 75 | 79.49 | 103.51 | 108.08 | 1312.15 |
| 10 | 7.80 | 65 | 82.18 | 102.99 | 106.02 | 1308.30 |
| 11 | 7.90 | 51 | 82.97 | 107.64 | 100.68 | 1305.82 |
| 12 | 8.10 | 36 | 93.80 | 84.48 | 113.16 | 1300.44 |
| 13 | 8.20 | 76 | 92.00 | 100.31 | 99.27 | 1295.63 |
| 14 | 8.30 | 43 | 97.42 | 89.37 | 104.90 | 1292.88 |
| 15 | 8.40 | 39 | 97.57 | 95.73 | 98.49 | 1290.15 |
| 16 | 8.50 | 42 | 97.84 | 100.30 | 93.75 | 1288.60 |
| 17 | 8.60 | 55 | 100.10 | 99.13 | 92.76 | 1286.33 |
| 18 | 8.70 | 24 | 101.64 | 99.70 | 9075 | 1284.55 |
| 19 | 8.80 | 46 | 103.37 | 99.63 | 89.18 | 1282.77 |
| 20 | 8.90 | 52 | 109.75 | 86.68 | 95.86 | 1279.81 |
| 21 | 9.00 | 43 | 112.73 | 82.60 | 96.99 | 1278.43 |
| 22 | 9.10 | 34 | 110.81 | 93.18 | 88.50 | 1276.81 |
| 23 | 9.20 | 49 | 114.11 | 88.57 | 89.90 | 1274.98 |
| 24 | 9.30 | 43 | 114.25 | 88.57 | 89.90 | 1273.94 |
| 25 | 9.40 | 100 | 120.00 | 80.61 | 92.18 | 1272.54 |
| 26 | 9.50 | 42 | 116.44 | 94.37 | 82.07 | 1271.84 |
| 27 | 9.60 | 15 | 118.60 | 92.63 | 81.756 | 1270.44 |
| 28 | 9.70 | 100 | 120.00 | 92.54 | 80.522 | 1269.43 |
| 29 | 9.80 | 100 | 120.00 | 95.99 | 77.21 | 1269.14 |
| 30 | 9.90 | 100 | 120.00 | 99.04 | 74.25 | 1269.11 |

From the above Table it is observed that:

Minimum loss is $F_{Lmin}$ = 6.90 MW

Minimum cost is $F_{Cmin}$ =1269.11 $/hr.

Value of loss at minimum cost is $F_{LatFcmin}$ =9.90 MW

Value of cost at minimum loss is $F_{CatFLmin}$ =1358.07 $/hr.

**TABLE 6.16**

**Minimum Relative Attainments of 30-Bus System in 2-D Space**

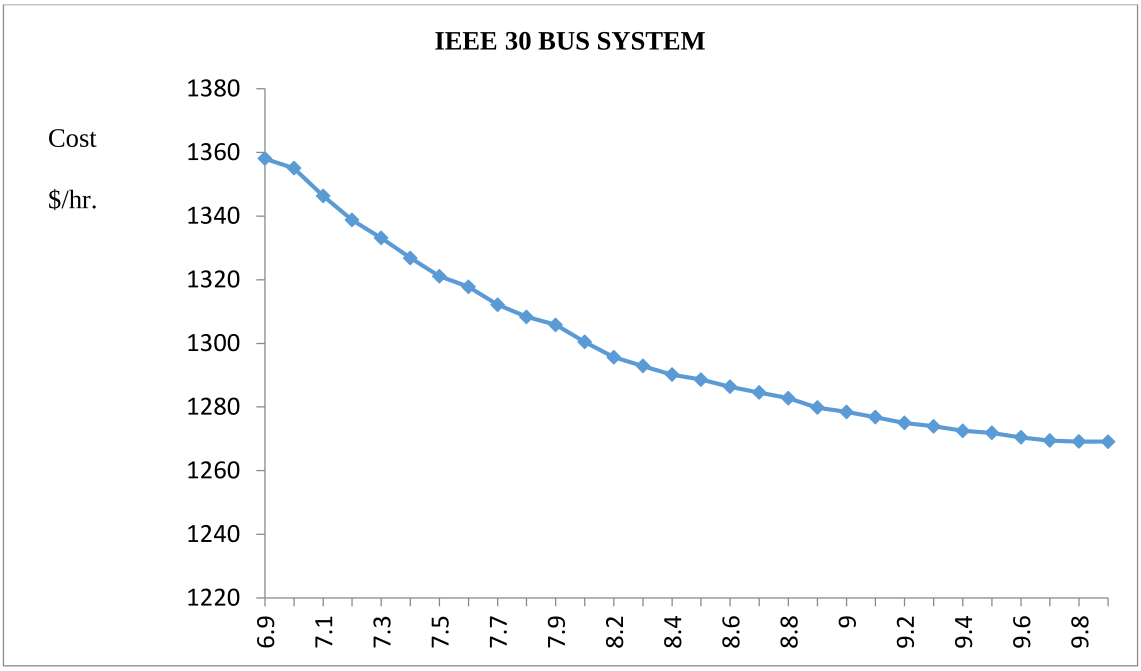
| S.No | $F_C$ ($/hr) | $\tau_C$ | $F_L$ (MW) | $\tau_L$ | $\sum\tau$ |
|---|---|---|---|---|---|
| 1 | 1358.07 | 0.0000 | 6.90 | 1.0000 | 1 |
| 2 | 1355.07 | 0.0337 | 7.00 | 0.9667 | 1.0040 |
| 3 | 1346.35 | 0.1317 | 7.10 | 0.9333 | 1.0656 |
| 4 | 1338.74 | 0.2167 | 7.20 | 0.9000 | 1.1167 |
| 5 | 1333.12 | 0.2805 | 7.30 | 0.8667 | 1.1472 |
| 6 | 1326.81 | 0.3514 | 7.40 | 0.8333 | 1.1847 |
| 7 | 1321.11 | 0.4155 | 7.50 | 0.8000 | 1.2155 |
| 8 | 1317.80 | 0.4527 | 7.60 | 0.7667 | 1.2194 |
| 9 | 1312.15 | 0.5161 | 7.70 | 0.7333 | 1.2495 |
| 10 | 1308.30 | 0.5595 | 7.80 | 0.7000 | 1,2595 |
| 11 | 1305.82 | 0.5873 | 7.90 | 0.6667 | 1,2540 |
| 12 | 1300.44 | 0.6478 | 8.10 | 0.6000 | 1.2478 |
| **13** | **1295.63** | **0.7019** | **8.20** | **0.5667** | **1.2686** |
| 14 | 1292.88 | 0.7328 | 8.30 | 0.5333 | 1.2661 |
| 15 | 1290.15 | 0.7635 | 8.40 | 0.5000 | 1.2635 |
| 16 | 1288.60 | 0.7809 | 8.50 | 0.4667 | 1.2476 |
| 17 | 1286.33 | 0.8064 | 8.60 | 0.4333 | 1.2397 |
| 18 | 1284.55 | 0.8264 | 8.70 | 0.4000 | 1.2264 |
| 19 | 1282.77 | 0.8464 | 8.80 | 0.3667 | 1.2131 |
| 20 | 1279.81 | 0.8797 | 8.90 | 0.3333 | 1.2130 |
| 21 | 1278.43 | 0.8952 | 9.00 | 0.3000 | 1.1952 |
| 22 | 1276.81 | 0.9134 | 9.10 | 0.2667 | 1.1851 |
| 23 | 1274.98 | 0.9340 | 9.20 | 0.2333 | 1.1673 |
| 24 | 1273.94 | 0.9457 | 9.30 | 0.2000 | 1.1457 |
| 25 | 1272.54 | 0.9614 | 9.40 | 0.1667 | 1.1281 |
| 26 | 1271.84 | 0.9693 | 9.50 | 0.1333 | 1.1026 |
| 27 | 1270.44 | 0.9850 | 9.60 | 0.1000 | 1.0856 |
| 28 | 1269.43 | 0.9964 | 9.70 | 0.0667 | 1.0631 |
| 29 | 1269.14 | 0.9997 | 9.80 | 0.0333 | 1.0330 |
| 30 | 1269.11 | 1.0000 | 9.90 | 0.0000 | 1 |

The relative attainments of the objectives are added and the result is shown in the last column of Table 6.16. The Target Point is one for which the sum is maximum. It is seen Target Point achieved for 30-Bus system in 2-D space is:

$F_C$ = 1295.63 $/hr. $F_L$ = 8.20 MW

It is shown at S.No.13 of the Table 6.16.

The noninferior set has also been shown in Fig 6.23.




**$F_L$ (MW)**

**Fig 6.23** Noninferior set of IEEE 30 bus system

Table 6.17 shows the Target Point (T.P.). Here Maximization of minimum relative attainment has been used for identifying the Target point ( **$F_C$*, $F_L$***) in 2-D space. Genetic algorithm has been used to develop noninferior set. The formulation of MELD problem has been done by constraint method. This has been done for IEEE 5,14 and 30 bus systems.

**TABLE 6.17**
**Target point using GA technique**

| **S. No.** | **IEEE Bus System** | **Target Point** | |
|---|---|---|---|
| | | **$F_C$* ($/hr.)** | **$F_L$* (MW)** |
| **1** | IEEE 5 Bus | 761.491 | 5.09 |
| **2** | IEEE 14 Bus | 1153.20 | 7.50 |
| **3** | IEEE 30 Bus | 1295.63 | 8.2 |

Table 6.18 shows the comparison of Target point achieved by Equal percentage saving based [160] method as shown in columns (3) and (4) and Maximization of minimum relative attainment as shown in columns (5) and (6) respectively. Problem was formulated by Weighting method and Constraint method respectively. Noninferior set were generated by PSO [160] and Genetic Algorithm in 2-D space.

**TABLE 6.18**
**Comparison of Target Point**

| S. No. | IEEE Bus System | Target point using equal percentage saving concept | | Target point using maximization of minimum relative attainment | |
|---|---|---|---|---|---|
| | | $F_C$* ($/hr.) | $F_L$* (MW) | $F_C$* ($/hr.) | $F_L$* (MW) |
| (1) | (2) | (3) | (4) | (5) | (6) |
| 1 | IEEE 5 Bus | 761.67 | 5.1379 | 761.491 | 5.09 |
| 2 | IEEE 14 Bus | 1183.28 | 6.7582 | 1153.20 | 7.50 |
| 3 | IEEE 30 Bus | 1318.05 | 7.5652 | 1295.63 | 8.2 |

It is observed from Table 6.18, the Target point identified by maximization of minimum relative attainment is better than that obtained by equal percentage saving method.

## 6.9 CONCLUSIONS

### 6.9.1 Weighting Method

Multiobjective economic load dispatch problem has been formulated as the weighted sum of cost of generation, system transmission losses and environmental emissions for IEEE 5,14 and 30 bus systems. The noninferior set in 3-D space is generated by systematically varying the weights attached to the objectives and solving the MELD problem by Particle Swarm Optimization (PSO) technique. Target Point is achieved by maximization of minimum relative attainment method and Fuzzy Logic system for IEEE 5,14 and 30 bus systems. Noninferior set has also been developed in 2-D space considering all combinations of two objectives. This explicitly shows the behavior of objectives i.e. cost of generation is conflicting with system transmission losses and environmental emissions whereas transmission losses and environmental emissions are supportive. But this behavior may not be true for all domains. Weighted sum method has no limitation in handling more than three objectives.

### 6.9.2 Constraint Method

MELD problem has been formulated by constraint method considering cost of generation and system transmission losses for IEEE 5,14 and 30 bus systems. Noninferior set has been generated by Genetic Algorithm. Target point has been achieved by MMR method.

Weighted sum method and constraint method are generating techniques which emphasize on the development of information about a problem. The computational effort in weighting method is more than that in constraint method. Weighting method may give poor coverage of noninferior set, if the weights are not selected properly but constraint method give good coverage of the noninferior set.

Target point achieved by PSO technique in 2-D space and 3-D space are found to be better than those achieved by GA for IEEE 5,14 and 30 bus systems.

## Research Publications

1. N.K. Jain, Uma Nangia, Jyoti Jain, "GA Based Multiobjective Economic Load Dispatch by Maximization of Minimum Relative Attainments", 2012 IEEE Fifth International Conference on Power Electronics (ICPEICES 2012), Dec. 6-8, 2012, Delhi Technological University, Delhi.
2. N. K. Jain, Uma Nangia, Jyoti Jain, "Multiobjective economic load dispatch studies in 2-D and 3-D space by particle swarm optimization technique", Journal of Institution of Engineers India: Series B, Springer, 100(3), pp. 237-247, (2019).
3. N. K. Jain, Uma Nangia, Jyoti Jain, "Multiobjective economic load dispatch using fuzzy logic decision system" IEEE second International Conference on Power Electronics, Intelligent Control and Energy, (ICPEICES-2018) Delhi, 22-24 October, 2018.

# CHAPTER 7
# MELD USING FEASIBILITY ORIENTED PARTICLE SWARM OPTIMIZATION ALGORITHM

## 7.1 INTRODUCTION

In this Chapter, Feasibility Oriented Particle Swarm Optimization (FOPSO) Algorithm has been developed which determines uniformly distributed Pareto-Optimal Front in a partial run for Multiobjective Economic Load Dispatch (MELD) problem in 2-D space and 3-D space. In 2-D space, two objective of Power Systems - cost of generation and systems transmission losses have been considered whereas in 3-D space, three objectives of Power Systems - cost of generation, system transmission losses and environmental emission have been considered.

The Pareto-Optimal Front for Multiobjective optimization problem is usually obtained by optimizing the Multiobjective function at least as many times as the number of points required on the Pareto-Front [160]. But this algorithm produces the Pareto-optimal Front in a partial run by employing two phases of selection mechanism and thus overcomes the problem of multiple runs required to produce the well distributed Pareto-Optimal Front.

In the *first phase*, it starts with a large population of points and identifies the feasible points which satisfy the constraints of the MELD problem after each iteration. These points are copied to the Qualifying set and are not allowed to participate in future iterations of PSO. This phase is complete after sufficient number of points have been copied to the Qualifying set and the algorithm switches over to second phase. In the second phase, a well distributed Pareto-Optimal front is selected from the Qualifying set based on the minimum distance from equidistant points between minimum and maximum cost of generation along cost of generation ($F_C$) axis and between minimum and maximum system transmission losses along ($F_L$) axis.

The ideal situation where one would like to operate the power system is one where all the objectives-cost of generation ($F_C$), system transmission losses ($F_L$) and Environmental emission ($F_E$) attain their minimum value simultaneously. This point has been defined as the Ideal Point (I.P.) and is represented by ($F_{Cmin}$,$F_{Lmin}$) in 2-D space. Ideal Point is represented by ($F_{Cmin}$, $F_{Lmin}$, $F_{Emin}$) in 3-D space. It is of utmost importance to recognize that this is an infeasible point. The solution of Multiobjective problem gives a number of solutions, all of which are non-dominated solutions. The power system operator chooses the best compromise solution ($F_C$*, $F_L$*) in 2-D space and ($F_C$*, $F_L$*, $F_E$*) in 3-D space, out of these solutions as per his requirements. In this chapter, the best compromise solution is obtained. It lies at minimum distance from I.P. It has been defined as target point (T.P.).

## 7.2 PROBLEM STATEMENT

Two objectives of Multiobjective Economic Load Dispatch considered in 2-D space are:

i. To minimize the cost of generation

ii. To minimize the system transmission losses

The Multiobjective function F in 2-D space is

Minimize

$$F = (F_C, F_L) \tag{7.1}$$

Three objectives of Multiobjective Economic Load Dispatch considered in 3-D space are:

i. To minimize the cost of generation

ii. To minimize the system transmission losses

iii. To minimize the environmental emission

The Multiobjective function F in 3-D space is

Minimize

$$F = (F_C, F_L, F_E) \tag{7.2}$$

Where

$$F_C = \sum_{i=1}^{NG} a_i P_{gi}^2 + b_i P_{gi} + c_i \tag{7.3}$$

$$F_L = \sum_{i=1}^{NG} \sum_{j=1}^{NG} P_{gi} B_{ij} P_{gj} \tag{7.4}$$

$$F_E = \sum_{i=1}^{NG} d_i P_{gi}^2 + e_i P_{gi} + f_i \tag{7.5}$$

subject to

Inequality constraint

$$P_{gimin} \leq P_{gi} \leq P_{gimax} \quad i = 1, 2 \ldots NG \tag{7.6}$$

and Equality constraint

$$f = \sum_{i=1}^{NG} P_{gi} - P_D - F_L = 0 \tag{7.7}$$

## 7.3 FEASIBILITY ORIENTED PARTICLE SWARM OPTIMIZATION (FOPSO) ALGORITHM

The proposed algorithm **Feasibility Oriented Particle Swarm Optimization (FOPSO)** algorithm determines the well distributed Pareto-Front in partial run by employing two phases of selection mechanism. The algorithm is explained in detail for MELD problem in 2-D space.

### *First Phase*

In the first phase, the algorithm identifies sufficient number of points lying in the feasible region from a randomly generated large population of points as shown in Fig.7.1 for 2-D space. This is accomplished by identifying the points which satisfy the constraints of the problem after each iteration of PSO. These points are copied to the Qualifying Set and are not allowed to participate in future iterations of PSO. If this is not done, the experience has been that these particles play their own role in the movement of other particles. This results in collapse of all particles to the then global best. It defeats the purpose of getting various solutions for Multiobjective problem. The process of generating Qualifying set stops after required number of such points are identified which means that there is no need to run the PSO programme completely with this strategy. For this phase the function to be minimized contains only the equality constraints defined as

$$f = \sum_{i=1}^{NG} P_{gi} - P_D - F_L \leq 10^{-6} \quad (7.8)$$

Where Ɛ is a small number

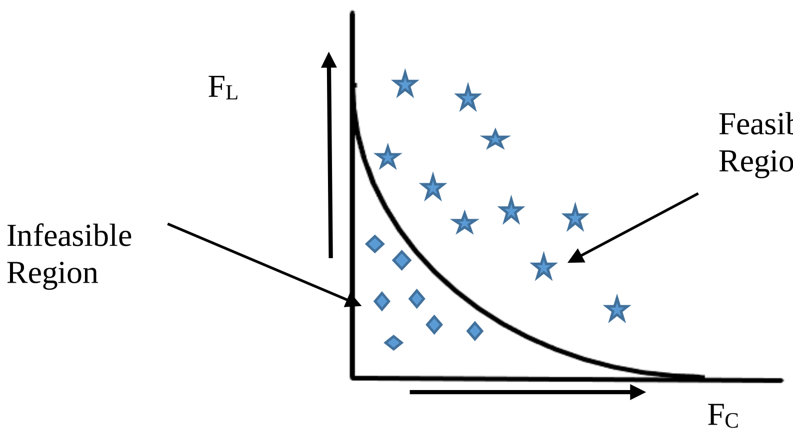



**Fig. 7.1** Qualifying Set in 2- D space

### *Second Phase*

In the second phase, the algorithm identifies the well – distributed Pareto Front. This is the boundary of feasible region which means the points for which constraints are just satisfied. The following strategy has been adopted:

From the Qualifying Set I.P., $F_{Cmax}$ and $F_{Lmax}$ are identified. Both $F_C$ and $F_L$ axis are divided into ten equidistant points between $F_{Cmin}$ and $F_{Cmax}$ along $F_C$ axis and between $F_{Lmin}$ and $F_{Lmax}$ along $F_L$ axis. The Pareto-Optimal Front is selected by identifying the points which lie at minimum distance from these equidistant points along each axis as shown in Fig. 7.2.

FOPSO algorithm for MELD problem considering three objectives: $F_C, F_L$ and $F_E$ in 3-D space is exactly the same as that considering two objectives $F_C$ and $F_L$ in 2-D space. The

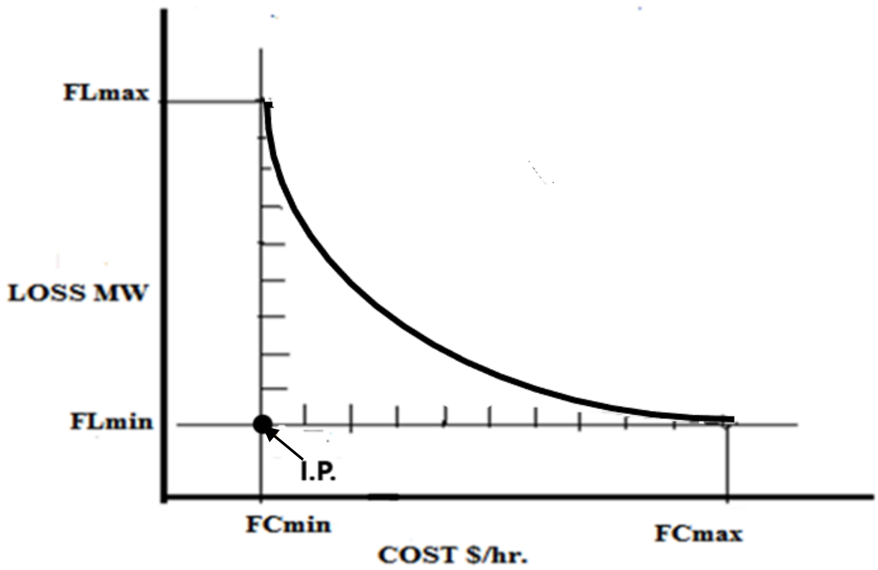



**Fig. 7.2** Pareto-Optimal Front in 2-D space

Qualifying set for First phase in 3-D space is obtained by identifying the points which satisfy the constraints of the problem. In the second phase, twenty equidistant points between $F_{Emin}$ and $F_{Emax}$ along $F_E$ axis are also identified in addition to that along $F_C$ and $F_L$ axis. The Pareto – Optimal Front is selected by identifying points which lie at minimum distance from these equidistant points along each axis. Thus Pareto Front in 3-D space is obtained. The Target Point (T.P.) is identified as the one which lies at minimum distance from the IP.

## 7.4 COMPUTATIONAL PROCEDURE

The steps for the Feasibility Oriented Particle Swarm Optimization (FOPSO) algorithm for First Phase are:

1. Initialize Parameters of PSO - W, $C_p, C_g, r_p, r_g$, ITmax, $\varepsilon$, q , N ( number of particles for the Pareto-Optimal front), k, Kount . Parameters setting for PSO algorithm are

   W= 0.6; $C_p, C_g, r_p, r_g$ = 1; ITmax = 3000; $\varepsilon$ =1*10$^{-6}$.

   For IEEE 5, 14 and 30 bus system initial populations of 400, 600 and 600 are chosen respectively. The number of points to be copied to the Qualifying set are chosen to be 300, 200 and 400 for IEEE 5, 14 and 30 bus systems respectively.
2. Initialize k = 0.
3. Initialize cost, losses and emission characteristics of generators.
4. Generate particles between $P_{gimin}$ and $P_{gimax}$.
5. Generate random velocity for all particles of the initial population.

6. Calculate function (fitness) value corresponding to these particles and increment Kount by q.
7. Determine Pbest and gbest.
8. Determine particle number corresponding to gbest.
9. For all particles check if function value $f \leq \varepsilon$ , if yes store, particle number and its coordinates in the Qualifying set and remove from main population.
10. Increase the iteration by one. $k = k + 1$.
11. Calculate velocity of remaining particles for next iteration.
12. Check velocity limits. Fix the velocity to the limit violated.
13. Update position of all particles using this velocity.
14. Check generator constraints.
15. Calculate function f, defined by equation (7.7).
16. Increment function evaluations i.e. Kount by q.
17. Update pbest values and gbest values.
18. For all particles check if function value $\leq \varepsilon$. For a particular particle, if yes, store particle number and its coordinates in the Qualifying set. Remove from main population.
19. Check if the number of particles in the Qualifying set exceeds the specified value. If yes, go to 20. Else go to 10.
20. Store the particles in Qualifying set to be used for Second Phase.

The steps for Second Phase using the Feasibility Oriented Particle Swarm Optimization (FOPSO) algorithm are:

1. Identify the ideal point for 2-D space represented by ($F_{Cmin}$, $F_{Lmin}$), and for 3-D space identified by ($F_{Cmin}$, $F_{Lmin}$, $F_{Emin}$) from the Qualifying set. Also, identify $F_{Cmax}$, $F_{Lmax}$ and $F_{Emax}$ from the Qualifying set.
2. Determine the ranges of objective functions i.e. $F_{Crange}$, $F_{Lrange}$ , $F_{Erange}$

   range of cost of generation is $F_{Crange} = F_{Cmax} - F_{Cmin}$

   range of system transmission losses is $F_{Lrange} = F_{Lmax} - F_{Lmin}$

   and range of environmental emission is $F_{Erange} = F_{Emax} - F_{Emin}$
3. Divide the $F_C$ axis from $F_{Cmin}$ to $F_{Cmax}$ in ten and twenty equidistant points for 2-D space and 3-D space respectively.
4. Similarly, identify ten equidistant points between $F_{Lmin}$ and $F_{Lmax}$ along $F_L$ axis in 2-D space and twenty equidistance point along $F_C$ , $F_L$ and $F_E$ axis in 3-D space.
5. Using distance functions, identify the points of Qualifying set which lie at minimum distance from these equidistant points.
6. This gives us the well distributed Pareto - Optimal Front. Finally, the best compromise solution (Target Point) is located by determining the point which lies at minimum distance from the I.P. The distance function in 2-D space and 3-D space are:

$$\text{Distance} = [(F_C - F_{Cmin})^2 + (F_L - F_{Lmin})^2 ]^2 \text{ for 2-D space} \tag{7.9}$$

$$\text{Distance} = [(F_C - F_{Cmin})^2 + (F_L - F_{Lmin})^2 + (F_E - F_{Emin})^2 ]^{1/2} \text{ for 3-D space} \tag{7.10}$$

Flow chart of Feasibility Oriented Particle Swarm Optimization (FOPSO) Algorithm for First ***phase*** and ***Second Phase*** are shown in Fig.7.3. and 7.4 respectively.

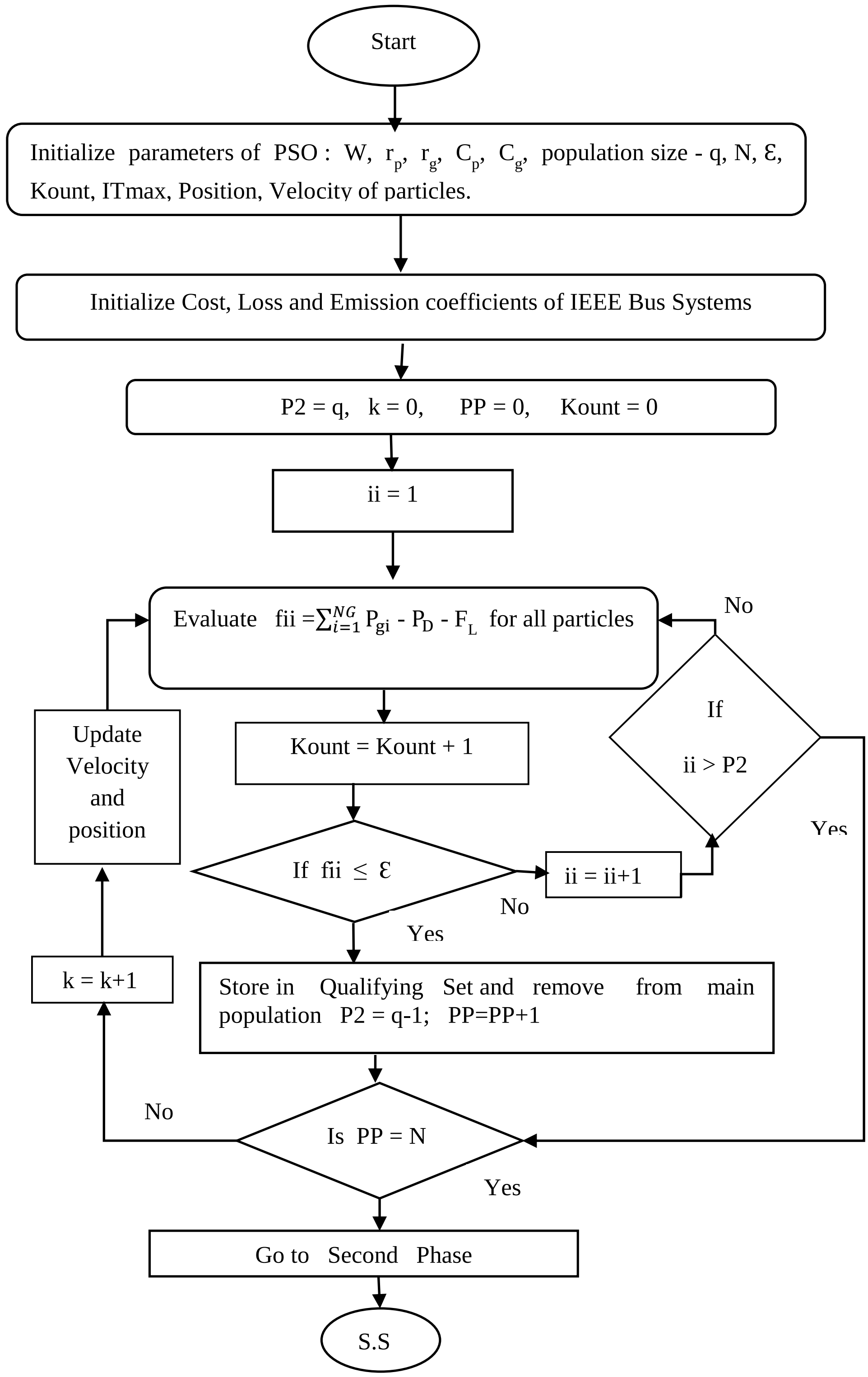



**Fig.7.3** Flow chart of Feasibility Oriented Particle Swarm Optimization (FOPSO) Algorithm for *First phase*

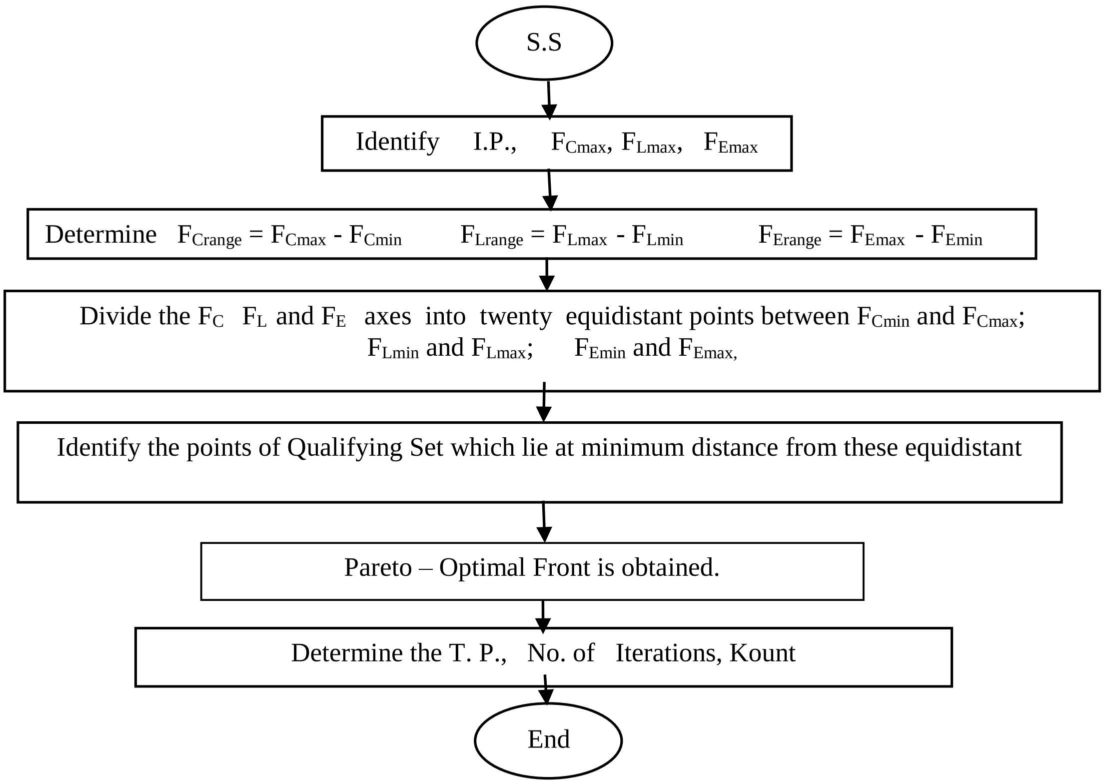



**Fig.7.4** Flow chart of Feasibility Oriented Particle Swarm Optimization (FOPSO) Algorithm for *Second phase*

## 7.5 COMPUTATIONAL RESULTS AND DISCUSSION

In the first phase, the algorithm identifies sufficient number of points lying in the feasible region. The feasible points obtained from the First phase are stored in the Qualifying set. From these points, $F_{Cmin}$, $F_{Cmax}$, $F_{Lmin}$, $F_{Lmax}$ are identified for 2-D space and $F_{Cmin}$, $F_{Cmax}$, $F_{Lmin}$, $F_{Lmax}$, $F_{Emin}$ and $F_{Emax}$ are identified for 3-D space. These are used to calculate the range of cost of generation ($F_{Crange}$), range of transmission losses ($F_{Lrange}$), and range of Emission ($F_{Erange}$).Target point is also identified for 2-D space and 3-D space.

### 7.5.1 Results of MELD in 2-D Space

Table 7.1 shows the range of $F_C$ and $F_L$ for IEEE 5, 14 and 30 bus systems in 2-D space. From the results of Table 7.1, Ideal points I.P. ($F_{Cmin}$, $F_{Lmin}$) can be identified for IEEE 5, 14 and 30 bus systems. Columns (4) and (7) show the value of $F_{Cmin}$ and $F_{Lmin}$..

**TABLE 7.1**
**Range of $F_C$ and $F_L$**

| S. No. (1) | IEEE System (2) | $F_{Cmax}$ ($/hr.) (3) | $F_{Cmin}$ ($/hr.) (4) | $F_{Crange}$ (5) | $F_{Lmax}$ (MW) (6) | $F_{Lmin}$ (MW) (7) | $F_{Lrange}$ (8) |
|---|---|---|---|---|---|---|---|
| 1 | 5 Bus | 762.47 | 760.980 | 1.49 | 5.150 | 5.060 | 0.092 |
| 2 | 14 Bus | 1180.34 | 1148.674 | 31.66 | 7.616 | 7.263 | 0.352 |
| 3 | 30 Bus | 1287.30 | 1252.000 | 34.80 | 9.416 | 8.783 | 0.632 |

Ideal Point (I.P.) for IEEE 5, 14 and 30 bus system are (760.98, 5.06), (1148.67, 7.26), and (1252.00, 8.78) respectively. $F_C$ and $F_L$ axis are then divided into ten equidistant points between $F_{Cmin}$ and $F_{Cmax}$ along $F_C$ axis and between $F_{Lmin}$ and $F_{Lmax}$ along $F_L$ axis for 2-D space. The Pareto optimal front is obtained by identifying the points which lie at minimum distance from these equidistant points along each axis.

### IEEE 5 Bus System

Initially, a population of 400 points is generated randomly by MATLAB Programme. Out of this, 300 points which satisfy the equality constraints of the problem are copied to the Qualifying Set. This is shown in Fig.7.5.

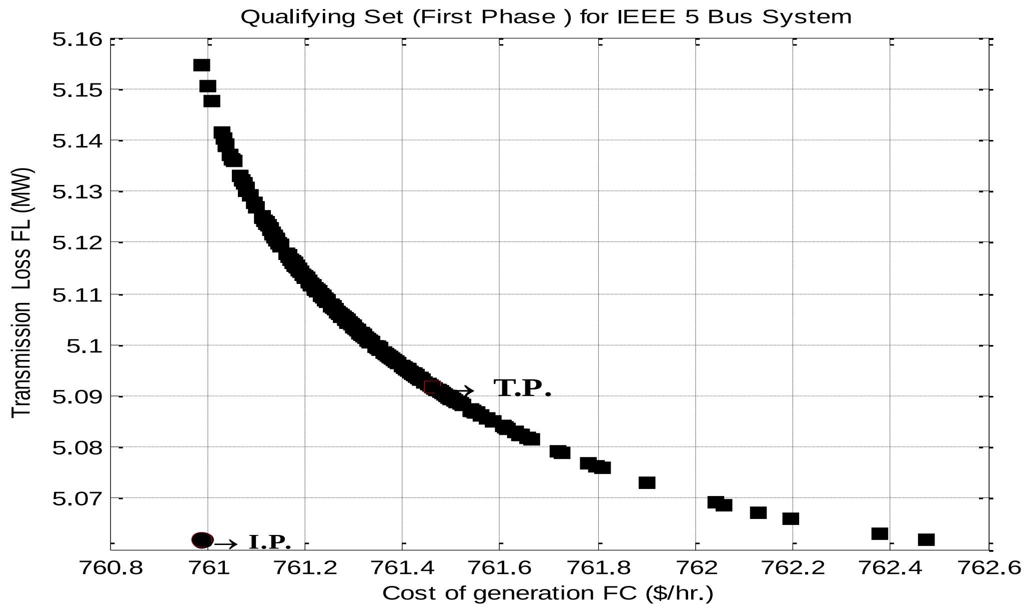



**Fig. 7.5** Qualifying Set (First Phase) for IEEE 5 Bus System

The points of Qualifying set which lie at minimum distance from the equidistant points are shown in Table 7.2 and Fig. 7.6. This forms the Pareto Front of IEEE 5 bus system. First and Fourth Columns of Table 7.2 show Serial Number. Columns (2) & (3) and (5) & (6) show the Pareto Optimal front for Cost of generation and corresponding Losses.

**TABLE 7.2**
**Pareto Optimal Front for IEEE 5 Bus System**

| S. No. (1) | $F_C$ $/hr. (2) | $F_L$ MW (3) | S. No. (4) | $F_C$ $/hr. (5) | $F_L$ MW (6) |
|---|---|---|---|---|---|
| 1 | 761.41 | 5.09 | 11 | 761.52 | 5.09 |
| 2 | 761.36 | 5.10 | 12 | 761.59 | 5.09 |
| 3 | 761.30 | 5.10 | 13 | 761.66 | 5.08 |
| 4 | 761.24 | 5.11 | 14 | 761.73 | 5.08 |
| 5 | 761.19 | 5.12 | 15 | 761.81 | 5.08 |
| 6 | 761.14 | 5.12 | 16 | 761.90 | 5.07 |
| 7 | 761.09 | 5.13 | 17 | 762.06 | 5.07 |
| 8 | 761.05 | 5.14 | 18 | 761.19 | 5.07 |
| 9 | 761.01 | 5.15 | 19 | 761.38 | 5.06 |
| 10 | 760.99 | 5.15 | 20 | 762.47 | 5.06 |

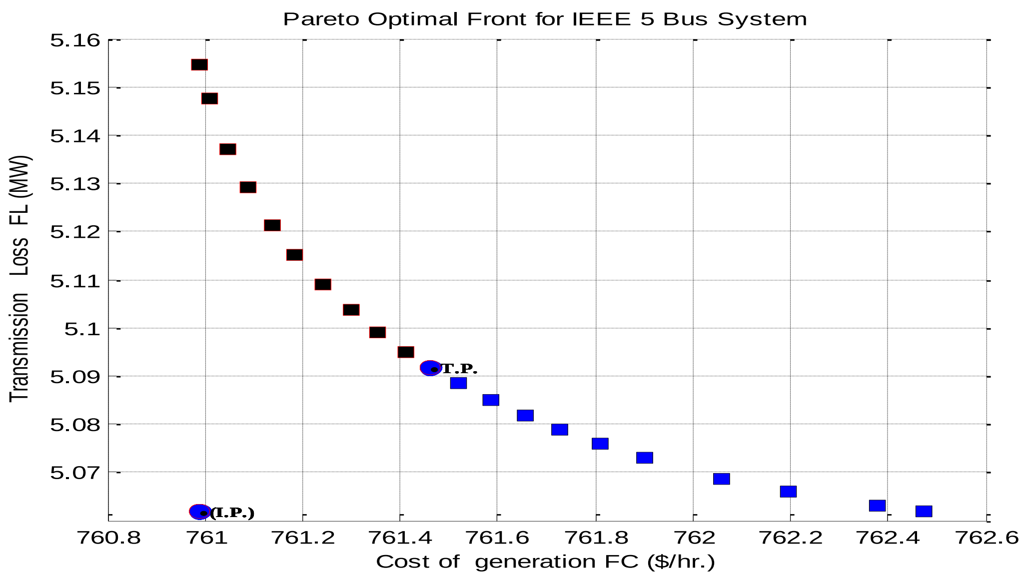



**Fig.7.6** Pareto Optimal Front for IEEE 5 Bus System

## IEEE 14 Bus System

In this case, six hundred (600) points are randomly generated. Out of these, two hundred points which satisfy the equality constraints of the problem are identified and stored in the Qualifying set. Qualifying set for IEEE 14 bus systems is shown in Fig. 7.7.

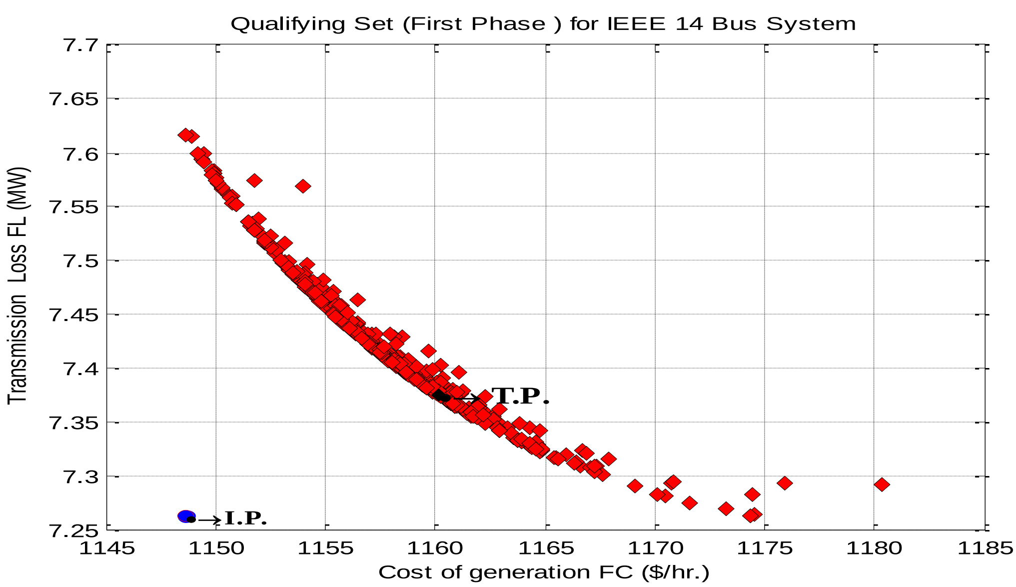



**Fig. 7.7** Qualifying Set (First Phase) for IEEE 14 Bus System

From the points of Qualifying set, Pareto optimal Front is identified which is shown in Table 7.3 and Fig.7.8

**TABLE 7.3**
**Pareto Optimal Front for IEEE 14 bus system**

| S. No. | $F_C$ (\$/hr.) | $F_L$ (MW) | S. No. | $F_C$ (\$/hr.) | $F_L$ (MW) |
|---|---|---|---|---|---|
| 1 | 1159.10 | 7.39 | 11 | 1161.49 | 7.36 |
| 2 | 1157.89 | 7.41 | 12 | 1162.26 | 7.35 |
| 3 | 1156.40 | 7.43 | 13 | 1163.71 | 7.33 |
| 4 | 1155.52 | 7.45 | 14 | 1165.60 | 7.32 |
| 5 | 1154.19 | 7.47 | 15 | 1167.27 | 7.30 |
| 6 | 1153.09 | 7.50 | 16 | 1169.10 | 7.29 |
| 7 | 1151.78 | 7.53 | 17 | 1171.63 | 7.28 |
| 8 | 1150.80 | 7.55 | 18 | 1174.35 | 7.26 |
| 9 | 1149.81 | 7.58 | 19 | 1174.50 | 7.27 |
| 10 | 1148.67 | 7.62 | 20 | 1180.34 | 7.29 |

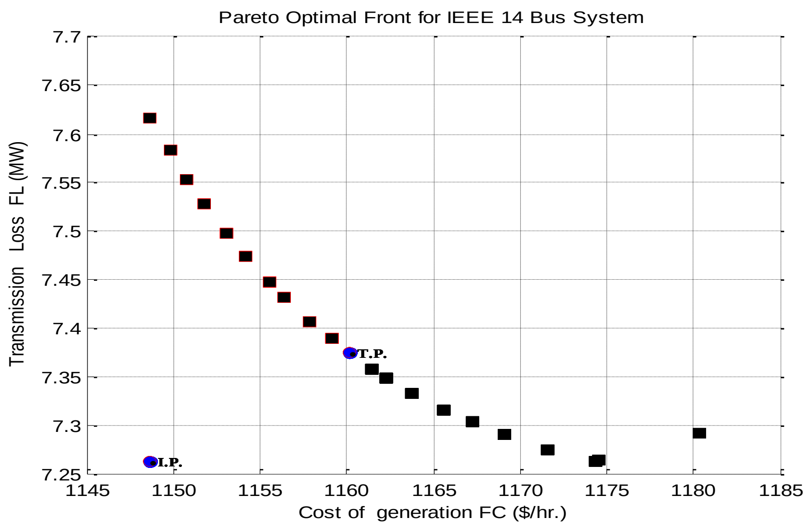



**Fig. 7.8** Pareto Optimal Front for IEEE 14 Bus System

## IEEE 30 Bus System

In this case, six hundred (600) points are randomly generated. Out of these, four hundred points which satisfy the equality constraints of the problem are identified and stored in the Qualifying set.. Qualifying set for IEEE 30 bus systems is shown in Fig. 7.9.

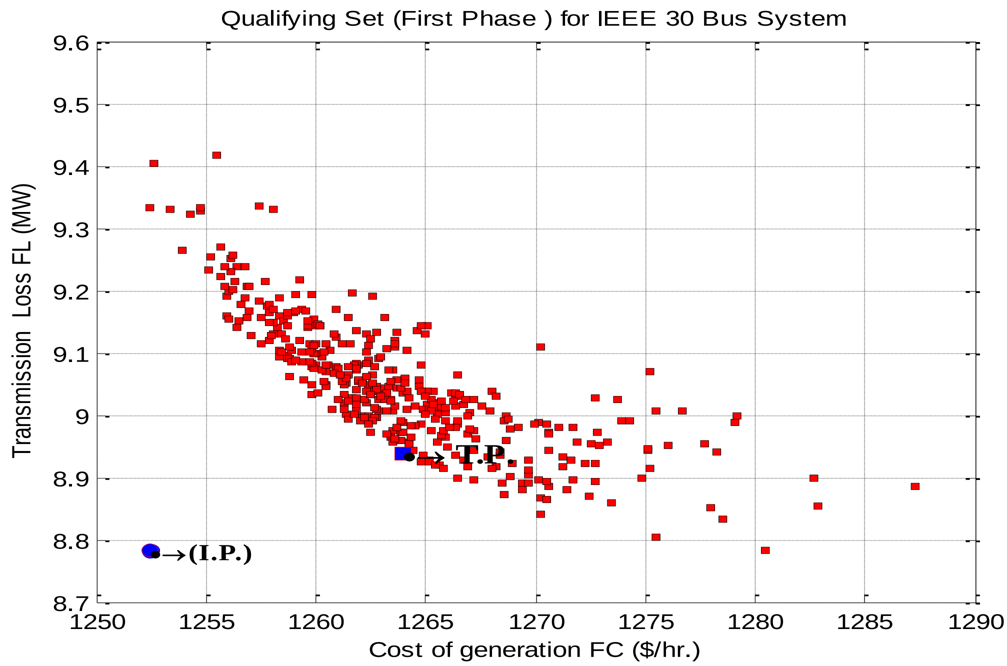



**Fig.7.9** Qualifying Set (First Phase) for IEEE 30 Bus System

The points of Pareto optimal Front are selected by identifying the points which lie at minimum distance from the equidistant points along each axis. These points are shown in Table 7.4 and Fig. 7.10.

**TABLE 7. 4**
**Pareto Optimal Front for IEEE 30 bus system**

| S. No. | $F_C$ ($/hr.) | $F_L$ (MW) | S. No. | $F_C$ ($/hr.) | $F_L$ (MW) |
|---|---|---|---|---|---|
| 1 | 1252.5 | 9.40 | 10 | 1264.0 | 8.94 |
| 2 | 1252.6 | 9.33 | 11 | 1264.8 | 8.93 |
| 3 | 1253.9 | 9.26 | 12 | 1265.0 | 8.90 |
| 4 | 1255.9 | 9.16 | 13 | 1266.4 | 8.94 |
| 5 | 1256.4 | 9.14 | 14 | 1270.2 | 8.84 |
| 6 | 1258.8 | 9.06 | 15 | 1275.5 | 8.80 |
| 7 | 1260.8 | 9.01 | 16 | 1275.5 | 8.80 |
| 8 | 1261.4 | 8.99 | 17 | 1280.4 | 8.78 |
| 9 | 1264.0 | 8.94 | 18 | 1287.3 | 8.88 |

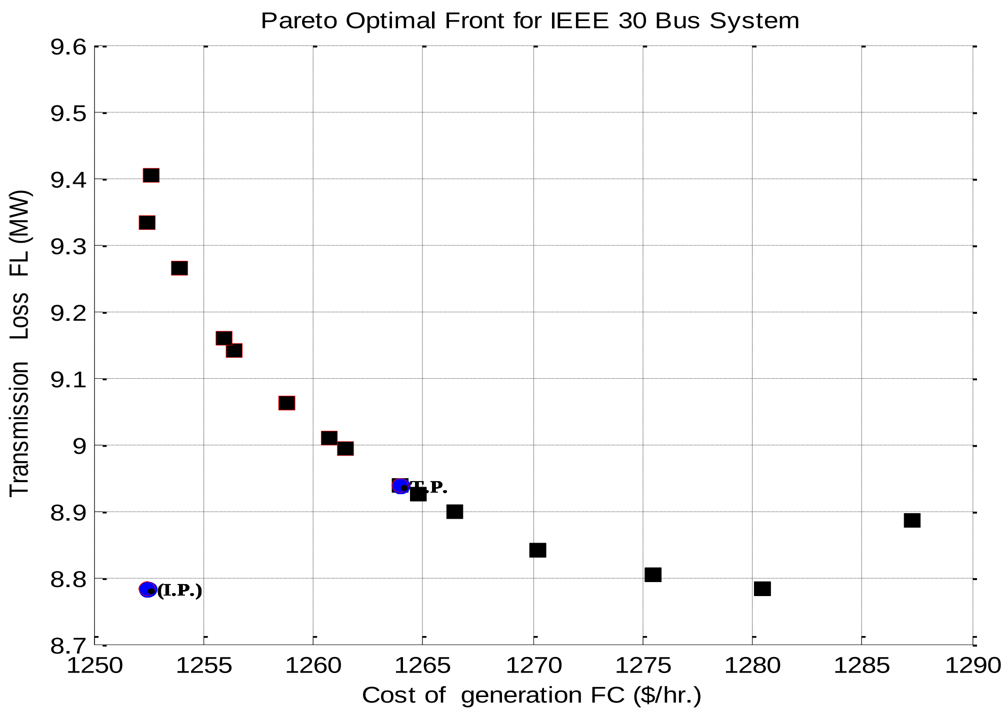



**Fig.7.10** Pareto Optimal Front for IEEE 30 Bus System

The Best Compromise solution/ Target point (T.P.) i.e. ($F_C$*, $F_L$*) for all the systems is obtained by identifying the point which lies at minimum distance from Ideal point (I.P.) using equation (7.9). Table 7.5 shows the Best Compromise solution (T.P.) i.e. ($F_C$*, $F_L$*) for IEEE 5, 14 and 30 bus systems.

**TABLE 7. 5**
**Best Compromise Solutions in 2-D space**

| IEEE System | $F_C$*($/hr.) | $F_L$*(MW) |
|---|---|---|
| 5 bus | 761.46 | 5.09 |
| 14 bus | 1160.15 | 7.38 |
| 30 bus | 1264.00 | 8.93 |

Table 7.6 shows the comparison of Best Compromise solutions / Target points for IEEE 5 bus,14 bus and 30 bus systems using proposed algorithm, Genetic Algorithm [113] and PSO [160].

**TABLE 7.6**
**Comparison of Best Compromise Solutions in 2-D space**

| | FOPSO Algorithm | | PSO | | Genetic Algorithm | |
|---|---|---|---|---|---|---|
| **IEEE System** | **$F_C$*($/hr.)** | **$F_L$*(MW)** | **$F_C$*($/hr.)** | **$F_L$*(MW)** | **$F_C$*($/hr.)** | **$F_L$*(MW)** |
| 5 bus | 761.46 | 5.09 | 761.67 | 5.14 | 761.49 | 5.09 |
| 14 bus | 1160.15 | 7.38 | 1183.28 | 6.76 | 1150.89 | 7.60 |
| 30 bus | 1264.00 | 8.93 | 1318.05 | 7.57 | 1295.63 | 8.20 |

For IEEE 5 bus system, cost of generation obtained by the proposed algorithm is lesser than that obtained by PSO and GA. In case of GA, the cost of generation obtained by the proposed algorithm is lesser for the same transmission losses. For IEEE 14 bus system, the cost of generation obtained by proposed algorithm is lesser than that obtained by PSO but slightly higher (0.8%) than that obtained by GA. For IEEE 30 bus system, cost of generation obtained by the proposed algorithm is minimum but there is small increase in transmission losses compared to that obtained by PSO and GA.

In general, the Pareto-Optimal Front in a multiobjective optimization problem is obtained by running the problem multiple times. In this chapter, well distributed Pareto Front of twenty points has been obtained in a partial run for IEEE 5 bus, 14 bus and 30 bus systems by implementing Feasibility Oriented Particle Swarm Optimization (FOPSO) algorithm. In the first phase, a large population of points is chosen for each system so as to maintain the diversity. The first phase ends when sufficient number of feasible points for the Qualifying set are obtained. For IEEE 5, 14 and 30 bus systems, initial populations of 400, 600 and 600 is chosen respectively. The number of points to be copied to the Qualifying set are chosen to be 300, 200 and 400 for IEEE 5, 14 and 30 bus systems respectively. In the second phase, ten points on each axis which are at minimum distance from the equidistant points are identified. Therefore, the Pareto-Front is obtained before the PSO algorithm is completely executed for whole population.

### 7.5.2 Results of MELD in 3-D Space

The feasible points obtained from the ***First phase*** are stored in the Qualifying set. From these points, $F_{Cmin}$, $F_{Cmax}$, $F_{Lmin}$, $F_{Lmax}$, $F_{Emax}$ and $F_{Emin}$ are identified. These are then used to calculate the range of cost of generation ($F_{Crange}$), range of transmission losses ($F_{Lrange}$) and range of Emission ($F_{Erange}$). Table 7.7 shows the range of $F_C$, $F_L$ and $F_E$ for IEEE 5, 14 and 30 bus systems for 3-D space. Row (1) shows the IEEE bus system.

**TABLE 7.7**
**Ranges of $F_C$, $F_L$, $F_E$**

| Row No. 1 | | IEEE 5 BUS | IEEE 14 BUS | IEEE 30 BUS |
|---|---|---|---|---|
| 2 | $F_{Cmax}$ | 768.64 | 1171.39 | 1298.48 |
| 3 | $F_{Cmin}$ | 760.96 | 1140.74 | 1253.46 |
| 4 | $F_{Crange}$ | 7.68 | 30.65 | 45.02 |
| 5 | $F_{Lmax}$ | 5.29 | 7.99 | 9.25 |
| 6 | $F_{Lmin}$ | 5.06 | 7.29 | 8.76 |
| 7 | $F_{Lrange}$ | 0.23 | 0.71 | 0.50 |
| 8 | $F_{Emax}$ | 135.32 | 561.23 | 572.41 |
| 9 | $F_{Emin}$ | 118.79 | 501.01 | 519.13 |
| 10 | $F_{Erange}$ | 16.53 | 60.22 | 53.28 |

From the results of Table 7.7, Ideal points I.P. ($F_{Cmin}$, $F_{Lmin}$, $F_{Emin}$) can be identified for IEEE 5, 14 and 30 bus systems. This is represented by rows (3), (6) and (9) of Table 7.7. The ideal point for IEEE 5 bus, 14 bus and 30 bus systems are (760.96, 5.06, 118.79), (1140.74, 7.29, 501.01) and (1253.46, 8.76, 519.13) respectively. $F_C$, $F_L$, and $F_E$ axis are then divided into twenty equidistant points between $F_{Cmin}$ and $F_{Cmax}$ along $F_C$ axis and between $F_{Lmin}$ and $F_{Lmax}$ along $F_L$ axis and $F_{Emin}$ and $F_{Emax}$ along $F_E$ axis. The Pareto optimal front is obtained by identifying the points which lie at minimum distance from these equidistant points along each axis. A total of sixty points are obtained for Pareto-Optimal Front.

## IEEE 5 Bus System

Initially, a population of 1000 points is generated randomly by MATLAB Programme. Out of this 300 points which satisfy the equality constraints of the problem are copied to the Qualifying Set. This is shown in Fig.7.11.

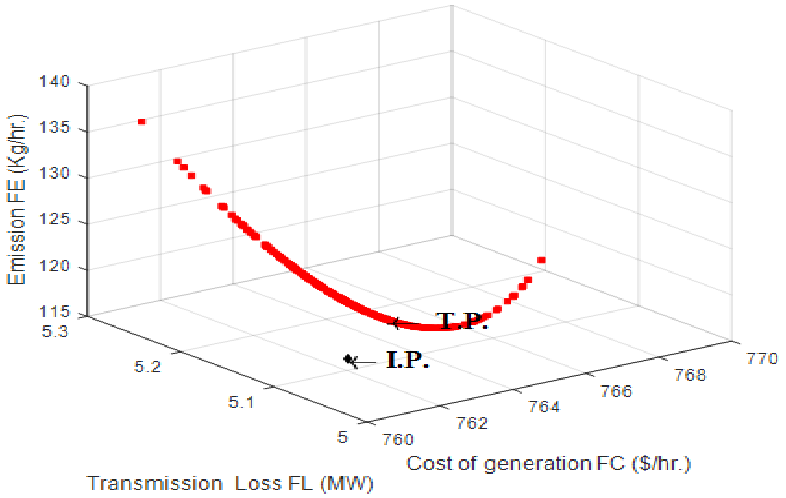



**Fig. 7.11** Qualifying Set for IEEE 5 Bus System in 3D Space

From the points of Qualifying Set, IP, $F_{Cmax}$, $F_{Lmax}$ and $F_{Emax}$ are identified. All three axes ($F_C$, $F_L$ and $F_E$) are divided into twenty equidistant points. Fig. 7.12 represent the Pareto front of the proposed algorithm for IEEE 5 bus system.

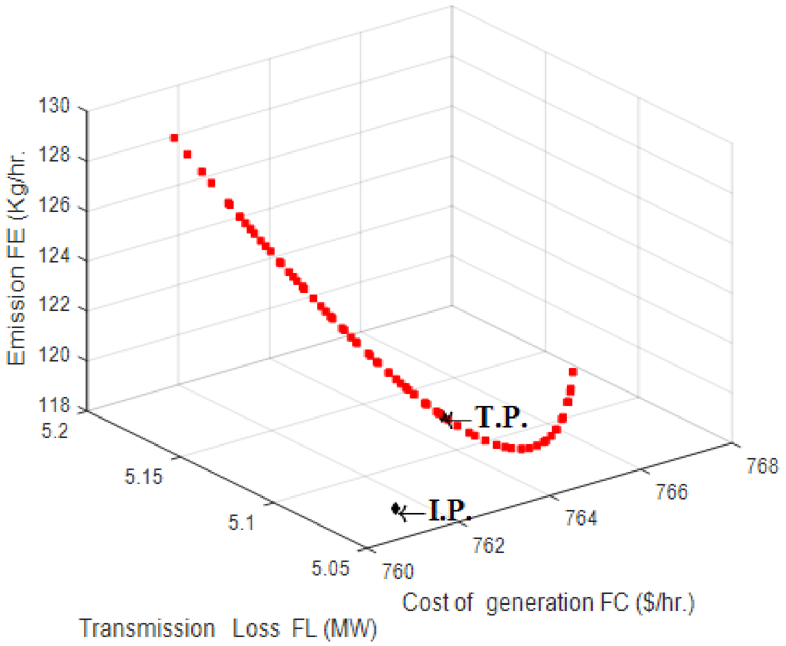



**Fig. 7.12** Pareto Front for IEEE 5 Bus System in 3D Space

The points of Qualifying which lie at minimum distance from the equidistant points are shown in Fig. 7.12. This forms the Pareto Front of IEEE 5 bus system. Fig.7.13 represents the Qualifying set and Fig.7.14 shows Pareto front in 2D space considering cost of generation and transmission losses objectives for IEEE 5 bus system.

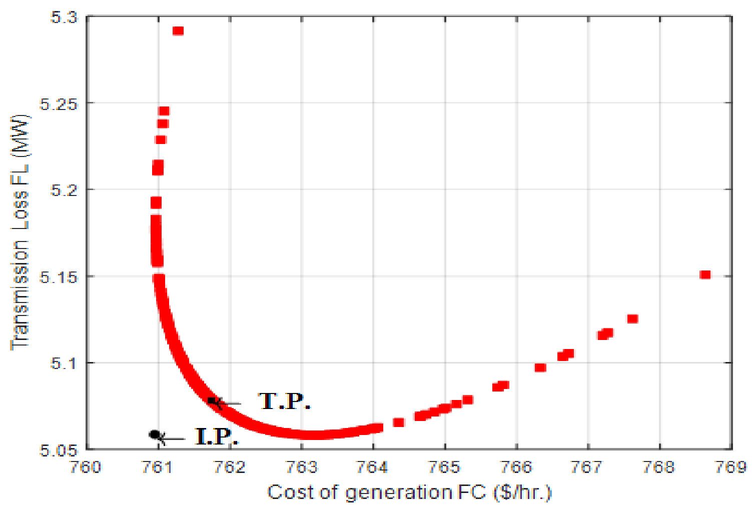



**Fig. 7.13** Qualifying Set for IEEE 5 Bus System ($F_C$-$F_L$)

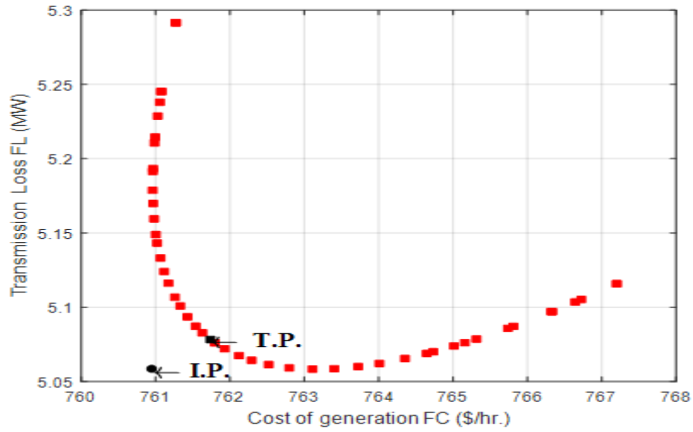



**Fig. 7.14** Pareto Front for IEEE 5 Bus System ($F_C$-$F_L$)

Fig. 7.15 and Fig.7.16 shows the Qualifying set and Pareto Front in 2D space for Cost of generation and Emission objectives for IEEE 5 bus system.

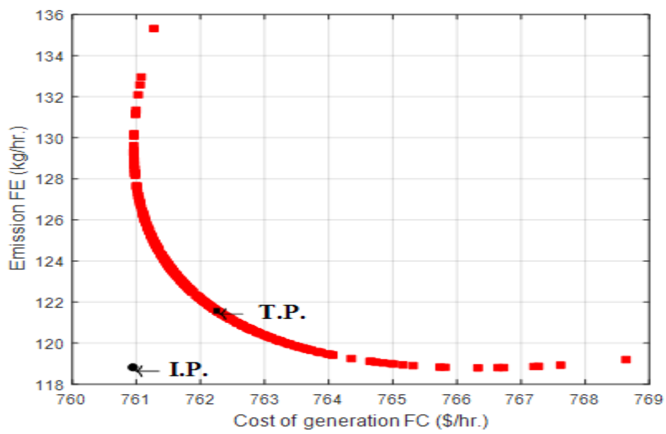



**Fig.7.15** Qualifying Set for IEEE 5 Bus System ($F_C$-$F_E$)

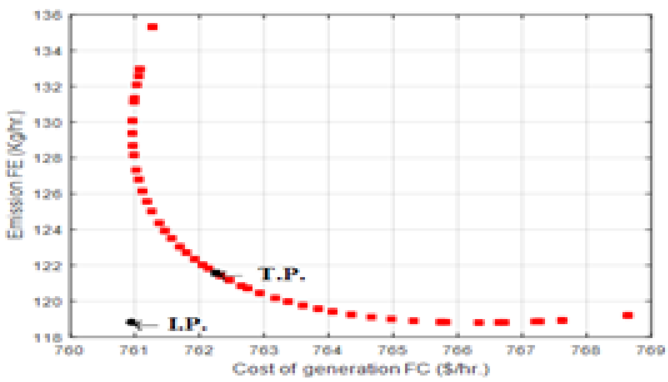



**Fig. 7.16** Pareto Front for IEEE 5 Bus System ($F_C$-$F_E$)

Fig.7.17 and 7.18 represent the Qualifying set of feasible points and Pareto Front respectively for IEEE 5 bus systems in 2D space for transmission losses and emission.

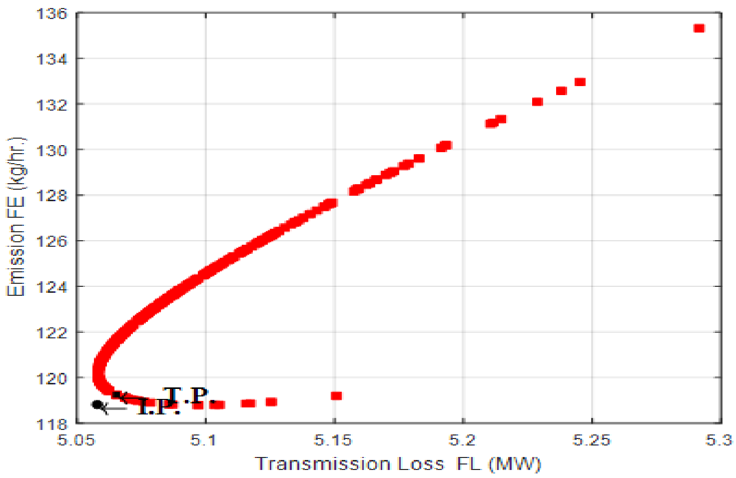



**Fig. 7.17** Qualifying Set for IEEE 5 Bus System ($F_L$-$F_E$)

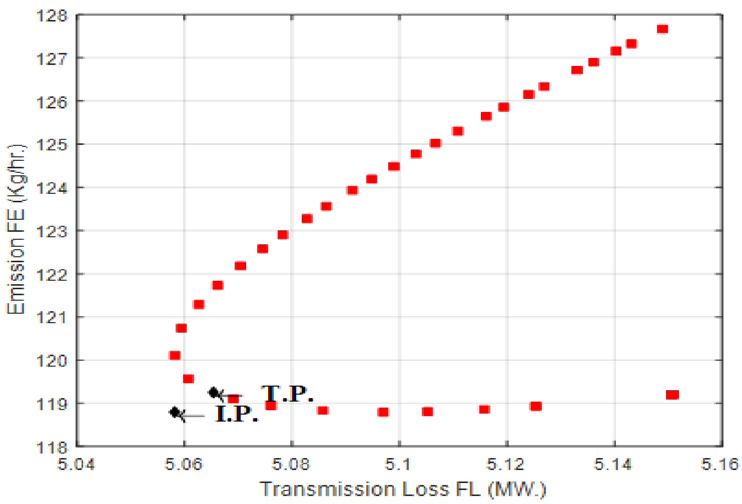



**Fig. 7.18** Pareto Front for IEEE 5 Bus System ($F_L$-$F_E$)

## IEEE 14 Bus System

Initially, a population of 3000 points is generated randomly by MATLAB Programme. Out of this 1000 points which satisfy the equality constraints of the problem are copied to the Qualifying Set. Qualifying set of feasible points for IEEE 14 bus system in 3D space is shown in Fig. 7.19.

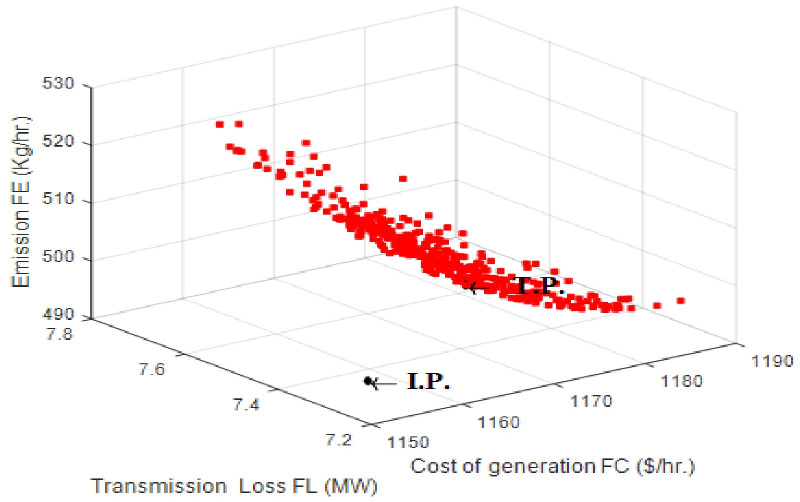



**Fig. 7.19** Qualifying Set for IEEE 14 Bus System in 3D space

The Pareto Front of IEEE 14 bus system is shown in Fig.7.20.

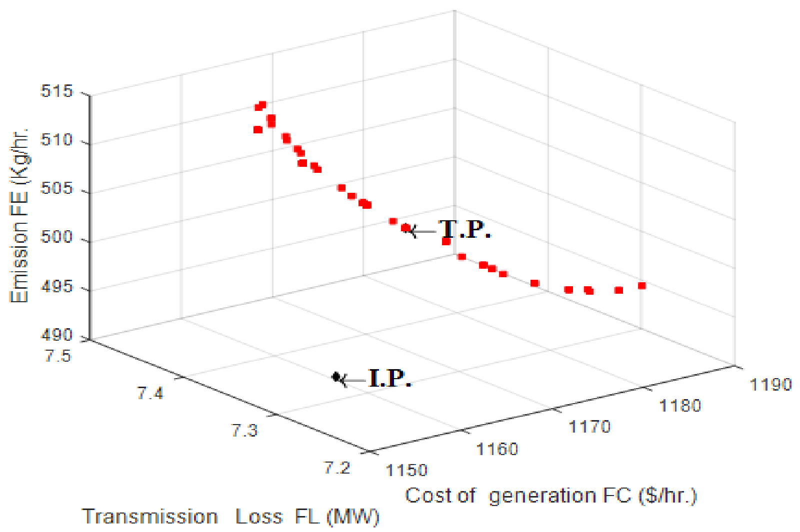



**Fig. 7.20** Pareto Front for IEEE 14 Bus System in 3D system

The Qualifying set and Pareto – optimal Front for all combinations of two objectives are derived from the Qualifying set and Pareto Optimal Front obtained in 3-D space.

Qualifying set and Pareto Optimal front for ($F_C$-$F_L$) is shown in Fig.7.21 and Fig.7.22 respectively. Qualifying set and Pareto Optimal Front for ($F_C$-$F_E$) is shown in Fig.7.23 and 7.24 respectively. Similarly the Qualifying set and Pareto Optimal front for ($F_L$-$F_E$) is shown in Fig. 7.25 and 7.26 respectively..

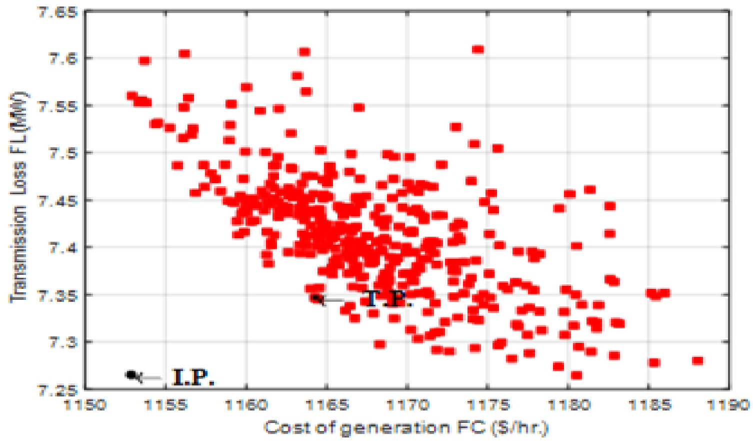



**Fig. 7.21** Qualifying Set for IEEE 14 Bus System ($F_C$-$F_L$)

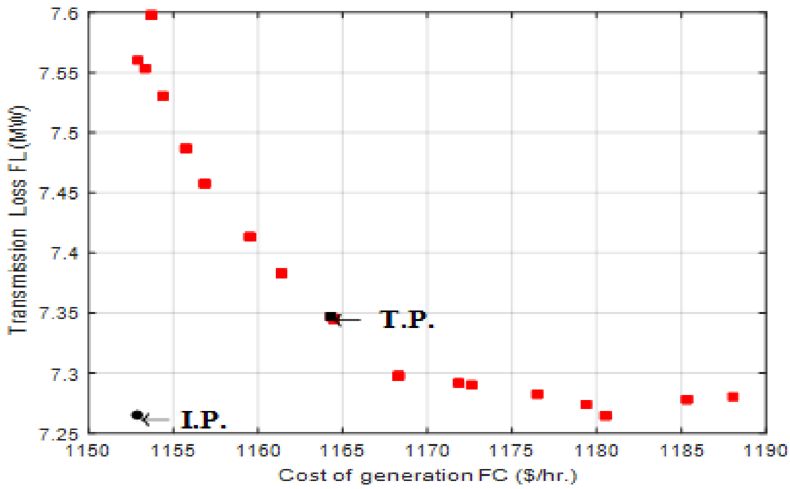



**Fig. 7.22** Pareto Front of IEEE 14 Bus System ($F_C$-$F_L$)

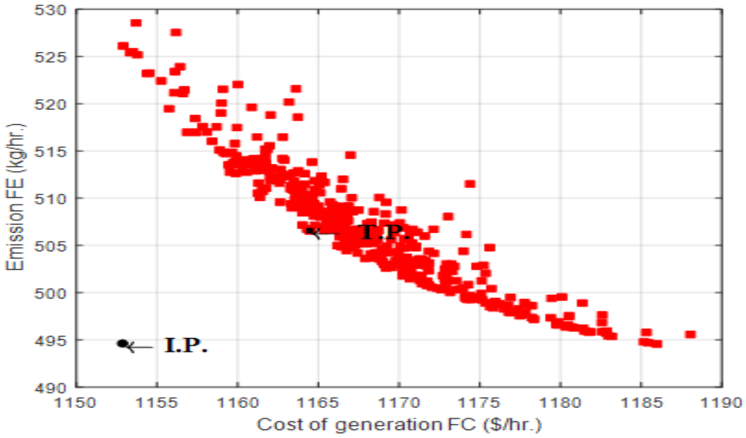



**Fig. 7.23** Qualifying Set for IEEE14 Bus System ($F_C$-$F_E$)

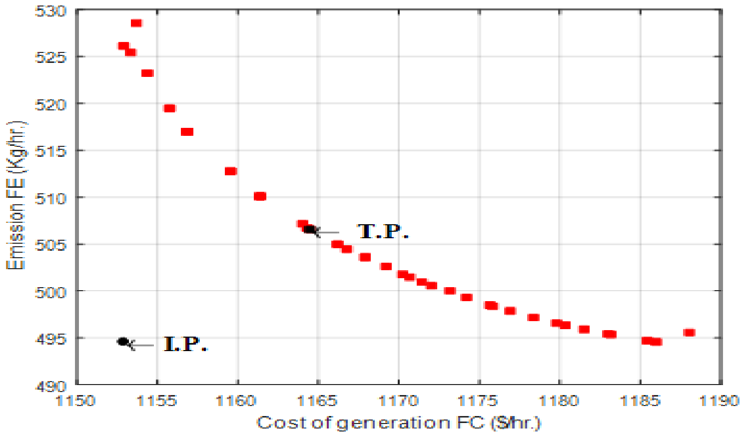



**Fig. 7.24** Pareto Front for IEEE 14 Bus System($F_C$-$F_E$)

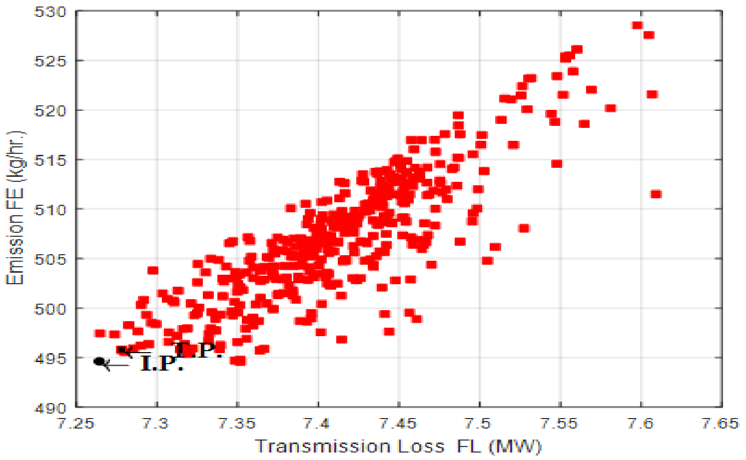



**Fig. 7.25** Qualifying Set for IEEE 14 Bus System ($F_L$-$F_E$)

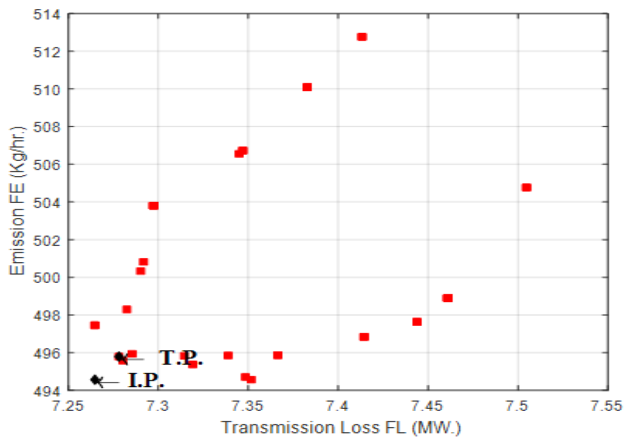



**Fig. 7.26** Pareto Front for IEEE 14 Bus System ($F_L$-$F_E$)

## IEEE 30 Bus System

Initially, a population of 300 points is generated. Out of this 200 points which satisfy the equality constraints of the problem are copied to the Qualifying Set. Qualifying set and Pareto Front in 3-D space for MELD problem of IEEE 30 system is shown in Fig.7.27. and Fig. 7.28 respectively.

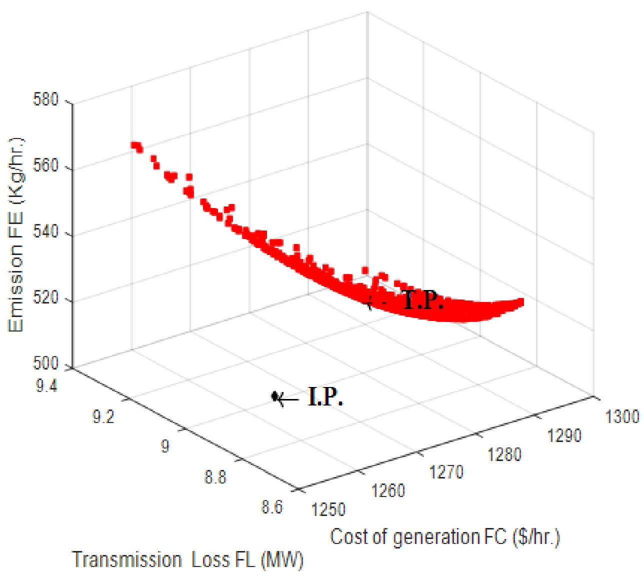



**Fig. 7.27** Qualifying Set for IEEE 30 Bus System in 3D space

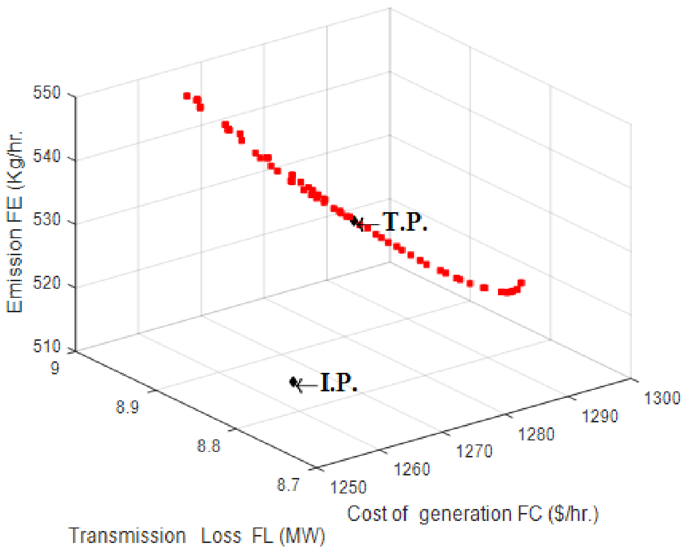



**Fig.7.28** Pareto Front for IEEE 30 Bus System in 3D Space

Qualifying set and Pareto Optimal front for ($F_C$-$F_L$) is shown in Fig.7.29 and Fig.7.30 respectively. Qualifying set and Pareto Optimal Front for ($F_C$-$F_E$) is shown in Fig.7.31 and Fig. 7.32 respectively. Similarly, the Qualifying set and Pareto Optimal front for ($F_L$-$F_E$) is shown in Fig. 7.33 and Fig. 7.34 respectively.

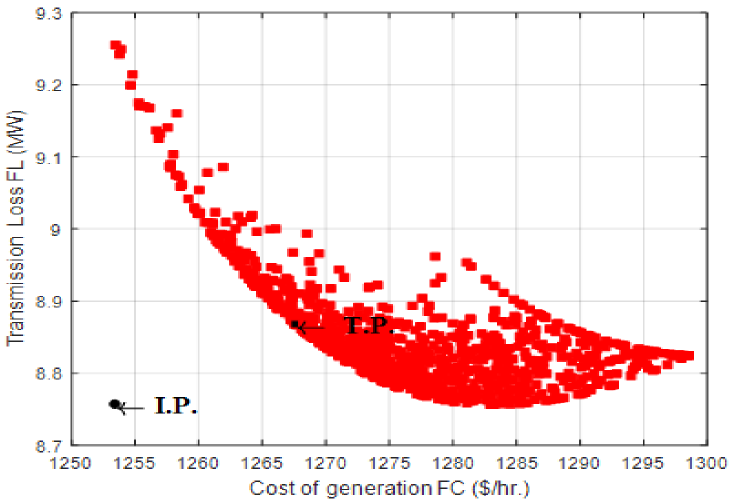



**Fig. 7.29** Qualifying Set for IEEE 30 Bus System($F_C$-$F_L$)

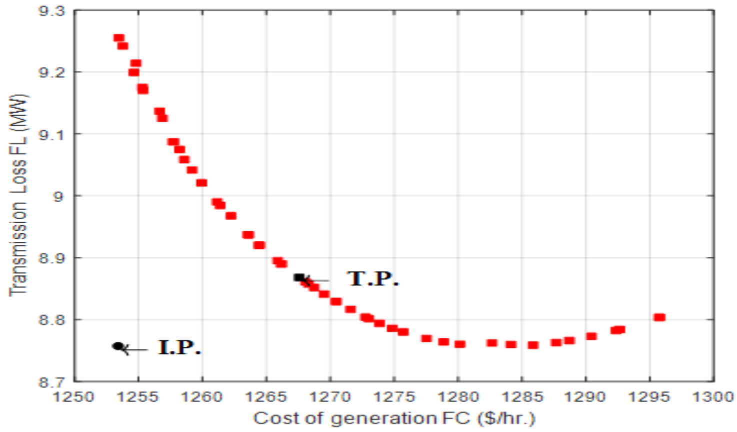



**Fig.7.30** Pareto Front for IEEE 30 Bus System($F_C$-$F_L$)

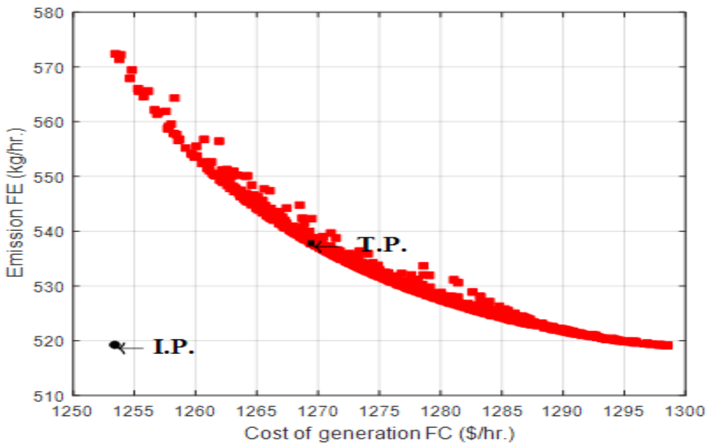



**Fig. 7.31** Qualifying Set for IEEE 30 Bus System($F_C$-$F_E$)

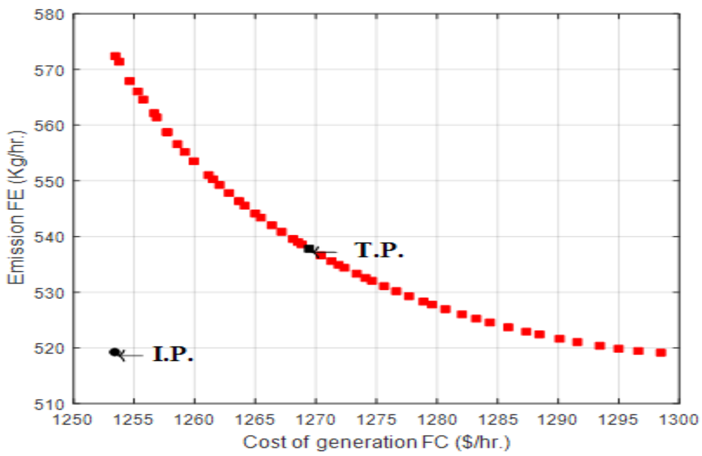



**Fig.7.32** Pareto Front for IEEE 30 Bus System ($F_C$-$F_E$)

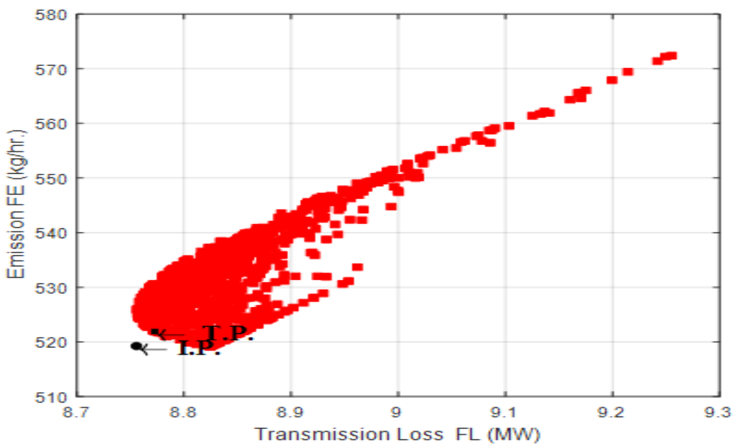



**Fig. 7.33** Qualifying Set for IEEE 30 Bus System ($F_L$-$F_E$)

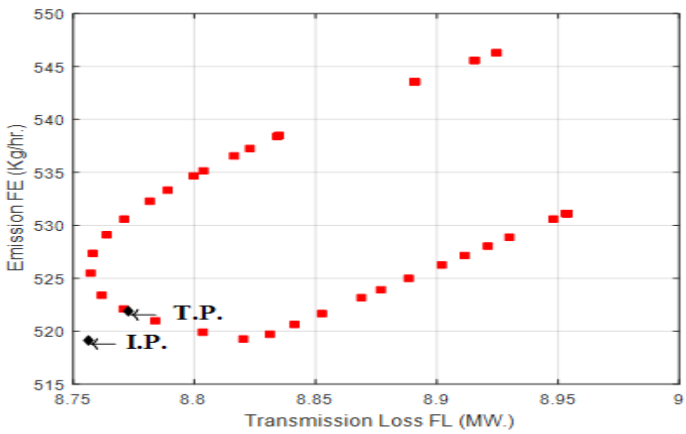



**Fig. 7.34** Pareto Front for IEEE 30 Bus System ($F_L$-$F_E$)

Table 7.8 shows the Target Points for IEEE 5, 14 and 30 bus systems in 3D Space. This is achieved when the distance between IP and the set of Pareto-Optimal solution is minimum.

**TABLE 7.8**
**Target Point for MELD problem in 3D space**

| IEEE System | $F_C$*($/hr.) | $F_L$*(MW) | FE*(kg/hr.) |
|---|---|---|---|
| 5 bus | 762.2537 | 5.064763 | 121.5713 |
| 14 bus | 1154.009 | 7.472081 | 519.9834 |
| 30 bus | 1271.64 | 8.822827 | 535.3162 |

Target point in 2-D systems for various combinations of objectives i.e. $F_C$-$F_L$, $F_C$-$F_E$, $F_L$ - $F_E$ are shown in Table 7.9 for MELD problem of IEEE 5,14 and 30 bus systems.

**TABLE 7.9**
**Target Point for MELD problem in 2D space**

| **S. No.** | **IEEE Bus System** | **Target Point in 2 D space for $F_C$-$F_L$** | | **Target Point in 2 D space for $F_C$-$F_E$** | | **Target Point in 2 D space for $F_L$ - $F_E$** | |
|---|---|---|---|---|---|---|---|
| | | $F_C$*($/hr.) | $F_L$*(MW) | $F_C$*($/hr.) | FE*(kg/hr.) | $F_L$* MW) | FE*(kg/hr.) |
| 1 | 5 bus | 761.738 | 5.078 | 762.2537 | 121.571 | 5.065 | 119.248 |
| 2 | 14 bus | 1151.009 | 7.540 | 1152.442 | 522.783 | 7.290 | 501.008 |
| 3 | 30 bus | 1267.576 | 8.868 | 1269.442 | 537.798 | 8.772 | 521.887 |

Comparison of Target points / Best compromise solution in 3D space are shown in Table 7.10.

**TABLE 7.10**
**Target point of MELD problem in 3-D space**

| | **GA** | | | **FOPSO** | | |
|---|---|---|---|---|---|---|
| **IEEE Bus System** | **$F_C$*($/hr.)** | **$F_L$*(MW)** | **FE*(kg/hr.)** | **$F_C$*($/hr.)** | **$F_L$*(MW)** | **FE*(kg/hr.)** |
| 5 | 761.66 | 5.08 | 123 | 762.25 | 5.06 | 121.57 |
| 14 | 1163.28 | 7.28 | 508 | 1154.00 | 7.47 | 519.98 |
| 30 | 1273.65 | 9.44 | 538 | 1271.64 | 8.82 | 535.31 |

In general, the Pareto-Front for a multiobjective optimization problem is obtained by running the problem multiple times. In this chapter, FOPSO algorithm has been developed to solve MELD problem and to generate uniformly distributed Pareto Front of twenty points in 2-D space and sixty points in 3-D space in a partial run for

IEEE 5 bus, 14 bus and 30 bus systems. It works in two phases, in the first phase, a large population of points is chosen for each system so as to maintain the diversity. The first phase ends when sufficient number of feasible points for the Qualifying set are obtained. For IEEE 5, 14 and 30 bus system initial populations of 1000, 3000 and 300 is chosen respectively. The number of points to be copied to the Qualifying set are chosen to be 300, 1000 and 200 for IEEE 5, 14 and 30 bus systems respectively. In the second phase twenty points on each axis which are at minimum distance from the equidistant points are identified. Therefore, the Pareto-Front is obtained before the PSO algorithm is completely executed for whole population.

## 7.6 CONCLUSIONS

In this chapter, Feasibility Oriented Particle Swarm Optimization (FOPSO) Algorithm has been developed. This algorithm is capable of producing uniformly distributed Pareto Optimal Front in a partial run thus overcoming the need of multiple runs to generate the Pareto - Optimal Front. It has been successfully implemented on Multiobjective Economic Load Dispatch problem of IEEE 5 bus, 14 bus and 30 bus systems to obtain uniformly distributed Pareto –Optimal Front in 2-D and 3-D space. In 2-D space, two objectives cost of generation and system transmission losses have been considered whereas in 3-D space environmental emission is also considered in addition to the above mentioned objectives.

From 3-D Pareto- Optimal Front, 2-D analysis of all three combinations of objectives has been carried out and there is no need run to the programme for each combination separately. The proposed algorithm works in two phases. In the first phase, it identifies the feasible points whereas in the second phase it identifies the uniformly distributed Pareto-Optimal Front. The computational effort required for the first phase forms the major part of computational effort and that required by the second phase is quite small. Further, the proposed algorithm has no limitation in handling the more than three objectives.

## Research Publications

[1] N. K. Jain, Uma Nangia, Jyoti Jain, "Feasibility oriented PSO (FOPSO) algorithm to determine Pareto - Optimal front in a partial run for multiobjective economic load dispatch considering cost of generation and system transmission losses." Journal of the Institution of Engineers India: Series B, Springer.(Communicated)

[2] N. K. Jain, Uma Nangia, Jyoti Jain, "Feasibility oriented PSO (FOPSO) algorithm to determine Pareto - Optimal front in a partial run for multiobjective economic load dispatch considering cost of generation and system transmission losses and environmental emission". Communicated to Applied Soft Computing, Elsevier.

# CHAPTER 8
# CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH WORK

## 8.1 INTRODUCTION

The focus of this research work has been to solve Economic Load Dispatch (ELD) and Multiobjective Economic Load Dispatch (MELD) problem by Intelligent techniques mainly BPSO and GA. These techniques have also been implemented on mathematical bench mark functions. Intelligent techniques have been found to be slower than conventional techniques. So, an attempt has been made to modify and improve intelligent techniques for Power system problems - ELD and MELD problem. BPSO, Improved and modified PSO techniques have been implemented on ELD and MELD problem considering cost of generation, system transmission losses and environmental emission for IEEE 5,14 and 30 bus systems.

Complete Pareto Optimal Front has been obtained and the Target point has been identified by the Power System Operator depending on his own requirements.

## 8.2 CONCLUSIONS

Following are the conclusions based on the results of previous chapters.

1 Literature survey reveals that intelligent techniques are slower than conventional techniques computationally.

2 Conventional techniques are capable of reaching a local optimization, whereas, intelligent techniques can search for global optimum.

3 On one hand, the research work for making intelligent techniques computationally

more efficient is called for. On the other hand, the strengths of intelligent optimization need be explored.

4 A sincere attempt has been made in both these directions.

5 GA has been implemented on various mathematical test function. A relation between generation and population size has been established to make GA computationally more efficient.

6 It has also been observed that population size and bit size should be large enough so that it can support sufficient genetic variation and therefore, higher accuracy can be achieved.

7 BPSO has also been implemented on Rosenbrock function manually as well as by MATLAB programme. The effect of population size and maximum number of iterations on accuracy has been obtained, maximum accuracy and fast convergence is achieved for population size greater than 40. Further, the population size should not be less than 20 to optimize the function accurately.

8 Two improvement in basic PSO have been suggested. These are: Improved particle swarm optimization based on Initial selection of Particles (IPSO IS), Adaptive Social Acceleration Constant based PSO (ASACPSO). These new algorithms have been implemented on mathematical test functions as well as Economic Load Dispatch problem. Both algorithms proved to be much more efficient as compared to basic PSO. However, IPSO IS algorithm proves to the best from computational effort point of view.

9 An algorithm named as Split Phase Economic Load Dispatch Algorithm has been designed. This proved to be more efficient computationally than the basic PSO. Further, the new algorithm outperforms the lambda iteration method.

10 Multiobjective Economic Load Dispatch (MELD) problem has been formulated using weighting method and constraint method. GA and PSO techniques have been used for generating noninferior set. Ideal point as well as target point has been identified. Target point has been searched by using Fuzzy Logic System and Maximization of Minimum Relative attainments (MMR).

11 The behavior of various objectives of Electric Power System namely Cost of generation, System transmission losses and Environmental Emissions have been studied. This revealed that cost of generation conflicts with transmission losses and as well environmental emission. However, System transmission losses and environmental emissions are supportive in nature. The Objectives may not follow this behavior in all domains.

12 Weighting method gives a Pareto Front which is not evenly distributed. These results may help for decision making to some extent. However, constraint method gives an evenly distributed Pareto Optimal Front. These results are more convenient for decision making.

13 A Feasibility Oriented Particle Swarm Optimization (FOPSO) algorithm has been developed. This algorithm is used to generate Pareto optimal front in 3-D space considering cost of generation, system transmission losses and environmental emission in a less than single run. From 3-D Optimal Front, 2-D analysis of all three combinations of objectives has been carried out. The proposed algorithm has no limitation in handling more than three objectives.

## 8.3 SUGGESTIONS FOR FUTURE RESEARCH WORK

Following are the suggestions for future research work:

1. The objectives considered in the present research work are – Cost of Generation ($F_C$), System Transmission Losses ($F_L$) and Environmental Pollution ($F_E$) which are ***noncommensurable*** because of their nature. However, transmission losses and environmental emission can be expressed in monetary units to be compatible with the cost of generation. This objective function can be minimized to reach the Target Point (TP) or the best – compromise solution.
2. Neural networks can be used to predict the load demand and to identify the noninferior set from a set of feasible solutions.
3. An attempt can be made to include more objectives of the power systems like security, integration of wind energy and solar energy etc. in Multiobjective Economic Load Dispatch problem. Such analysis will provide better understanding of the power systems problem with the application of Renewable energy.
4. Fuzzy set theory has been used to model the objectives of power systems. However, it can be used to model the power systems constraints as well.
5. Feasibility oriented PSO can be used for objectives of renewable energy sources.

# APPENDIX- I

## IEEE 5-Bus System

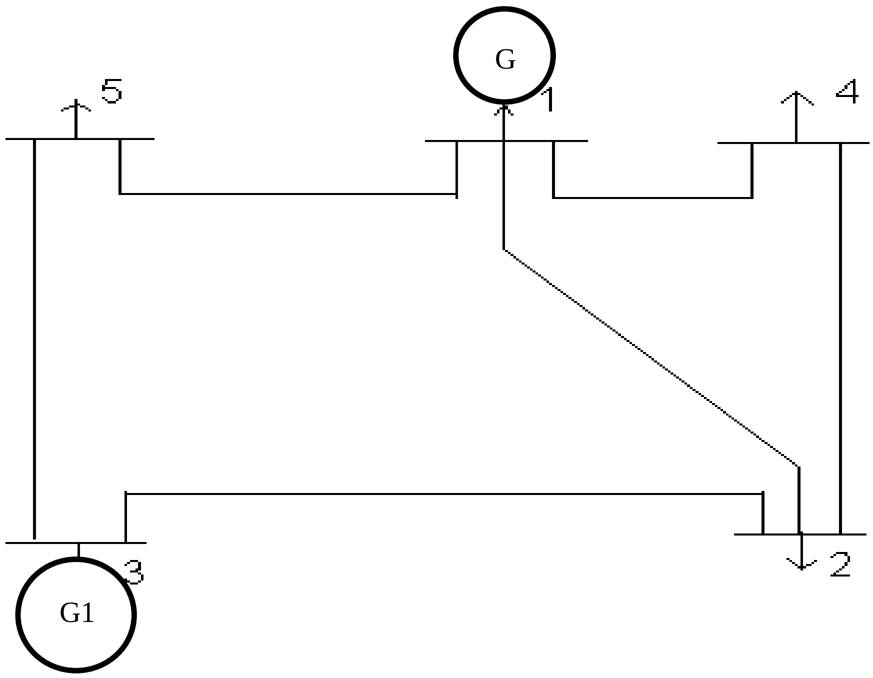



**Fig. 1 IEEE 5 - Bus System**

**TABLE 1**
**Line data / Impedance data**

| Line signation | *R (p.u.) | *X (p.u.) | Line charging |
|---|---|---|---|
| 1-2 | 0.10 | 0.4 | 0.0 |
| 1-4 | 0.15 | 0.6 | 0.0 |
| 1-5 | 0.05 | 0.2 | 0.0 |
| 2-3 | 0.05 | 0.2 | 0.0 |
| 2-4 | 0.10 | 0.4 | 0.0 |
| 3-5 | 0.05 | 0.2 | 0.0 |

* The impedances are based on MVA as 100.

**TABLE 2**
**Bus data / Operating conditions (IEEE - 5 bus system)**

| | Generation | | Load | |
|---|---|---|---|---|
| **Bus No.** | **MW** | **Voltage Magnitude** | **MW** | **MVAR** |
| 1* | ------- | 1.02 | ------- | ------- |
| 2 | ------- | ------- | 60 | 30 |
| 3 | 100 | 1.04 | ------- | ------- |
| 4 | ------- | ------- | 40 | 10 |
| 5 | ------- | ------- | 60 | 20 |

*Slack Bus

**TABLE 3**
**Regulated Bus data (IEEE 5-Bus System)**

| Bus No. | Voltage magnitude | Minimum MVAR capability | Maximum MVAR capability | Minimum MW capability | Maximum MW capability |
|---|---|---|---|---|---|
| 1 | 1.02 | 0.0 | 60 | 30 | 120 |
| 3 | 1.04 | 0.0 | 60 | 30 | 120 |

The nodal load voltage inequality constraints are $0.9 \leq V_i \leq 1.05$

## Cost Characteristics

The cost characteristics of the IEEE 5 Bus System are as follows:

$C_1 = 50 \ P_{g1}^2 + 351 \ P_{g1} + 44.4$ \$ / hr.

$C_3 = 50 \ P_{g3}^2 + 389 \ P_{g3} + 40.6$ \$ / hr.

Total load demand of the system is $P_D = 160$ MW.

Maximum and minimum active power constraint on the generator bus for the given system is 120 MW and 30 MW respectively.

## Emission Characteristics

$E_1 = 135.5 \ P_{g1}^2 - 126.5 \ P_{g1} + 22.9$ Kg/hr.

$E_3 = 124.8 \ P_{g3}^2 - 137.8 \ P_{g3} + 137.3$ Kg/hr.

## B - Coefficients in $\mathbf{MW^{-1}}$

$B_{11} = 0.00035336$ $B_{12} = B_{21} = 0.0000103196$ $B_{22} = 0.000368992$

## (2) IEEE 14 - BUS SYSTEM

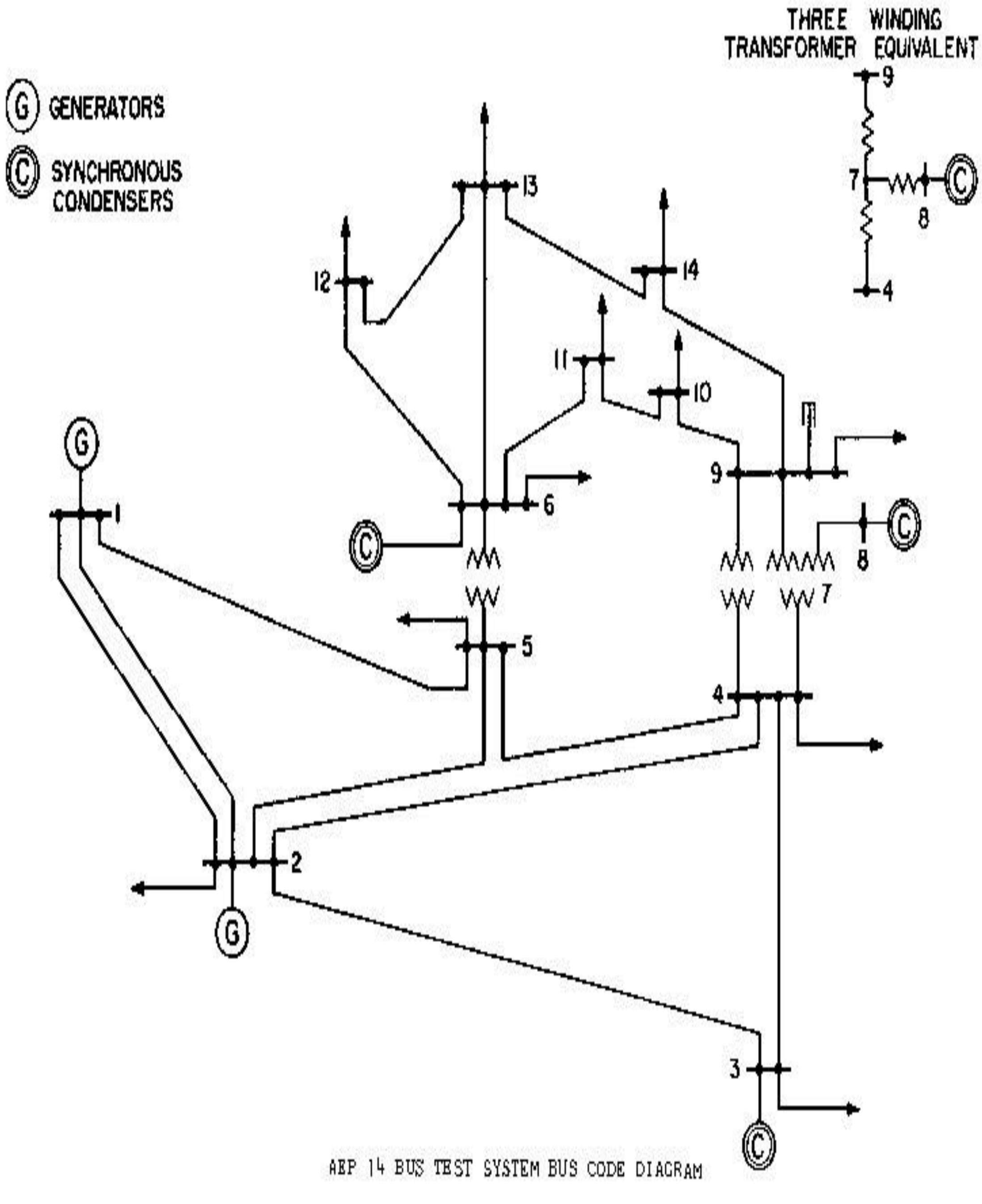



**Fig. 2** IEEE 14 Bus System

**TABLE 4**
**Impedance and Line-charging Data (IEEE 14 Bus System)**

| Line Designation | Resistance p.u.* | Reactance p.u.* | Line Charging | Tap Setting |
|---|---|---|---|---|
| 1-2 | 0.01938 | 0.05917 | 0.0264 | 1 |
| 1-5 | 0.05403 | 0.22304 | 0.0246 | 1 |
| 2-3 | 0.04699 | 0.19797 | 0.0219 | 1 |
| 2-4 | 0.05811 | 0.17632 | 0.0187 | 1 |
| 2-5 | 0.05695 | 0.17388 | 0.0170 | 1 |
| 3-4 | 0.06701 | 0.17103 | 0.0173 | 1 |
| 4-5 | 0.01335 | 0.04211 | 0.0064 | 1 |
| 4-7 | 0 | 0.20912 | 0 | 1 |
| 4-9 | 0 | 0.55618 | 0 | 1 |
| 5-6 | 0 | 0.25202 | 0 | 1 |
| 6-11 | 0.09498 | 0.19890 | 0 | 1 |
| 6-12 | 0.12291 | 0.25581 | 0 | 1 |
| 6-13 | 0.06615 | 0.13027 | 0 | 1 |
| 7-8 | 0 | 0.17615 | 0 | 1 |
| 7-9 | 0 | 0.11001 | 0 | 1 |
| 9-10 | 0.03181 | 0.08450 | 0 | 1 |
| 9-14 | 0.12711 | 0.27038 | 0 | 1 |
| 10-11 | 0.08205 | 0.19207 | 0 | 1 |
| 12-13 | 0.22092 | 0.19988 | 0 | 1 |
| 13-14 | 0.17093 | 0.34802 | 0 | 1 |

* Impedance and line-charging susceptance in p.u. on a 100 MVA base. Line charging one-half of total charging of line.

**TABLE 5**
**Bus data / Operating conditions (IEEE - 14 bus system)**

| Starting bus voltage | | | Generation | | Load | |
|---|---|---|---|---|---|---|
| **Bus No.** | **Magnitude p.u.** | **Phase angle deg.** | **MW** | **MVAR** | **MW** | **MVAR** |
| 1* | 1.06 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1.00 | 0 | 40 | 0 | 21.7 | 12.7 |
| 3 | 1.00 | 0 | 0 | 0 | 94.2 | 19.0 |
| 4 | 1.00 | 0 | 0 | 0 | 47.8 | -3.9 |
| 5 | 1.00 | 0 | 0 | 0 | 7.6 | 1.6 |
| 6 | 1.00 | 0 | 0 | 0 | 11.2 | 7.5 |
| 7 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 9 | 1.00 | 0 | 0 | 0 | 29.5 | 16.6 |
| 10 | 1.00 | 0 | 0 | 0 | 9.0 | 5.8 |
| 11 | 1.00 | 0 | 0 | 0 | 3.5 | 1.8 |
| 12 | 1.00 | 0 | 0 | 0 | 6.1 | 1.6 |
| 13 | 1.00 | 0 | 0 | 0 | 13.5 | 5.8 |
| 14 | 1.00 | 0 | 0 | 0 | 14.9 | 5.0 |

* Slack Bus

**TABLE 6**
**Regulated Bus data (IEEE 14-Bus System)**

| Bus Number | Voltage Magnitude p.u. | Minimum MVAR capability | Maximum MVAR capability |
|---|---|---|---|
| 2 | 1.045 | -40 | 50 |
| 3 | 1.010 | 0 | 40 |
| 6 | 1.070 | -6 | 24 |
| 8 | 1.090 | -6 | 24 |

## Cost Characteristics

$C_1 = 50\, P_{g1}^2 + 245\, P_{g1} + 105$ $/ hr.

$C_2 = 50\, P_{g2}^2 + 351\, P_{g2} + 44.4$ $/ hr.

$C_6 = 50\, P_{g6}^2 + 389\, P_{g6} + 40.6$ $/ hr.

Total load demand of the system is $P_D = 259$ MW.

Maximum and minimum active power constraint on the generator bus for the given system is given below:

**TABLE 7**
**Active power constraints (IEEE 14 - Bus System) on the generator**

| Generator Number | $P_{gimin}$ | $P_{gimax}$ |
|---|---|---|
| $P_{g1}$ | 50 | 200 |
| $P_{g2}$ | 20 | 100 |
| $P_{g6}$ | 20 | 100 |

## Emission Characteristics

$E_1 = 135.5\, P_{g1}^2 - 126.5\, P_{g1} + 22.9$ $/hr.

$E_2 = 124.8\, P_{g2}^2 - 137.8\, P_{g2} + 137.3$ $/hr.

$E_6 = 80.5\, P_{g6}^2 - 76.7\, P_{g6} + 367.7$ $/hr.

## B-Coefficients in $MW^{-1}$ (IEEE 14 Bus system)

$B_{11} = 10^{-4} * 3.49$ $B_{12} = 10^{-4} * 0.68$ $B_{13} = 10^{-4} * (-0.39)$

$B_{21} = 10^{-4} * 0.68$ $B_{22} = 10^{-4} * 1.57$ $B_{23} = 10^{-4} * 0.15$

$B_{31} = 10^{-4} * 0.39$ $B_{32} = 10^{-4} * 0.15$ $B_{33} = 10^{-4} * 2.75$

### (3) IEEE 30 BUS SYSTEM

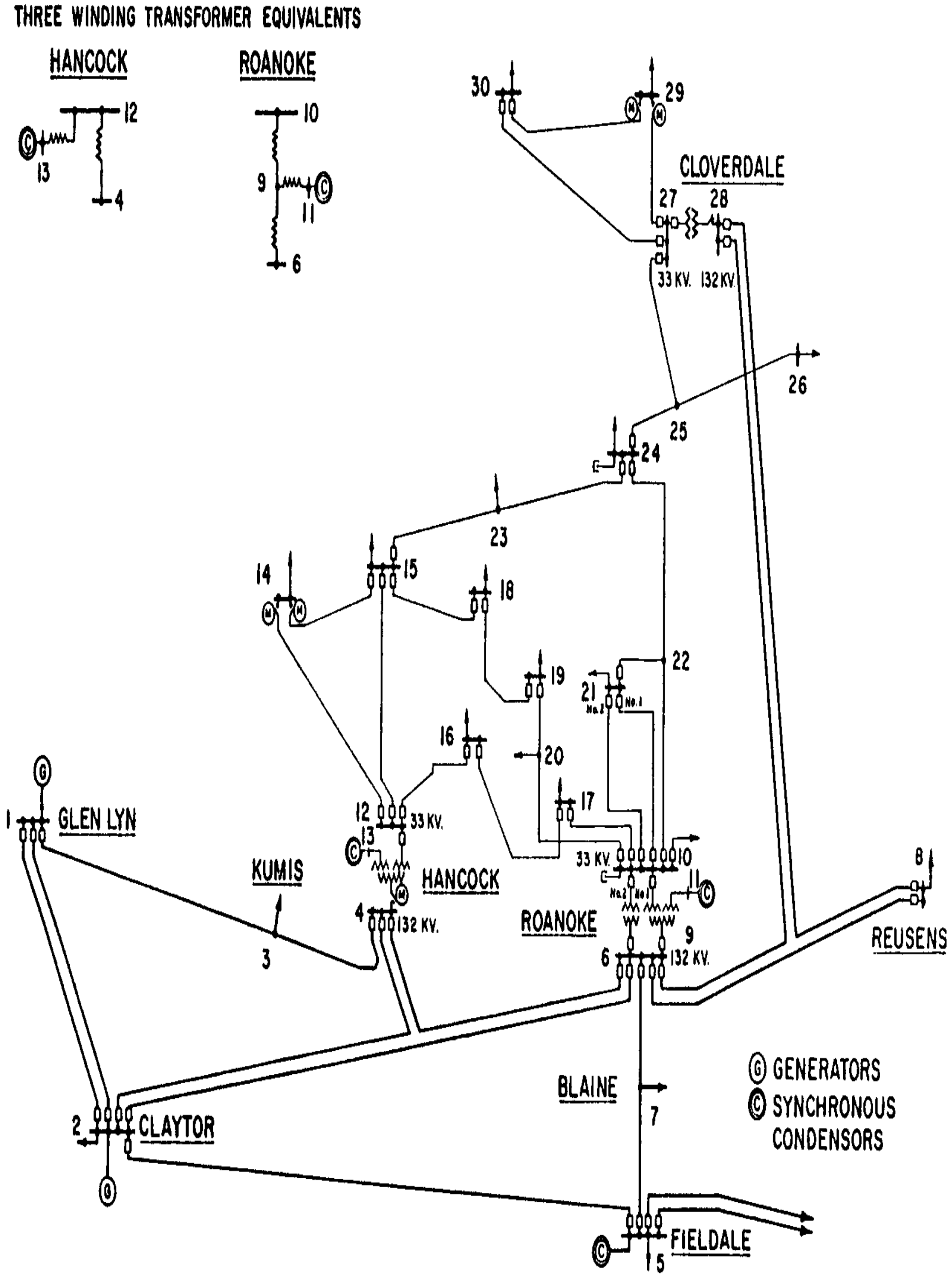



**Fig. 3** IEEE 30 Bus System

## TABLE 8
## Impedance and Line-charging Data (IEEE - 30 Bus System)

| Line Designation | Resistance p.u.* | Reactance p.u.* | Line Charging p.u.* | Tap Setting |
|---|---|---|---|---|
| 1-2 | 0.0192 | 0.0575 | 0.0264 | 1.00 |
| 1-3 | 0.0452 | 0.1852 | 0.0204 | 1.00 |
| 2-4 | 0.0570 | 0.1737 | 0.0184 | 1.00 |
| 3-4 | 0.0132 | 0.0379 | 0.0042 | 1.00 |
| 2-5 | 0.0472 | 0.1983 | 0.0209 | 1.00 |
| 2-6 | 0.0581 | 0.1763 | 0.0187 | 1.00 |
| 4-6 | 0.0119 | 0.0414 | 0.0045 | 1.00 |
| 5-7 | 0.0460 | 0.1160 | 0.0102 | 1.00 |
| 6-7 | 0.0267 | 0.0820 | 0.0085 | 1.00 |
| 6-8 | 0.0120 | 0.0420 | 0.0045 | 1.00 |
| 6-9 | 0 | 0.2080 | 0 | 0.978 |
| 6-10 | 0 | 0.5560 | 0 | 0.969 |
| 9-11 | 0 | 0.2080 | 0 | 1.00 |
| 9-10 | 0 | 0.1100 | 0 | 1.00 |
| 4-12 | 0 | 0.2560 | 0 | 0.932 |
| 12-13 | 0 | 0.1400 | 0 | 1.00 |
| 12-14 | 0.1231 | 0.2559 | 0 | 1.00 |
| 12-15 | 0.0662 | 0.1304 | 0 | 1.00 |
| 12-16 | 0.0945 | 0.1987 | 0 | 1.00 |
| 14-15 | 0.2210 | 0.1997 | 0 | 1.00 |
| 16-17 | 0.0824 | 0.1923 | 0 | 1.00 |
| 15-18 | 0.1070 | 0.2185 | 0 | 1.00 |
| 18-19 | 0.0639 | 0.1292 | 0 | 1.00 |
| 19-20 | 0.0340 | 0.0680 | 0 | 1.00 |
| 10-20 | 0.0936 | 0.2090 | 0 | 1.00 |
| 10-17 | 0.0324 | 0.0845 | 0 | 1.00 |
| 10-21 | 0.0348 | 0.0749 | 0 | 1.00 |
| 10-22 | 0.0727 | 0.1499 | 0 | 1.00 |
| 21-22 | 0.0116 | 0.0236 | 0 | 1.00 |
| 15-23 | 0.1000 | 0.2020 | 0 | 1.00 |
| 22-24 | 0.1150 | 0.1790 | 0 | 1.00 |
| 23-24 | 0.1320 | 0.2700 | 0 | 1.00 |
| 24-25 | 0.1885 | 0.3292 | 0 | 1.00 |
| 25-26 | 0.2544 | 0.3800 | 0 | 1.00 |
| 25-27 | 0.1093 | 0.2087 | 0 | 1.00 |
| 27-28 | 0 | 0.3960 | 0 | 0.968 |
| 27-29 | 0.2198 | 0.4153 | 0 | 1.00 |
| 27-30 | 0.3202 | 0.6027 | 0 | 1.00 |
| 29-30 | 0.2399 | 0.4533 | 0 | 1.00 |
| 8-28 | 0.0636 | 0.2000 | 0.0214 | 1.00 |
| 6-28 | 0.0169 | 0.0599 | 0.0065 | 1.00 |

Impedance and line-charging suceptance in p.u. on a 100 MVA base. Line charging one-half of total charging of line.

**TABLE 9**
**Bus data / Operating conditions (IEEE - 30 bus system)**

| | | | Generation | | Load | |
|---|---|---|---|---|---|---|
| **Bus No.** | **Magnitude p.u.** | **Phase Angle deg.** | **MW** | **MVAR** | **MW** | **MVAR** |
| 1* | 1.06 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1.00 | 0 | 40 | 0 | 21.7 | 12.7 |
| 3 | 1.00 | 0 | 0 | 0 | 2.4 | 1.2 |
| 4 | 1.00 | 0 | 0 | 0 | 7.6 | 1.6 |
| 5 | 1.00 | 0 | 0 | 0 | 94.2 | 19.0 |
| 6 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1.00 | 0 | 0 | 0 | 22.8 | 10.9 |
| 8 | 1.00 | 0 | 0 | 0 | 30.0 | 30.0 |
| 9 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 10 | 1.00 | 0 | 0 | 0 | 5.8 | 2.0 |
| 11 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 12 | 1.00 | 0 | 0 | 0 | 11.2 | 7.5 |
| 13 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 14 | 1.00 | 0 | 0 | 0 | 6.2 | 1.6 |
| 15 | 1.00 | 0 | 0 | 0 | 8.2 | 2.5 |
| 16 | 1.00 | 0 | 0 | 0 | 3.5 | 1.8 |
| 17 | 1.00 | 0 | 0 | 0 | 9.0 | 5.8 |
| 18 | 1.00 | 0 | 0 | 0 | 3.2 | 0.9 |
| 19 | 1.00 | 0 | 0 | 0 | 9.5 | 3.4 |
| 20 | 1.00 | 0 | 0 | 0 | 2.2 | 0.7 |
| 21 | 1.00 | 0 | 0 | 0 | 17.5 | 11.2 |
| 22 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 23 | 1.00 | 0 | 0 | 0 | 3.2 | 1.6 |
| 24 | 1.00 | 0 | 0 | 0 | 8.7 | 6.7 |
| 25 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 26 | 1.00 | 0 | 0 | 0 | 3.5 | 2.3 |
| 27 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 28 | 1.00 | 0 | 0 | 0 | 0 | 0 |
| 29 | 1.00 | 0 | 0 | 0 | 2.4 | 0.9 |
| 30 | 1.00 | 0 | 0 | 0 | 10.6 | 1.9 |

* Slack Bus

**TABLE 10**
**Regulated Bus Data (IEEE - 30 Bus System)**

| Bus Number | Voltage Magnitude p.u. | Minimum MVAR Capability | Maximum MVAR Capability |
|---|---|---|---|
| 2 | 1.045 | -40 | 50 |
| 5 | 1.01 | -40 | 40 |
| 8 | 1.01 | -10 | 40 |
| 11 | 1.082 | -6 | 24 |
| 13 | 1.071 | -6 | 24 |

## Cost Characteristics

$C_1 = 50\ P_1^2 + 245\ P_1 + 105$ $/hr.

$C_2 = 50\ P_2^2 + 351\ P_2 + 44.4$ $/hr.

$C_8 = 50\ P_8^2 + 389\ P_8 + 40.6$ $/hr.

Maximum and minimum active power constraint on the generator bus for the given system is given below:

**TABLE 11**
**Active power constraints (IEEE 30-Bus system) on the generator**

| Generator Number | $P_{gimin}$ | $P_{gimax}$ |
|---|---|---|
| $P_{g1}$ | 50 | 250 |
| $P_{g2}$ | 30 | 100 |
| $P_{g8}$ | 30 | 100 |

## Emission Characteristics

$E_1 = 135.5\ P_{g1}^2 - 126.5\ P_{g1} + 22.9$ $/hr.

$E_2 = 124.8\ P_{g2}^2 - 137.8\ P_{g2} + 137.3$ $/hr.

$E_8 = 80.5\ P_{g8}^2 - 76.7\ P_{g8} + 367.7$ $/hr.

## B-Coefficients in $MW^{-1}$ (IEEE 30 Bus System)

$B_{11} = 0.0307$ $B_{12} = 0.0129$ $B_{13} = 0.0002$

$B_{21} = 0.0129$ $B_{22} = 0.0152$ $B_{23} = -0.0011$

$B_{31} = 0.0002$ $B_{32} = -0.0011$ $B_{33} = 0.0190$

## References


[1] J. L. Cohan, D. H. Marks, “A review and evaluation of multiobjective programming techniques,” Water Resource Research, Vol.11, pp.208-220, (1975).

[2] J. H. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press (1975).

[3] Starr M., Zeleny M., “Multiobjective criteria decision making,” North Holland publication. Amsterdam, (1977).

[4] Gerald W. Evans, “An overview of techniques for solving multiobjective mathematical programs,” Management Science, 30 (11), pp.1268-1282, (1984).

[5] A. J. Wood, B. F. Wollenberg, “Power generation, operation and control,” John Wiley & Sons, New York, (1984).

[6] Po Lung Yu P.L., “Multiple criteria decision making concepts, techniques and extrusions,” (Plenum Press, New York and London), (1985).

[7] C. L. Wadhwa, N. K. Jain, “Multiple objective optimal load flow: a new perspective,” IEE Proc. C, 137(1), pp.11-18, (1990).

[8] Elliot R. Lieberman, “Soviet multiobjective mathematical programming method: An overview,” Management Science, 37(9), pp.1147-1165, (1991).

[9] Mori Hiroyuki, Horiguchi Takuya, “Genetic algorithm based approach to economic load dispatching,” IEEE Transactions on Power Systems, Vol.1, pp.145-150, (1993).

[10] J. Kennedy, R. Eberhart, “Particle swarm optimization,” Proceedings of the IEEE International Joint Conference on Neural Networks. IEEE Press, pp.1942-1948, (1995).

[11] Russell Eberhart, James Kennedy, “A new optimizer using particle swarm theory,” Proceedings of Sixth International Symposium on Micro Machine and Human Science, pp. 39-43, (1995).

[12] Ahmed Farag, Samir AL-Baiyat, T. C. Cheng, “Economic load dispatch multiobjective optimization procedures using linear programming techniques,” IEEE Transactions on Power Systems, 10(2), pp.731-738, (1995).

[13] Park June Ho, Yang Sung Oh, Lee Hwa Seok, Young Moon Park, “Economic load dispatch using evolutionary algorithms,” IEEE Transactions on Power Systems, Vol. 2, pp. 441- 445, (1996).

[14] K.F Man, K.S Tang, S. Kwong, “Genetic algorithms: concept and applications in engineering design,” IEEE Transaction on Industrial Electronics, 43(5), pp.519 - 534, (1996).

[15] U. Nangia, N.K. Jain, C.L. Wadhwa, “Investigations of multiobjective optimal load flow study by sequential goal programming,” J. IE, 77, pp.99–103 (1996).

[16] J. X. Xu, C. S. Chang, X. W. Wang, "Constrained multiobjective global optimization of longitudinal interconnected power system by genetic algorithm," IEE Proc. - Generation, Transmission. Distribution,143(5), pp.435-446, (1996).

[17] U. Nangia, N.K. Jain, C.L. Wadhwa, "Surrogate worth tradeoff technique for multiobjective optimal power flows," IEE Proc. Generation Transmission Distribution, 144(6), pp.547–553, (1997).

[18] U. Nangia, N.K. Jain, C.L. Wadhwa, "Noninferior set estimation for multiobjective optimal load flow study," J. IE, 78, pp.229–235 (1998).

[19] Yuhui Shi, Russell Eberhart, "A modified particle swarm optimizer," Proceedings IEEE Congress on Evolutionary Computation, pp. 69-73, (1998).

[20] J. H. Park, S. O. Yang, K. J. Mun, H. S. Lee, J. W. Jung, "An application of evolutionary computations to economic load dispatch with piecewise quadratic cost functions," IEEE Transactions on Power Systems, Vol. 69, pp. 289-294, (1998).

[21] U. Nangia, N.K. Jain, C.L. Wadhwa, "Optimal weight assessment based on range of objectives in multiobjective optimal load flow study," IEE Proc. C 145(1), pp.65–69 (1998).

[22] Chira Achayuthakan, S. C. Srivastava, "A genetic algorithm based economic load dispatch solution for Eastern region of EGAT system having combined cycle and cogeneration plants," IEEE Transactions on Power Systems, Vol. 1, pp. 165-170, (1998).

[23] Giridhar Kumaran, V. S. R. K. Mouly, "Using evolutionary computation to solve the economic load dispatch problem," IEEE Transactions on Power Systems, Vol. 3, pp. 296-301, (2001).

[24] Ying Y. Hong, Chih Yuan Li, "Genetic algorithms based economic dispatch for cogeneration units considering multiplant multibuyer wheeling," IEEE Transactions on Power Systems,17(1), pp.134-140, (2002).

[25] M. A. Abido, "A novel multiobjective evolutionary algorithm for environmental/ economic power dispatch," Electric Power System Research 65(1), pp.71-81, (2002).

[26] K. E. Parsopoulos, M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," Journal of Natural Computing, Vol.1, pp.235-306, (2002).

[27] K.E. Parsopoulos, M.N. Vrahatis, "Particle swarm optimization method in multiobjective problems," Symposium on Applied Computing (SAC 2002), Madrid, Spain, pp.603-607, (2002).

[28] M.A Abido, "Environment/economic power dispatch using multiobjective evolutionary algorithms," IEEE Transactions on Power Systems 18(4), pp.1529-1537, (2003).

[29] U.Nangia, N.K. Jain, C.L. Wadhwa, "Comprehensive comparison of various multiobjective techniques," Eng. Intell. Syst. J. 11(3), pp.123–132 (2003).

[30] I. J. Nagrath, D.P. Kothari, “Modern Power System Analysis,” Tata McGraw, (2003).

[31] S. H. Ling, H. K. Lam, F.H.F. Leung, Y. S. Lee, “Improved genetic algorithm for economic load dispatch with valve- point loadings,” IEEE Transactions on Power Systems, Vol.1, pp. 442 - 447, (2003).

[32] Shin-Der Chen, Jiann - Fuh Chen, “A direct Newton–Raphson economic emission dispatch,” International journal of Electrical Power and Energy System, 25(5), pp. 411- 417, (2003).

[33] Nidul Sinha, R. Chakrabarti, P. K. Chattopadhyay, “Improved fast evolutionary program for economic load dispatch with non-smooth cost curves,” IE (I) Journal. EL Vol. 85, pp.110-114, (2004).

[34] T. Aruldoss, Albert Victorie, A Ebenezer Jeyakumar, “Hybrid PSO-SQP for economic dispatch with valve-point effect,” Electric Power System Research, 71(1), pp. 51-59, (2004).

[35] Zwe – Lee Gaing, “A particle swarm optimization approach for optimum design of PID controller in AVR system,” IEEE Transactions on Energy Conversion, 19(2), pp. 384-391, (2004).

[36] W. B. Langdon, Riccardo Poli, Chirstopher R Stephens, “Kernel methods for PSOs,” Technical Report, Department of Computer Science, University of Essex, UK, 20, pp.1-7, December (2005).

[37] M. S. Voss, “Principal component particle swarm optimization (PCPSO),” Proceedings of IEEE Swarm Intelligence Symposium, pp.1-4, (2005).0-7803-8916-6/05/$20.00 © (2005).

[38] Jaco F. Schutte, Albert A. Groenwold, “A study of global optimization using particle swarms,” Journal of Global Optimization, Springer, 31(1), pp. 93-108, (2005).

[39] Ouiddir R., Rahli M., Korid L. Abdelhakem, “Economic dispatch using genetic algorithm: application to western Algeria’s Electrical Power Network,” Journal of Information Science and Engineering, 21(3), pp. 659 – 668, (2005).

[40] Chao - Lung Chiang, “Improved genetic algorithm for power economic dispatch of units with valve point effects and multiple fuels,” IEEE Transactions on Power Systems, 20(4), pp.1690-1699, (2005).

[41] Boonserm Kaewkamnerdpong, Peter J. Bentley, “Perceptive particle swarm optimization: an investigation,” Department of Computer Science University College, London UK, Proceedings 2005, IEEE Swarm Intelligence Symposium, (2005).

[42] Chunming Yang, Dan Simon, “A new particle swarm optimization technique,” Proceedings of 18th International Conference on Systems Engineering (ISCEng’05) (2005).

[43] Younes Mimoun, Rahli Mostafa, “Economic power dispatch using the combination of two genetic algorithms,” Journal of Electrical and Electronics Engineering, 69(2), pp. 175-181, (2006).

[44] Younes Mimoun, Mostefa Rahli, Koridak Lahouri Abdelhakem, "Economic power dispatch using Evolutionary Algorithm," Journal of Electrical Engineering, 57(4), pp. 211-217, (2006).

[45] Y.S. Brar, J. S. Dhillon, D.P. Kothari, "Multiobjective load dispatch based on Genetic - Fuzzy Technique," IEEE PES Power Systems Conference and Exposition, pp. 931-937, (2006).

[46] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE Transactions on Evolutionary Computation, 10(3), pp.281-295, (2006).

[47] M.A Abido, "Multiobjective evolutionary algorithms for electric power dispatch problem," IEEE Transactions evolutionary computation, 10(3), pp. 315–329, (2006).

[48] Ajith Abraham, He Guo, Hongbo Liu, "Swarm intelligence: foundations, perspectives and applications," Studies in Computational Intelligence (SCI) 26, pp. 1-25, (2006).

[49] A. Ismael F. Vaz, L.N. Vicente, "A particle swarm pattern search method for bound constrained global optimization," Journal of Global optimization, Springer, 39(2), pp.197-219, (2007).

[50] Yan Jiang, Tisong Hu, Chong Chao Huang, Xianing Wu, "An improved particle swarm optimization algorithm," Applied Mathematics and Computation, Elsevier, 193(1), pp.231-239, (2007).

[51] Tao Gong, Andrew L. Tuson, "Particle swarm optimization for quadratic assignment problems–a forma analysis approach," International journal of computational Intelligence Research Vol.2, No. X, pp.1-9, (2007).

[52] Lingfeng Wang, Chanan Singh, "Environmental / Economic power dispatch using fuzzified multi-objective particle swarm optimization algorithm," Electrical Power Systems Research, 77(12), pp.1654-1664, (2007).

[53] Ying- Ping Cen, Wen - Chih Peng, "Particle swarm optimization with recombination and dynamic linkage discovery," IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, 37(6), pp.1460-70, (2007).

[54] Chiang C. L., "Genetic based algorithm for power economic load dispatch," IET Generation Transmission Distribution, 1(2), pp.261-269, (2007).

[55] Wei-Bing Liu, Xian - Jia Wang, "An evolutionary game based particle swarm optimization algorithm," Journal of computational and Applied Mathematics, 214, pp.30-35, (2008).

[56] Wei Zu, Yan - ling Hao, Hai -tao Zeng, Wen jing Tang, "Enhancing the particle swarm optimization based on equilibrium of distribution," IEEE 2008 Chinese Control and Decision Conference (CCDC 2008), pp. 285-289, (2008).

[57] Yueting Chen, Fang He, "Research on particle swarm optimization in location assignment optimization," Proceedings of the 7th World Congress on Intelligent Control and Automation, Chonggqing, China, pp.111-116, (2008).

[58] J.S. AI - Sumait, J. K. Suklski, A.K. AI - Othmamn, "Solution of different types of economic load dispatch problems using a pattern search method," Electrical Power Components and Systems, Taylor Francis, 36(3), pp.250-265, (2008).

[59] N. Noman, H. Iba, "Differential evolution for economic load dispatch problems," Electric Power Systems Research, 78(8), pp.1322–1331, (2008).

[60] I. Musirin, N. Ismail, M. Kalil, M. Idris, T. Rahman, M. Adzman, "Ant colony optimization (ACO) technique in economic power dispatch problems," Proc. of the International Multi Conference of Engineers and Computer Scientists, Hong Kong, Vol. II, IMECS (2008).

[61] G. Baskar, M.R. Mohan, "Security constrained economic load dispatch using improved particle swarm optimization suitable for utility system,'' Electrical Power and Energy Systems, 30, pp.609–613, (2008).

[62] Lahouari Abdelhakem Koridak, Mostefa Rahli, Mimoun Younes, "Hybrid optimization of the emission and economic dispatch by the genetic algorithm," Leonardo Journal of Sciences, Issue 14, pp.193-203, (2008).

[63] Chen - Chien Hsu, Chun - Hwui Gao, "Particle swarm optimization incorporating simplex search and center particle for global optimization," IEEE Conference on Soft Computing in Industrial Applications (SMCIA/08), Muroran, Japan, pp. 26-31, (2008).

[64] Yutaka Maeda, Naota Matsushita, "Empirical study of simultaneous perturbation particle swarm optimization", SICE Annual Conference, pp.2545-2548, (2008).

[65] Junqi Zhang Zhongmin Xiao, Ying Tan, Xingui He, "Hybrid particle swarm optimizer with advance and retreat strategy and clonal mechanism for global numerical optimization", in Proc. of IEEE Congress on Evolutionary Computation (CEC 2008), pp. 2059-2066, (2008).

[66] Zhi - Xiang Hou, "Wiener model identification based on adaptive particle swarm optimization," in Proceedings of the seventh IEEE International Conference on Machine Learning and Cybernetics, Kunming, pp. 1041-1045, (2008).

[67] Qi Luo, Jun - yong Liu, Chuan Long "An improved particle swarm optimization algorithm," IEEE International Conference on Granular Computing (GRC), Hangzhou, China, (2008).

[68] Li Jian, Liu Zhiming, Chen Peng, "Solving constraint optimization via dual particle swarm optimization with stochastic ranking," 2008 International Conference on Computer Science and Software Engineering, pp.1215-1218, (2008).

[69] Jiao Wei, Liu Guangbin, Liu Dong, "Elite particle swarm optimization with mutation," Proc. of IEEE 7th Intl. Conf. on System Simulation and Scientific Computing, pp. 800 - 803, (2008).

[70] Zhixiong Liu, Runjun Zhao, "Equipment possession quantity modelling and particle swarm optimization," Proc. of Third IEEE International Conference on Genetic Evolutionary Computing, pp.628-632, (2009).

[71] Youxin Luo, Xiaoyi Che, "Chaos immune particle swarm optimization algorithm with hybrid discrete variables and its application to mechanical optimization," Third IEEE International Symposium on Intelligent Information Technology Application Workshops, (2009).

[72] Md. Sakhawat Hossen, Fazle Rabbi, Md. Mainur Rahman, "Adaptive particle swarm optimization (APSO) for multimodal function optimization," International Journal of Engineering and Technology, 1(3), pp.98-103, (2009).

[73] Keisuke Kameyama, "Particle swarm optimization–a survey," The Institute of Electronics Information and Communication Engineers, Vol. E92-D, pp. 1354-1361, (2009).

[74] C.A. Floudas, C.E. Gounaris, "A review of recent advances in global optimization," Journal of Global Optimization Vol. 45, pp.3-38, (2009).

[75] He Jing, She Dejia, Wang Li, "A quadratic particle swarm optimization for weight optimization," Third IEEE International Symposium on Intelligent Information Technology Application", 2009.

[76] Praveen Ranjan Srivastava, Tai-hoon Kim, "Application of genetic algorithm in software testing," International Journal of Software Engineering and its Applications, 3(4), pp.87- 96, (2009).

[77] Saoussen Brini, Hsan Hadj Abdallah, Abderrazak Ouali, "Economic Dispatch for Power System included Wind and Solar Thermal," Leonardo Journal of Sciences, Issue14, pp.204-220, (2009).

[78] M. Fesanghary, M. M. Ardehali, "A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem," Energy, 34(6), pp. 757–766, (2009).

[79] Rui Li, Yirong Guo, Yujuan Xing, Ming Li, "A novel multi-swarm particle swarm optimization algorithm applied in active contour model," Proc. IEEE Global Congress on Intelligent Systems, pp.139-143, (2009).

[80] Xu Jun, Huiyou Chang, "The discrete binary version of the improved particle swarm optimization algorithm," International Conference on Management and Service Science, MASS, pp.1-6, (2009).

[81] Jun Tang, Xiaojuan Zhao, "A fine tuning hybrid particle swarm optimization algorithm," International Conference on Future Bio-Medical Information Engineering, (2009).

[82] Wei Wang, Jin - Mu Wu, Jie - Hua Liu, "A particle swarm optimization based on chaotic neighborhood search to avoid premature convergence," Third IEEE International Conference on Genetic and Evolutionary

Computing, (2009).

[83] Adel M. Sharaf, Adel A. A. Ei - Gammal, "A novel discrete multi-objective particle swarm optimization (MOPSO) of optimal shunt power filter," Power Systems Conference and Exposition, PSCE, IEEE PES, pp.1-7, (2009).

[84] C. K. Goh, K.C. Tan, D.S. Liu, S. C. Chaim, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," European Journal of Operation Research, Vol. 202, pp.42-54, (2010).

[85] Bilal Benmessahel, Mohamed Touahria, "An improved combinatorial particle swarm optimization algorithm to database vertical partition," Journal of Emerging Trends in Computing and Information Sciences, 2(3), pp.130-135, (2010).

[86] Zhiyu You, Weirong Chen, Xiaoqiang Nan, "Adaptive weight particle swarm optimization algorithm with constriction factor," Proceedings of International Conference of Information Science and Management Engineering, pp. 245-248, (2010).

[87] Zhi Li, Yong Chen, "Design and implementation for parallel particle swarm optimization color quantization algorithm," International Conference on Computer and Information Application, ICCIA 2010.

[88] S. Hemamalini, Sishaj P. Simon, "Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions," Electric Power Components and Systems, 38(7), pp. 786–803, (2010).

[89] A. Bhattacharya, P. K. Chattopadhyay, "Biogeography-based optimization for different economic load dispatch problems," IEEE Transactions on Power Systems, 25(2), pp. 1064–1077, (2010).

[90] Huanhuan Ji, Jing Jie, Junji Li, Ying Tan, "A bi-swarm particle swarm optimization with cooperative co-evolution," IEEE International Conference on Computational Aspects of Social Networks, CASoN (2010), pp.323-326, (2010). DOI:10.1109/CASoN.2010.79.

[91] Jun-qing Li, Sheng-Xie, Bao-xian Jia, Yu-Ting Wang, "A Hybrid particle swarm optimization and Tabu search algorithm for flexible job – shop scheduling problem," International Journal of Computer Theory and Engineering, 2(2), pp.1793-8201, (2010).

[92] C.L. Wadhwa, "Economic load dispatch," Electrical power system, 6th ed. New Age International, pp. 628, (2010).

[93] D.P. Kothari, J.S. Dhillon, "Power system optimization," PHI, Second Edition, (2010).

[94] Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin, Kwang Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," IEEE Transactions on Power Systems, 25(1), pp.156-166, (2010).

[95] Na Li, Song Zhu, "Modified particle swarm optimization and its application in multimodal function optimization," International Conference on

Transportation, Mechanical, and Electrical Engineering (TMEE), Changchun, China, (2011).

[96] Nai - Jen Li, Wen-June Wang, Chen - Chien Hsu, Chih - Min Lin, "Global optimization using novel randomly adapting particle swarm optimization approach," IEEE International conference on Systems, Man and Cybernetics, pp. 1783-1787, (2011), DOI:10.1109/ICSMC.2011.6083930

[97] Changming Ji, Fang Liu, Xinming Zhang, "Particle swarm optimization based on catfish effect for flood optimal operation of reservoir," Proc. Seventh IEEE International Conference on Natural Computation, pp.1197-1201, (2011).

[98] Weidong Ji, Keqi Wang, "An improved particle swarm optimization algorithm," International Conference on Computer Science and Network Technology, pp. 585-589, (2011).

[99] Qun Niu, Xiaohai Wang, Zhuo Zhoua, "An efficient cultural particle swarm optimization for economic load dispatch with valve-point effect," Elsevier, Procedia Engineering, Vol.23, pp.828-834, (2011).

[100] V. P. Ravi kumar, B. K. Panigrahi, A. Mohapatra, M. K. Mallick, "Economic load dispatch solution by improved harmony search with wavelet mutation," International Journal of Computational Sciences and Engineering, 6(2), pp. 122–131, (2011).

[101] Z.-H. Zhan, J. Zhang, Y. Li, Y.-H. Shi, "Orthogonal learning particle swarm optimization," IEEE Transactions on Evolutionary Computation", 15(6), pp.832-847, (2011).

[102] Ehsan Shahamatnia, Mohamad Mehdi Ebadzadeh, "Application of particle swarm optimization and snake model hybrid on medical imaging," IEEE Third International Workshop on Computational Intelligence in Medical Imaging (CIMI), 2011.Pages 8, DOI:10.1109/CIMI.2011.5952043.

[103] Serkan Kiranyaz, Jenni Pulkkinen, Moncef Gabbouj, "Multi-dimensional particle swarm optimization in dynamic environments," Expert Systems with Application, Elsevier, 38(3), pp. 2212-2223, (2011).

[104] S. Chandrashekhar Reddy, P.V.N. Prasad, A Jaya Laxmi, "Power quality and reliability improvement of distribution system by optimal number, location and size of DGs using particle swarm optimization," IEEE 7$^{th}$ International Conference on Industrial and Information, ICIIS, 978-1-4673-2605-6/12, (2012) IEEE.

[105] Steffen Freitag, Rafi L. Muhanna, Wolfgang Graf, "A particle swarm optimization approach for training artificial neural networks with uncertain data," Proc. 5th International Conference on Reliable Engineering Computing (REC 2012), pp.151-170, (2012).

[106] Ismail Ibrahim, Zulkifli Md. Yusof, Sophan, Hamzah Ahmad, Zuwairie Ibrahim, "A novel multi-state particle swarm optimization for discrete combinatorial optimization problems," Fourth IEEE International Conference on Computational Intelligence, Modelling and Simulation, pp.18-23, (2012)

[107] Z. Cui, Z. Shi, J. Abonyi et al., “Economic load dispatch using population-variance harmony search algorithm,” Transactions of the Institute of Measurement and Control, 34(6), pp. 746–754, (2012).

[108] M. R. Lohokare, B. K. Panigrahi, S. S. Pattnaik, S. Devi, A. Mohapatra, “Neighborhood search-driven accelerated biogeography-based optimization for optimal load dispatch,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42(5), pp. 641–652, (2012).

[109] S. Kanoongo, P. Jain, “Biogeography based optimization for different economic load dispatch problem with different migration models,” Proceedings of the IEEE Students' Conference on Electrical, Electronics and Computer Science: Innovation for Humanity (SCEECS '12), pp. 1–4, (2012).

[110] K. Sudhakara Reddy, M. Damodar Reddy, “Economic load dispatch using firefly algorithm,” International Journal of Engineering Research and Application (IJERA),2(4), pp.2325-2330, (2012).

[111] Lily Chopra, Raghuwinder Kaur “Economic load dispatch using simple and refined genetic algorithm,” International Journal of Advances in Engineering and Technology, ©IJAET, 5(1), pp.584-590, (2012).

[112] C Li, S. Yang, T. T. Nguyen, “A self-learning particle swarm optimizer for global optimization problems,” IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 42(3), pp. 627-646, (2012).

[113] N.K. Jain, Uma Nangia, Jyoti Jain, “GA based multiobjective economic load dispatch by maximization of minimum relative attainments,” IEEE 5th India International Conference on Power Electronics (IICPE), (2012).

[114] Saibal K. Pal, C. S. Rai, “Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems,” International Journal of Intelligent Systems and Applications, Vol. 10, pp. 50-57, (2012).

[115] J.J. Jamian, M.W. Mustafa, H. Mokhlis, M.N. Abdullah, “Comparative study on distributed generator sizing using three types of particle swarm optimization,” Third IEEE International Conference on Intelligent Systems Modelling and Simulation,pp.131-136,(2012).

[116] Kyle Robert Harrison, “GP - based adaptable evolutionary particle swarm optimization,” Lecture Notes by Department of Computer Science, Brock University, (2012).

[117] Yan Zhe-ping, Deng Chao, Zhou Jia-jia, Chi Dong-nan, “A novel two-subpopulation particle swarm optimization,” Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China, pp. 4113-4117, (2012).

[118] Nikhil Padhye, Kalyanmoy Deb, Pulkit Mittal, “Boundary handling approaches in particle swarm optimization,” Kan GAL Report Number 2012014, pp.1-13, (2012).

[119] Lu Baiquan, Gao Gaiqin, Lu Zeyu, “The Block diagram method for

designing the particle swarm optimization," Journal of Global Optimization, Vol.52, pp.689–710, (2012), DOI 10.1007/s10898-011-9699-9

[120] Hardiansyah, Junaidi, Yohannes MS, "Solving economic load dispatch problem using particle swarm optimization technique," International Journal of Intelligent Systems and Application, 4(12), pp.12-18, (2012).

[121] Masashi Yamamoto, Takeshi Uchitane, Toshiharu Hatanaka, "An experimental study for multi–objective optimization by particle swarm with graph based archive," SICE Annual Conference, Akita University, Japan, pp.89-94, (2012).

[122] Kyle Robert Harrison, Beatrice Ombuki-Berman, Andries P. Engelbrecht, "Knowledge transfer strategies for vector evaluated particle swarm optimization," Technical Report Brock University, September (2012).

[123] N. Mishra, M. Pandit, "Environmental/economic power dispatch problem using particle swarm optimization," International Journal of Electronics and Computer Science Engineering, 2(2), pp.512-519, (2012).

[124] Zhicheng Qu, Qin Yang, "Improved particle swarm optimization for constrained optimization," International Journal of Education and Management Engineering, 2(2), pp.21-28, (2012).

[125] Ming Chen, Tao Wang, Jian Feng, Yong-Yong Tang, Li-Xin Zhao, "A hybrid particle swarm optimization improved by mutative scale chaos algorithm," Fourth International Conference on Computational and Information Sciences,2012, pp. 321-324, (2012).

[126] Liu Jin-yue, Zhu Bao-Ling, "The Application of particle swarm optimization algorithm in the extremum optimization of nonlinear function," 12th IEEE International Conference on Computer and Information Technology, pp.286-289, (2012).

[127] Abd Allah A. Mousa, Mohamed A. El – Shorbagy, "Enhanced particle swarm optimization based local search for reactive power compensation problem," Scientific Research, Applied Mathematics, Vol.3, pp.1276-1284, (2012).

[128] Andries Engelbrecht, "Particle swarm optimization with discrete crossover," IEEE Congress on Evolutionary Computation, Cancun, Mexico, (2013).

[129] Damian Obioma Dike, Moses Izuchukwu Adinfono, George Ogu, "Economic dispatch of generated power using modified lambda iteration method," IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), 7(1), pp.49-55, (2013).

[130] Ashish Dhamanda, Arunesh Dutt, Surya Prakash, A.K. Bhardwaj, "A traditional approach to solve economic load dispatch problem of thermal generating unit using MATLAB programming," IJERT 2(9), pp.3147-3152, (2013).

[131] Jaya Sharma, Anita Mahor "Particle swarm optimization approach for economic load dispatch: a review," International Journal of Engineering and Research and Application. Vol. 3, pp.13-22, (2013).

[132] N. Phanthuna, J. Jaturacherdchaiskul, S. Lerdvanitrip, S. Auchariyamet, "Application of particle swarm optimization technique for economic load dispatch and loss reduction," International Journal of Electrical, Electronics Science and Engineering,7(10), pp.1303-1307, (2013).

[133] Hardiansyah, "A modified particle swarm optimization technique for economics load dispatch with valve-point effect," International Journal of Intelligent systems and Applications, Vol.7, pp.32-41, (2013).

[134] Zahra Beheshti, Siti Mariyam Shamsuddin, Siti Yuhaniz, "Binary accelerated particle swarm algorithm (BAPSA) for discrete optimization problems," Journal of Global Optimization, Vol. 57, pp. 549–573, (2013).

[135] Luis Migue, Rios Nikolaos, V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," Journal of Global Optimization, Vol. 56, pp. 1247–1293, (2013).

[136] Zhimin Chen, Yuming Bo, Panlong Wu, Weijun Zhou, "A new particle filter based on organizational adjustment particle swarm optimization," Appl. Math. Inf. Sci. 7(1), pp.179-186, (2013).

[137] Arasomwan Akugbe Martins, Adewumi Aderemi Oluyinka, "An adaptive velocity particle swarm optimization for high-dimensional function optimization," Congress on Evolutionary Computation, pp.2352-2359, (2013).

[138] G. Xiong, D. Shi, X. Duan, "Multi-strategy ensemble biogeography-based optimization for economic dispatch problems," Applied Energy, Vol. 111, pp. 801–811, (2013).

[139] S. Ozyon, D. Aydin, "Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones," Energy Conversion and Management, Vol. 65, pp. 397–407, (2013).

[140] Sunny Orike, David Corne, "Constrained elitist genetic algorithm for economic load dispatch: case study on Nigerian power system," International Journal of Computer Applications (0975 – 8887),76(5), (2013).

[141] Bishnu Sahu, Avipsa Lall, Soumya Das, T Manoj Patra, "Economic load dispatch in power system using genetic algorithm," International Journal of Computer Applications, 67(7), pp. 17-22, (2013).

[142] Bharat Bhushan, Sarath S. Pillai, "Particle swarm optimization and firefly algorithm: performance analysis," Third IEEE International Advance Computing Conference (IACC), (2013), pp.746-751, 978-1-4673-4529-3/12 IEEE.

[143] Singiresu S. Rao, "Engineering Optimization Theory and Practice (Third Enlarged Edition)," New Age International Publishers ISBN:978-81-224-2723-3. (2013).

[144] Jun Sun Xiao-Jun Wu, Wei Fang, Zhenyu Wang, "Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization," IEEE Transactions on Industrial Informatics, 10(1), (2014).

[145] Gajavelli. S.S. Bhishma Rao, “Optimization Techniques,” SCITECH, (2014).

[146] G. Xiong, Y. Li, J. Chen, D. Shi, X. Duan, “Polyphyletic migration operator and orthogonal learning aided biogeography-based optimization for dynamic economic dispatch with valve-point effects,” Energy Conversion and Management, Vol. 80, pp. 457–468, (2014).

[147] M. Basu, “Teaching-learning-based optimization algorithm for multi-area economic dispatch,” Energy, Vol. 68, pp. 21–28, (2014).

[148] M. Basu, “Improved differential evolution for economic dispatch,” International Journal of Electrical Power & Energy Systems, Vol. 63, pp. 855–861, (2014).

[149] Q. Xu., Y. LI., X. Yang “Economic load dispatch using bacterial foraging optimization algorithm based on evolution Strategies,” Advanced Material Research, Vol. 860-863, pp.2040-2045, (2014).

[150] Arun sandhu, Ombeer saini, Shalini, “Steepest descent method for economic load dispatch using MATLAB,” Journal of Engineering Computers & Applied Sciences (JECAS), 4(6), pp.168-175, (2015).

[151] T. T. Nguyen, D. N. Vo, “The application of one rank cuckoo search algorithm for solving economic load dispatch problems,” Applied Soft Computing, Vol. 37, pp. 763–773, (2015).

[152] Nishant Saxena , Souvik Gangulib, “Solar and Wind Power Estimation and Economic Load Dispatch Using Firefly Algorithm,” Procedia Computer Science, 4th International Conference on Eco-friendly Computing and Communication Systems, ICECCS Vol.70, pp.688-700, (2015).

[153] L. Haldurai, V. Vinodhini, “Parallel indexing on color and texture feature extraction using r-tree for content based image retrieval,” International Journal of Computer Sciences and Engineering, 3(11), pp.11-15, (2015).

[154] S. K. Sreedharamurthy, H. R. Sudarshana Reddy, “Feature subset selection using genetic algorithms for hand written Kannada alphabets recognition,” International Journal of Computer Sciences and Engineering, 3(6), pp.94-99, (2015).

[155] M.N. Abdullah, A. Halim A. Bakar, N. A. Rahim, H. Mokhlis, “Modified particle swarm optimization for economic-emission load dispatch of power system operation,” Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 23, pp 2304 – 2318, (2015).

[156] N. Mishra, M. Pandit, “Environmental/economic power dispatch problem using particle swarm optimization,” Inte. J. of Electronics and Computer Science Engineering, IJECSE, 2(2), pp. 512-519.

[157] N. Singh, Y. Kumar, “Multiobjective economic load dispatch problem solved by new PSO,” Advances in Electrical Engineering, 2015, Article ID 536040, pp.1-6, (2015).

[158] B. Hadji, B. Mahdad, K. Srairi, N. Mancer, “Multi-objective PSO-TVAC for environmental / economic dispatch problem,” Science Direct Energy Procedia, 74(2015), pp.102-111, (2015).

[159] S. Phulambrikar, “Solving combined economic emission dispatch solution using Jaya optimization algorithm approach,” International Research Journal of Engineering and Technology (IRJET), 3(11), pp.501-512, (2016).

[160] N. K. Jain, U. Nangia, A. Jain, “PSO for multiobjective economic load dispatch (MELD) for minimizing generation cost and transmission losses,” Journal of Institution of Engineers Ser. B, 97(2), pp.185-191, (2016).

[161] Ravi Shankar, Hemant Mahala, “Multiobjective economic load dispatch using new PSO,” International Journal of Engineering Science and Computing, 6(3), pp. 2250-2255, (2016).

[162] Provas Kumar Roy, Tandra Pal, “Grey wolf optimization applied to economic load dispatch problems,” International Journal of Electrical Power & Energy Systems, Vol. 83, pp. 325-334, (2016).

[163] R. Rao, P. Srinivas, M. Divakar, G. Venkatesh, “Artificial bee colony optimization for multi objective economic load dispatch of a modern power system,” International Conference on Electrical and Electronics and optimization (ICEEOT), (2016).

[164] Mohammed Azmi Al-Betar, Mohhammed A. Awadallah, Ahamad Tajudin Khader, Asaju Laaro Bolaji, “Tournament-based harmony search algorithm for non-convex economic load dispatch problem,” Applied Soft Computing, Vol. 47, pp. 449–459, (2016).

[165] W.T. Elsayed, Y. G. Hegazy, F.M. Bendary, M.S. El-bages, “Modified social spider algorithm for solving the economic dispatch Problem,”, International Journal of Engineering Science and Technology, Vol.19, pp.1672–1681, (2016).

[166] Ly Huu Pham, Thang Nguyen, Dieu Ngoc Vo, Cuong Tran, “Adaptive cuckoo search algorithm based method for economic load dispatch with multiple fuel options and valve point effect,” International Journal of Hybrid Information Technology,1(19), pp. 41-50, (2016).

[167] Vijay kumar, Jagdev Singh, Yaduvir Singh, Sanjay Sood, “Optimal economic load dispatch using fuzzy logic & genetic algorithms,” International Journal of Computer Engineering & Technology, 17(1), pp. 62-77, (2016).

[168] A.Y. Abdelaziz, E. S. Ali, S. M. Abd Elazim, “Flower pollination algorithm to solve combined economic and emission dispatch problems,” International Journal of Engineering Science and Technology, 19(2), pp. 980–990, (2016).

[169] N. K. Jain, Uma Nangia, Jyoti Jain, “A review of particle swarm optimization,” Journal of the Institution of Engineers (India): Series B, Springer, 99(4), pp.407-411, (2016).

[170] N. K. Jain, U. Nangia, Jyoti Jain, “Effect of population and bit size on optimization of function by Genetic Algorithm,” International Conference on Computing for Sustainable Global Development, Delhi, pp.194-199, (2016).

[171] N. K. Jain, Uma Nangia, Jyoti Jain, "An improved PSO based on initial selection of particles," IEEE first International Conference on Power Electronics, Intelligent Control and Energy, Delhi, pp.1-5, (2016).

[172] N. K. Jain, Uma Nangia, Jyoti Jain, "An improved PSO based on initial selection of particles (ISBPSO) for economic load dispatch," IEEE first International Conference on Power Electronics, Intelligent Control and Energy, Delhi, pp.1-5, (2016).

[173] Bratati Ghosh, Ajay Kumar Chakraborty, Arup Ratan Bhowmik, Aniruddha Bhattacharya, "Krill Herd algorithm solution for the economic emission load dispatch in power system operations," 7th International Conference on Power Systems (ICPS) College of Engineering Pune, India. (2017).

[174] Akshay Rana, Shavet Sharma, Amandeep, "Stochastic economic load dispatch using Pareto distribution with multiple fuels local convex by bacterial foraging optimization," International Journal of Advance Research, Ideas and Innovations in Technology, 3(2), pp.317-321, (2017).

[175] D. Poornima, Sishaj P. Simon, B. Sonia, T. Sunita, "Multiobjective economic load dispatch problem using A-loss coefficients," International Journal of Pure and Applied Mathematics, 114(8), pp.143–153, (2017)

[176] N.K. Jain, Uma Nangia, Iqbal Singh, "Economic load dispatch in 3-D space by genetic algorithm," Journal of Institution of Engineers, Ser. B, 98(5), pp.495-501, (2017).

[177] Bahman Taheri, G Holamreza Aghajani, Mahsa Sedaghat, "Economic dispatch in a power system considering environmental pollution using a multi-objective particle swarm optimization algorithm based on the Pareto criterion and fuzzy logic" Int. J. Energy Environ Eng., 8(2), pp.99–107, (2017).

[178] A. Man-Im, W. Ongsakul, J. Govind Singh, C. Boonchuay, "Multi-objective economic dispatch considering wind power penetration using stochastic weight trade-off chaotic NSPSO," J. Electr. Power Compon. Syst., 45(14), pp.1525–1542, (2017).

[179] Sk Md Ali Bulbul, Moumita Pradhan, Provas Kumar Roy, Tandra Pal, "Opposition- based krill herd algorithm applied to economic load dispatch problem," Ain Shams Engineering Journal, 9(3), pp.423-440, (2018).

[180] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, A. L. Bolaji, A. Almomani, "Economic load dispatch problems with valve-point loading using natural updated harmony search," Neural Computing and Applications, 29(10), pp. 767–781, (2018).

[181] M. Amiri, S. Khanmohammadi, M. A. Badamchizadeh, "Floating search space: A new idea for efficient solving the economic and emission dispatch problem," Energy, Vol. 158, pp.564-579, (2018).

[182] M. Mohammadian, A. Lorestani, M.M. Ardehali, "Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm," Science Direct, Vol. 161, 710–724, (2018).

[183] M. Poornima, C. Kuyin, M. Revathy, “Brain tumor detection using genetic algorithm,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 3(1), pp.1101-1105, (2018).

[184] Pabitra Mohan Das, Asini kumar Balirsingh, Sangram Keshori Mohapatra, “Economic load dispatch using moderate random search PSO with ramp rate limit constraints,” Proceedings of IEEE International Conference on Technologies for Smart-City Energy Security and Power (ICSESP), Bhubaneswar, India, pp.1-5, (2018).

[185] P. Jiang, J. L. Ding, Y Guo., “Application and dynamic simulation of improved genetic algorithm in production workshop scheduling,” Int. Simul. Model 17, pp.159-169, (2018).

[186] N. K. Jain, Uma Nangia, Jyoti Jain, “Economic load dispatch using adaptive social acceleration constant based particle swarm optimization,” Journal of The Institution of Engineers (India): Series B, Springer, 99(5), pp. 431- 439, October (2018).

[187] N. K. Jain, Uma Nangia, Jyoti Jain, “Multiobjective economic load dispatch using fuzzy logic decision system,” IEEE Second International Conference on Power Electronics, Intelligent Control and Energy, Delhi, (2018).

[188] N. K. Jain, Uma Nangia, Jyoti Jain, “Impacts of PSO Parameters on its Convergence,” IEEE second International Conference on Power Electronics, Intelligent Control and Energy, Delhi, pp.1-5, (22-24 October, 2018).

[189] X.Yu, X. Yu, Y. Lu, J. Sheng, “Economic and emission dispatch using ensemble multiobjective differential evolution algorithm,” Sustainability, 10(2), 418 (2018).

[190] Diljinder Singh, J. S. Dhillon,“Ameliorated grey wolf optimization for economic load dispatch problem,” Energy, Vol. 169, pp.398-419, (2019).

[191] Lulu wen, Kaile Zhou, Shanlin Yang, Xinhui Lu, “Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting,” Energy, Vol. 171, pp.1053-1065, (2019).

[192] Jian Lin, Zhou-Jing Wang, “Multi-area economic dispatch using an improved stochastic fractal search algorithm,” Energy, Vol. 166, pp.47-58, (2019).

[193] Guanghui yuan, Weixin Yang, “Study on optimization of economic dispatching of electric power system based on hybrid intelligent algorithms (PSO and AFSA),” Energy, Vol. 183, pp.926-935, (2019).

[194] N. K. Jain, Uma Nangia, Jyoti Jain, “Multiobjective economic load dispatch studies in 2-D and 3-D space by particle swarm optimization technique,” Journal of Institution of Engineers India: Series B, Springer, 100(3), pp. 237-247, (2019).

## AUTHOR'S RESEARCH PUBLICATIONS

### List of papers published in Peer Reviewed Referred Journals:

[1] N. K. Jain, Uma Nangia, **Jyoti Jain**, "A review of particle swarm optimization", Journal of Institution of Engineers (India): Series B, Springer, 99(4), pp.407-411, (2016).

[2] N. K. Jain, Uma Nangia, **Jyoti Jain**, "Economic load dispatch using adaptive social acceleration constant based particle swarm optimization", Journal of Institution of Engineers (India): Series B, Springer, 99(5), pp. 431- 439, (2018).

[3] N. K. Jain, Uma Nangia, **Jyoti Jain**, Multiobjective economic load dispatch studies in 2-D and 3-D space by particle swarm optimization technique", Journal of Institution of Engineers India: Series B, Springer, 100(3), pp. 237-247, (2019).

## List of Papers Published in International / National Conferences:

[1] N. K. Jain, U. Nangia, **Jyoti Jain**, "GA based Multi-objective economic load dispatch by maximization of minimum relative attainments", IEEE 5th India International Conference on Power Electronics (IICPE 2012), Delhi, (December 2012).

[2] N. K. Jain, U. Nangia, **Jyoti Jain**,"Effect of population and bit size on optimization of function by Genetic Algorithm", International Conference on Computing for Sustainable Global Development, Delhi, pp.194-199 (16-18 March, 2016) .

[3] N. K. Jain, Uma Nangia, **Jyoti Jain**, "An improved PSO based on initial selection of particles", IEEE first International Conference on Power Electronics, Intelligent Control and Energy, Delhi. (4-6 July, 2016).

[4] N. K. Jain, Uma Nangia, **Jyoti Jain,** "An improved PSO based on initial selection of particles (ISBPSO) for economic load dispatch" IEEE first International Conference on Power Electronics, Intelligent Control and Energy, Delhi. (4-6 July, 2016).

[5] N. K. Jain, Uma Nangia, **Jyoti Jain,** "Multiobjective economic load dispatch using fuzzy logic decision system" IEEE second International Conference on Power Electronics, Intelligent Control and Energy, Delhi, (22-24 October, 2018).

[6] N. K. Jain, Uma Nangia, **Jyoti Jain,** "Impacts of PSO Parameters on its Convergence" IEEE second International Conference on Power Electronics, Intelligent Control and Energy, Delhi, (22-24 October, 2018).

## List of communicated papers:

[1] N. K. Jain, Uma Nangia, Jyoti Jain, “Feasibility oriented PSO algorithm to determine Pareto - Optimal front in a partial run for multiobjective economic load dispatch considering cost of generation and system transmission losses.” Journal of Institution of Engineers India: Series B, Springer.

[2] N. K. Jain, Uma Nangia, Jyoti Jain, “Feasibility oriented PSO algorithm to determine Pareto - Optimal front in a partial run for multiobjective economic load dispatch considering cost of generation and system transmission losses and environmental emission”. Applied Soft Computing, Elsevier.

[3] N. K. Jain, Uma Nangia, Jyoti Jain, “Split phase economic load dispatch algorithm”, Journal of Institution of Engineers (India) : Series B, Springer.