LOAD FLOW

USING

PARTICLE SWARM OPTIMIZATION

DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

POWER SYSTEM

(Electrical Engineering)

Submitted by:

GANESH KUMAR PANDEY

(2K13/PSY/06)

Under the supervision of

Prof. N K Jain & Prof. Uma Nangia

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

2015

CERTIFICATE

I, GANESH KUMAR PANDEY, Roll No. 2K13/PSY/06 student of M. Tech. (Power System), hereby declare that the dissertation titled “LOAD FLOW USING PARTICLE SWARM OPTIMIZATION” under the supervision of Prof. N.K. Jain & Prof. Uma Nangia of Electrical Engineering Department, Delhi Technological University in partial fulfillment of the requirement for the award of the degree of Master of Technology has not been submitted elsewhere for the award of any Degree.

Place: Delhi
Date:

GANESH KUMAR PANDEY
2K13/PSY/06
M.Tech (PS)

Prof. N.K. Jain
Professor
EE Dept., DTU

Prof. Uma Nangia
Professor
EE Dept., DTU

ACKNOWLEDGEMENT

Any project is a team work, successful completion of which is possible only through cooperation and sincere efforts of the team mates. This is of great pleasure for me to avail the opportunity of expressing my gratefulness to all those who were of immense help to me in completing this Major Project.

Firstly I want to show my sincere and profound thanks to **Prof. N K JAIN (project guide) & Prof. UMA NANGIA,** Who are the constant source of inspiration throughout the development of this Major Project and provided me with all the requisite guidance and information from time to time .I also thank Almighty for the entire blessings he has bestowed upon me. Last but not least I express my sincere thanks to my family members, colleagues, friends, college staff and all those who have contributed directly and indirectly through suggestions , thoughts, support, and presence for the completion of the Major Project ,they have remained a source of encouragement and inspiration for me.

GANESH KUAMR PANDEY

2K13/PSY/06

ABSTRACT

Load flow (LF) is an important tool in the planning and operation of power systems. It is usually solved using conventional numerical techniques like Newton-Raphson (NR) and Gauss-Siedel (GS). Most of these techniques depend on getting the inverse of the Jacobian matrix of the system. Such techniques fail to solve the load flow in some conditions, like Heavy loaded system, Ill-conditioned Jacobian matrix.

In this thesis an application of particle swarm optimization (PSO) in solving the load flow problem as an optimization problem is discussed.

A MATLAB program has been developed for PARTICLE SWARM OPTIMIZATION to solve load flow problem. Program is tested on IEEE 5, 14 and 30 bus system and the results are compared with conventional method.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PSO	Particle swarm optimization
NR	Newton-Raphson
LF	Load flow
GS	Gauss-Seidel

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Power demand is increasing day by day which needs for the requirement of availability of power all the time, for which proper planning and operation of power system is necessary. Exchange of power between different systems has become very important. Power system operation-centers run automatically to perform the function like AGC (automatic generation control), state estimation and load flow studies etc.

Almost all the functions performed in power system depends on the load flow study in steady state condition, which involves the calculation of bus voltages, load angles, power generation in generator bus and line power flow. As the load flow problem is non-linear and can be solved by numerical technique. Different methods are available like Gauss-Siedel. Generally newton-Raphson method is applied as it has good convergence characteristics however there are certain issues like complexity of calculations and Jacobian matrix inverse calculation which depends on initial estimated values.

Load flow is very important in power system planning and operation. If conventional method is applied then the convergence of the problem depends on the proper selection of slack bus and initial assumed voltages at load bus. Therefore initial estimation is very difficult and problem dependent require some new technique which are free from these shortcomings.

Several new techniques are there to solve equations of any type, like linear, nonlinear constrained and without constrained. Among these particle swarm optimization (PSO) is a very powerful technique and has less number of parameters. Therefore it has wide range of applications. Unlike genetic algorithm PSO does not has operators like crossover and mutation. Position of the particle in search space is the solution of the optimization problem.

In this work first load flow problem is taken, and optimization function is formulated taken into consideration of real power mismatch, reactive power mismatch and voltage mismatch.

Hence a load flow problem is treated as an optimization problem and PSO is applied to solve those. Algorithm is implemented on IEEE 5, 14 and 30 bus systems. Results are compared with those of obtained from conventional methods.

Our objectives are given below:

- Implementing PSO and programing the implemented code in MATAB 2013a
- Determining the Parameter dependency in PSO which means how to choose parameter in PSO for better and faster result.
- Application of PSO in load flow of IEEE 5, 14 and 30 bus systems.
- Comparison of results with Newton-Raphson method.

1.2 OBJECTIVE AND METHODOLOGY

Our main aim in this thesis is to solve the load flow problem for IEEE 5, 14 and 30 bus systems using PSO and compare the results with that of obtained from conventional method like newton Raphson.

Work has been carried out in following plan:

1 Develop the MATLAB code to perform Particle swarm optimization technique.

2 Application of PSO to optimize Rosenbrock function to determine the parameter dependency in PSO

3 Load flow problem formulation for IEEE 5, 14 and 30 bus system.

4 Comparison of results with Newton-Raphson method.

1.3 LITERATURE SURVEY

An optimization problem consists of minimizing or maximizing a given real function by taking input from a given allowed set and computing the value of the function. There are different optimization problems, single variable function, multivariable functions without constraints, multivariable functions with both equality and inequality constraints. Different optimization techniques are available like linear programming, nonlinear programming, geometric programming, dynamic programming, evolutionary algorithms, genetic algorithm, particle swarm optimization etc.

Load flow study is basically the problem of determining the quantities i.e. bus voltage, load angle, satisfying the inequality and equality constraints on real and reactive power injection at each bus. Therefore, the main objective of load flow is to find angles and the voltage magnitudes at different system buses, and match power demand to generation.

In power system load forecasting is done to meet the load demand, and match the generation with load to maintain constant frequency. As power system consists of thousands of buses most of them are load bus and few are generator bus, load are in the form of real and reactive power, load flow calculation computes voltage and load angle at each bus and real power generated at generator bus. Different conventional methods are there, but have certain limitations like Jacobian matrix inversion problem. Other methods based on artificial intelligence have been proposed to solve load flow such as particle swarm optimization, genetic algorithm.

Kennedy J. and R. Eberhart [1] in 1995 introduced a new technique of optimization of nonlinear function and that is Particle Swarm Optimization (PSO). Several benchmark functions are tested on PSO and application involving nonlinear function optimization is proposed.

Amgad A, EL-Dib, Hosam K.M. Youssef [2] presented the load flow using hybrid particle swarm optimization (HPSO).They formulated the load flow problem as an optimization problem and applied it on IEEE 6 and 14 bus system.

Yamille Del Valle [3] discussed the basics of particle swarm optimization and its parameters i.e. variants and application of PSO in power system. In power system we come across solving of many equations which are nonlinear in nature and has some constraints. In this paper PSO which is a very reliable optimization technique is used. Comprehensive survey on application of PSO in power system is done. For applying PSO in each application technical details are discussed.

P.G.Rizwankhan, R.Mageshvaran, I. Jacob Raglend, Sudheera V.Yuvaraj, and T.Vijayakumar [4] implemented nontraditional technique like CPSO, HDE, and PSO in solving optimal load flow problem. These optimization techniques have the capability to give global optimum point. Ward hale six bus system and IEEE14 bus system is tested. Solution obtained is acceptable and useful in solving optimal load flow. MATLAB software is used for implementation of algorithm.

M. R. AlRashidi [5] did the survey for application of PSO in power system. They presented the application of particle swarm optimization in the area of power system. Highlights on the PSO and its features and various advantages in comparison to other methods. Further they had discussed the recent trends in the area of PSO application in power system and its theoretical studies.

Prof. Daozhuo Jiang, Ibrahim Oumarou, and Prof. Cao Yijia [6] presented the paper on OPF (optimal power flow) by a simple PSO algorithm. As the method is dynamic it overcomes several shortcomings of conventional method like premature convergence, complexity and provides high quality solution. The main objective is to keep the voltage, reactive power generation in generator bus and transformer tap setting within their permissible limits. MATPOWER is used to test the effectiveness of PSO method. IEEE 30 bus System is taken to prove the superiority and effectiveness of the method.

Chuan Long, Lin Lu, Qi Luo, Jun-yong Liu [7] developed improved particle swarm optimization technique and that is HSPSO (hierarchical structure poly-particle swarm optimization). They observed that in the bottom layer calculations are performed using poly-particle swarms which enlarges the search domain, best position found by all the particles in the bottom layer is regarded as the best position of the particle in top layer. Optimization result of the top layer particle is fed back to bottom layer swarm. The particle velocity is updated if some particle trend to local maxima. In such way velocity is updated. The method is implemented on four functions and the result had been compared.

Camila P. Salomon [8] proposed a method to solve load flow via particle swarm optimization. The methodology depends on minimization of mismatch at each bus. Power flow is important in planning and operation of power system. Particle swarm optimization is applied to solve the new computational model. This model develops better convergence as compared to Newton Raphson method. Method is tested on IEEE 6 bus system.

A.Arunya Revathi and Dr.N.S.Marimuthu [9] presented a paper based on evolutionary computation approach to solve load flow using modified particle swarm optimization. Their method proposed a velocity updating to eliminate local minima. The objective is to minimize voltage mismatch, power mismatch, at each bus. The method is tested on ward hale 6 and 9 bus system. The results obtained show the improvement in solution.

C.Kumar, Dr. Ch. Padmanabha Raju [10] presented method to solve optimal load flow including constraints using PSO method. The goal of the paper is to discuss the feasibility of PSO method of solving different objective function. OPF is the problem of satisfying the constraints like voltage reactive power, Real power limit at each bus. Proposed PSO method had developed and the result obtained is compared with the evolutionary programming method by testing the algorithm on IEEE 14 bus system.

Carlos Henrique Valerio de Moraes, Camila Paes Salomon, Maurilio Pereira Coutinho Germano Lambert-Torres and Luiz Eduardo Borges da Silva [11] presented method to solve optimal load flow in power system. Proposed method includes the minimization of power mismatch at buses and applied HPSO mutation operation to achieve the goal. Two different power systems are taken to prove the method.

Bhupender Sharma, Shivani Sehgal and Ajay Nain [12] observed some parameters which affect the optimal load flow by particle swarm optimization and genetic algorithm. OPF provides economic information and variables which minimizes the cost of power system operation. Programming work was carried out using MATLB both in genetic algorithm and particle swarm optimization. IEEE 30 bus was taken for case study upon comparison it was found that both the method were almost comparable to exact value. Result shows the effectiveness of algorithm and applicable to optimum power flow under constraints too.

Deepak Saini and Neeraj Saini [13] studied the load flow using PSO. They did the load flow in different loading conditions using PSO for computation of load angle and voltage at each bus. STATCOM is placed optimally to improve the voltage profile.

1.4 PLAN OF THE THESIS

This thesis is arranged in 6 chapters. The contents are explained as indicated below:

Chapter 1: This chapter plans the research aim of the thesis and introduces the PARTICLE SWARM OPTIMIZATION and load flow. Literature survey of the related topic is also covered here.

Chapter 2: Introduction of particle swarm optimization

Chapter 3: Application of PSO to mathematical benchmark function

Chapter 4: Load flow problem formulation

Chapter 5: Load flow using particle swarm optimization for IEEE 5, 14 and 30 buses and Results are compared with Newton-Raphson method.

Chapter 6: Conclusion and future scope of work is discussed.

CHAPTER 2

PARTICLE SWARM OPTIMIZATION

2.1 INTRODUCTION

Particle swarm optimization (PSO) technique is a population based search method developed by Kennedy and Eberhart in 1995. It is based on natural behavior of swarm movement in search space for food. PSO has application in optimizing linear, nonlinear, multi-dimensional problem and single dimensional problems.

PSO technique performs search algorithm with population of particles. Dimension of the problem is decided by how many variables are there in the problem, Initial value is assigned to particles in search space. Particle moves in search space with some initial assumed velocity and adjusts its position according to its global best position and personal best position. Velocity component is kept on updating in every iteration, proper assumption of initial value is required inertia weight is other important component Particle finally arrives at the best position in whole search space by tracking the two positions i.e. personal best position and global best position.

PSO algorithm has parameters like velocity component, inertia component, retardation factor etc. on which the algorithm depends. Choosing the best parameter value is very difficult task. In searching of global optimum point requires good exploration in search space which depends on parameters of PSO like velocity, acceleration, inertia weight of the particles. If parameters values are not assigned in proper way algorithm may trap to local optimum point or may even not converge properly.

At time t, the i^{th} particle $X_i(t)$ can be described as

$$X_i(t) = [X_{i1}(t), X_{i2}(t), \dots . X_{in}(t)]$$

$$V_i(t) = [V_{i1}(t), V_{i2}(t), \dots . V_{in}(t)]$$

2.2 Basic PSO Model and its parameters

2.2.1 Basic PSO model

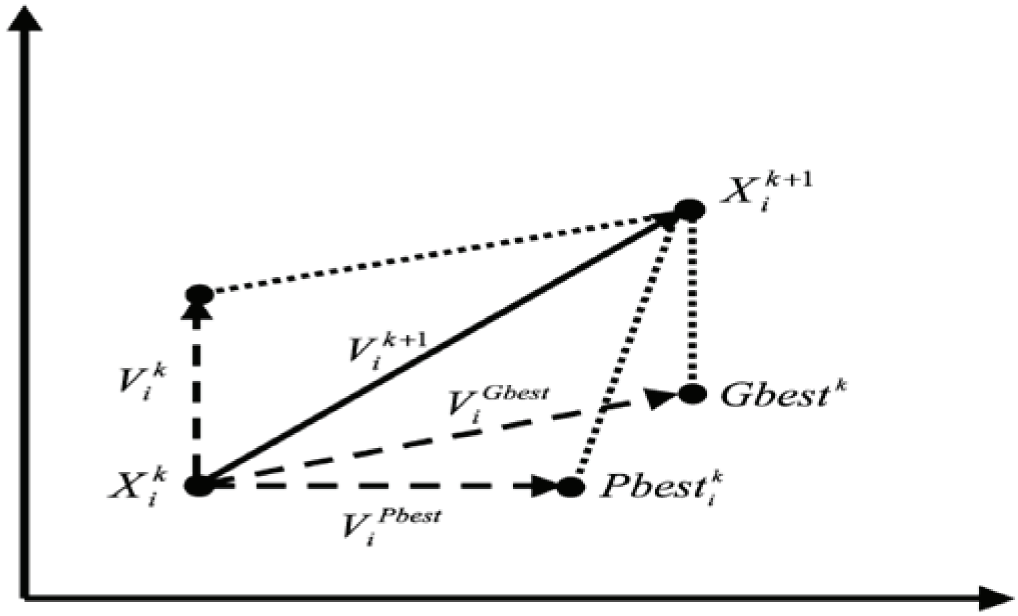

Fig. 1 The search mechanism of the particle swarm optimization.

Where

X_s And V_s optimized parameters

$X_{ik}(t)$ Position of the i^{th} particle with respect to the k^{th} iteration

$V_{ik}(t)$ Velocity of the i^{th} particle with respect to the k^{th} iteration

The particle updates its velocity and positions according to the following equations.

$$v_i^{k+1} = w * v_i^k + c_p * rand() * \left(p_{best_i} - p_i^k\right) + c_g * rand() * (g_{best_i} - g_i^k) \quad (1)$$

$$X_i^{k+1} = X_i^k + v_i^{k+1} \quad (2)$$

Where

c_p Self-confidence range and

c_g Known as the swarm range.

v_i^k Velocity of the i^{th} particle at iteration 'k'

k Iteration number

w Inertia constant

p_{best_i} Value of personal best position of each particle in (i)th iteration.

g_{best_i} Value of personal best position of each particle in (i)th iteration.

$rand()$ Random number between 0 and 1

2.2.2 PSO PARAMETERS

2.2.2.1 Swarm size

Number of particles in the population decides the swarm size. Number of particles affects the PSO performance. Very few particles swift the algorithm for getting trapped in local optimum point. While very large number of particles tends the algorithm to slow down. No one rule is there to select the number of particles but generally dimension of the problem increases, swarm size should be increased.

2.2.2.2 Iteration number

In particle swarm optimization we have to set the maximum number of iteration count, which is problem dependent. Less number of iteration tends to premature stopping of algorithm while very large number of iteration increases the complexity of algorithm and also time of computation i.e. it slows down the convergence.

2.2.2.3 Velocity Components

Velocity component updates the swarm position, and particles move with randomness following its own best position and global best position. Initially some random velocity is assigned to all the particles, in each iteration velocity is updated.

2.2.2.4 Acceleration coefficients

Acceleration coefficients affect the trajectory path of the particles based on the particle's own experience and other particles experience. This decides an important parameter in PSO. In order to efficiently keep track of global optimum and convergence of the solution proper setting of acceleration factor is a must.

2.2.2.5 INERTIA WEIGHT

Inertia Weight in PSO is very important as it decides the exploration in search space and convergence. There are many types of inertia weights available. Linearly decreasing inertia weight can improve the performance. In many papers other inertia weights are experimented like random decreasing, logarithmic etc. Different inertia weights are tested in respect of no of iteration, convergence characteristic, complexity of algorithm and result gives the comparative study.

2.4 APPLICATION OF PARTICLE SWARM OPTIMIZATION:

In this chapter we had discussed about different areas of application of PSO. At first Ebehart and kennedy implemented the PSO for practical application, they used PSO in combination with neural network and developed the combined algorithm. Now PSO is used to solve multidimensional problem with constraints. Now a days PSO is used in the area of telecommunication, power system, data mining, electromagnetics, and many other field also.

Different applications of PSO are listed in the below table:

TABLE: 1 DIFFERENT APPLICATIONS OF PSO

Electromagnetics	Temperature control based on FPGA microwave filter, high speed CMOS Design, RF optimization and design, designing frequency selective absorber and surface, measurement of flicker in voltage, circuit analysis, Fuel cell, parallel processor arrays.

Power system	Automatic generation control, Optimal power dispatch, load flow problem ,load flow including FACTS devices, Reactive power optimization, STATCOM placement in power system to minimize transmission loss, economic load dispatch, Photovoltaic system control
Biomedical	Drug design, human movement Biomechanics optimization, prediction of protein structure, electroencephalograph, biometrics, classification of cancer, DNA Detection, survival prediction.
Networking	Radar network, TCP network control mobile communication optimum equipment placement, wavelength division multiplexed network, WDM telecommunication network, wireless network, delayed and group broadcast, voltage regulation, cellular neural network, design of radial basis function network, product unit network, wireless sensor network design, wireless video sensor

	Neural network training based on feed forward, neuron controller, neural gas network
Robotics	Robotic arm control, robot running, playing soccer, swarm robotics, navigation of vehicle, mapping of Environment.
Modeling and design	Conceptual design, induction heating, RF circuit analysis, wideband amplifier design based on CMOS, friction model, model order reduction, thermal system identification, worst case electronics design, logic circuit design, transmission line design, design of filters
Optimization	Optimizing internal combustion engine, minimum spanning tree, path optimization, urban planning, FPGA placement, nuclear electric population system,
Forecasting and prediction	Load forecasting, prediction of surface roughness, urban traffic flow forecasting, stream flow forecast, battery charge estimation,

2.4 ADVANTAGES AND LIMITITATIONS OF PSO:

PSO is very powerful search algorithm technique, but it has some Limitations. The advantages and limitations are discussed below:

2.4.1 Advantages of PSO technique

1. PSO Algorithm is derivative free.
2. PSO has Very simple calculation. As very few equations are there in PSO to solve simple program can be implemented.
3. PSO algorithm has very less Computation as it has to update the velocity of the particle and update the position of the particle which is the solution of the optimization problem
4. It has very less complexity and simple logic is developed based on the personal best position and global best position.
5. PSO algorithm has very few parameter like velocity, acceleration, acceleration factor, retardation factor, therefore implementation of algorithm is very easy
6. Algorithm can be implemented easily for any engineering application. Like power system, electromagnetics, robotics etc.
7. Particle swarm optimization is not dependent on initial set of values.

2.4.2 Limitations of PSO technique:

1. PARTICLE-SWARM-OPTIMIZATION (PSO) algorithm may suffer from trapping into local optimum point which may lead to inaccuracy and degrading in speed.

2.3 COMPUTATIONAL PROCEDURE

BASIC PSO ALGORITHM:

1) Initialize number of particle
2) For each particle
   I. Initialize position and velocity in search space
   II. FOR i = 1 to maximum number of iteration
3) DO
   a) For each particle
      (i) Compute fitness value
      (ii) Check if the fitness value is best among all the previous values then assign it as P_{best}

      END
   b) For each particle
      (i) Find the particle with the best fitness value that value is assigned as g_{best}
      (ii) Update the swarm velocity according to velocity equation
      (iii) Apply velocity constraints
      (iv) Position is also updated according to position equation
      (v) Apply position constraint

      END
4) If maximum iteration limit crosses or error is less than the set value for error then

   Break

   END

CHAPTER 3

APPLICATION OF PSO TO MATHEMATICAL BENCHMARK TEST FUNCTION

3.1 BENCHMARK FUNCTION

Here Rosenbrock test function is used:

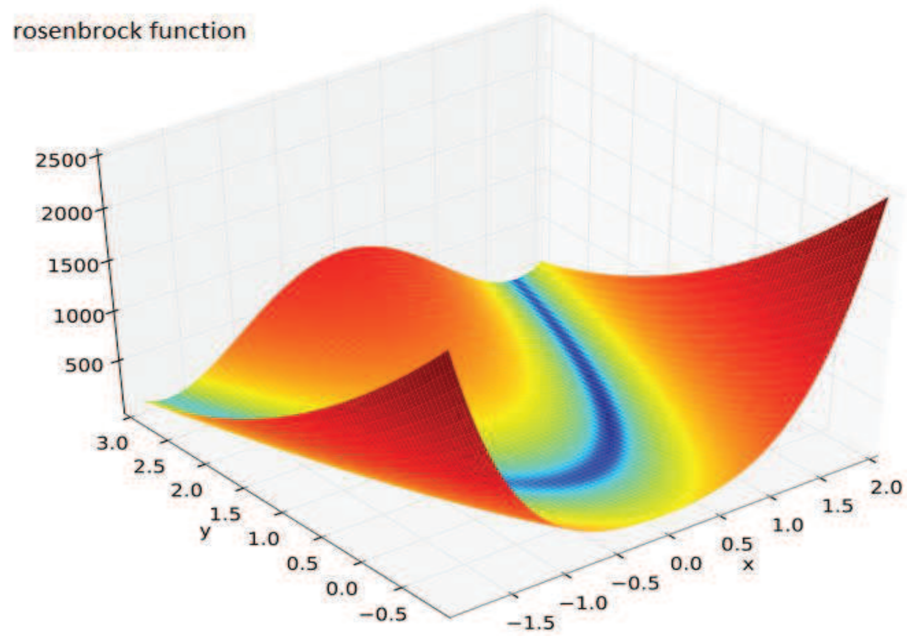

Fig 2 : Rosenbrock function

Rosenbrock function is given by:

$$f(x, y) = (1 - x)^2 + 100 * (y - x^2)^2 \quad (3)$$

Which has global minima at f(x,y)=(1,1)

MATLAB PROGRAME for Rosenbrock function is implemented with different case studies:

CASE 1: To select best inertia weight.

Case 2: Dependency of retardation factor with $(C_p + C_g)$

CASE 3: How to select $C_p and\ C_g$

3.5 DIFFERENT INERTIA WEIGHT

TABLE 2: DIFFERENT INERTIA WEIGHT

S.NO	NAME OF INERTIA WEIGHT	FORMULA
1	Constant Inertia Weight	w=c c=0.7(considered for experiments)
2	Random Inertia Weight	$0.5 + \frac{rand()}{2}$
3	Linear Decreasing Inertia Weight	$w_{max} - \frac{w_{max} - w_{min}}{it} * i$
4	Logarithm Decreasing Inertia Weight	$w_{max} + (w_{min-}\ w_{max}) * log10(a + \frac{10*i}{it})$

To obtain how different parameters selection depends on one another, one test function is taken. And keeping some parameter fixed other parameters are varied and the results are recorded in a table, which gives the idea of parameter selection

Here Rosenbrock test function is used which is given by:

$$f(x, y) = (1 - x)^2 + 100 * (y - x^2)^2$$

Following case studies had done in Rosenbrock test function, MATLAB code is run and the results are recorded in tabular form:

CASE 1: To select best inertia function

CASE 2: Dependency of retardation factor with $(C_p + C_g)$

CASE 3: How to select $C_p and\ C_g$

CASE 1: To select best inertia function

Here Rosenbrock test function is used which is given by:

$$f(x, y) = (1 - x)^2 + 100 * (y - x^2)^2$$

No of particles =6

Maximum number of iteration =1000

Retardation factor=1

Tolerance value=6

MATLAB code is run by keeping the above parameters constant and varying the inertia weight. Error in the value of optimizing function is taken as fitness value.

AVERAGE ERROR VALUE OF DIFFERENT INERTIA WEIGHT STRATEGIES

TABLE 3: Best inertia weight

S.NO	INERTIA WEIGHT	ERROR VALUE
1	CONSTANT	.0245
2	Random	.0004
3	Linear decreasing	0
4	Logarithmic decreasing	1.001

Observation from the table:

Linear decreasing inertia weight is best as it has least error value.

CASE 2: Dependency of retardation factor with $(C_p + C_g)$

Here Rosenbrock test function is used which is given by:

$$f(x, y) = (1 - x)^2 + 100 * (y - x^2)^2$$

No of particles =30

Maximum value of iteration =1000

Tolerance value=6

MATLAB code is run by keeping the above parameters constant and varying the value of $(C_p + C_g)$ with retardation factor rf. Error in the value of optimizing function is taken as fitness value.

TABLE 4: dependency of retardation factor with $(C_p + C_g)$

S .NO.	Retardation factor(rf)	Maximum value of $(C_p + C_g)$
1	.1	35.8
2	.6	5.9
3	.7	5.1
4	.8	4.4
5	.9	3.6
6	1	3.5
7	1.1	3.2
8	1.2	2.9

9	1.3	2.7
10	17	.2

Observation from table:

1. For Rosenbrock function

   $0.2 < (C_p + C_g) < 35.8$

   For 0.1 < rf < 17

2. As rf increases ($C_p + C_g$) decreases

CASE 3: How to select $C_p and\ C_g$

Here Rosenbrock test function is used which is given by:

$$f(x, y) = (1 - x)^2 + 100 * (y - x^2)^2$$

No of particles =30

Maximium value of iteration =1000

Tolerence value=6,

retardation factor =1

MATLAB code is run by keeping the above parameters constant and varying the value of $\mathrm{C_p}$ and $\mathrm{C_g}$. Error in the value of optimizing function is taken as fitness value.

TABLE 5: To select $C_p and\ C_g$

S. NO.	C_p	C_g	$C_p + C_g$	Number of iteration 'it'
1	1	1.1	2.1	137
	1.1	1		139
2	1	1.2	2.2	142
	1.2	1		141
3	1.3	1.7	3	139
	1.7	1.3		140
4	1.3	1.4	2.7	177
	1.2	1.5		179
5	1.3	2.2	3.5	152
	1.6	1.9		151

Observation from the above table:

1 ($C_p + C_g$) is determining the number of iteration irrespective of the individual value .

CHAPTER 4

LOAD FLOW

4.1 PROBLEM FORMULATION

Load flow solution of the power system requires mainly two steps:

- Formulation of network equation
- Appropriate mathematical technique to solve the equations

Let P_i and Q_i denote the net real and reactive power entering in the network at the bus (i)

$$P_i = \sum_{n=1}^{N} |Y_{in} V_i V_n| \, cos(\theta_{in} + \delta_n + \delta_i) \tag{4}$$

$$Q_i = \sum_{n=1}^{N} |Y_{in} V_i V_n| \, sin(\theta_{in} + \delta_n + \delta_i) \tag{5}$$

Equations 4 and 5 are the polar form of the power flow equations

Let P_{gi} denotes the schedule power being generated at bus (i)

Let P_{di} denotes the schedule power being demand at bus (i)

Then $P_{i,sch}$ = P_{gi} - P_{di} is the net scheduled power being injected into the network at bus (i)

Assuming P_i as the calculated value and denoting it by $P_{i,calc}$

Real power mismatch at bus (i) is defined as

$$\Delta P_i = P_{i,sch} - P_{i,cal} = (P_{gi} - P_{di}) - P_{i,cal} \tag{6}$$

Similarly

Let Q_{gi} denotes the schedule power being generated at bus (i)

Let Q_{di} denotes the schedule power being demand at bus (i)

Then $Q_{i,sch}$ = Q_{gi} - Q_{di} is the net scheduled power being injected into the network at bus (i)

Assuming Q_i as the calculated value and denoting it by $Q_{i,calc}$

Real power mismatch at bus (i) is defined as

$$\Delta Q_i = Q_{i,sch} - Q_{i,cal} = (Q_{gi} - Q_{di}) - Q_{i,cal} \quad (7)$$

4.2 Iterative method

Newton-Raphson method

This method is a powerful technique to solve load flow problem which includes first-derivative information in computing bus voltages. Generally 3-5 iterations are required to solve load flow problem irrespective of the size of the system. This is the most commonly used method to solve load flow problem

Load flow problem is formulated as:

$$\begin{bmatrix} P_{1spec} - P_{1calc}^{(m)} \\ P_{2spec} - P_{2calc}^{(m)} \\ \vdots \\ P_{Nspec} - P_{Ncalc}^{(m)} \\ --- \\ Q_{1spec} - Q_{1calc}^{(m)} \\ Q_{2spec} - Q_{2calc}^{(m)} \\ \vdots \\ Q_{Nspec} - Q_{Ncalc}^{(m)} \end{bmatrix} = \begin{bmatrix} J_1 = \frac{\partial P}{\partial \delta} & J_2 = \frac{\partial P}{\partial |V|} \\ J_3 = \frac{\partial Q}{\partial \delta} & J_4 = \frac{\partial Q}{\partial |V|} \end{bmatrix} \begin{bmatrix} (\delta_1)^{(m+1)} - (\delta_1)^{(m)} \\ (\delta_2)^{(m+1)} - (\delta_2)^{(m)} \\ \vdots \\ (\delta_N)^{(m+1)} - (\delta_N)^{(m)} \\ --- \\ |V_1|^{(m+1)} - |V_1|^{(m)} \\ |V_2|^{(m+1)} - |V_2|^{(m)} \\ \vdots \\ |V_N|^{(m+1)} - |V_N|^{(m)} \end{bmatrix}$$

or, in abbreviated form,

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}.$$

The dimension of the above problem is actually $\left(2N-\left(\text{Number of PV busses}\right)-2\right)$ since $|V|$ updates at PV busses are not required, and since $|V|$ and δ updates at the swing bus are not required.

The solution procedure for the Newton Raphson load flow proceeds with:

1. Initialize the bus voltages. For load busses, use *V* = 1 + *j*0. For generator busses (including the swing bus), use $V = V_{spec} + j0$.

2. Form the Jacobian matrix, and update all bus voltage magnitudes and phase angles, except for those at the swing bus, and except for the voltage magnitudes at PV busses.

3. Check the mismatch *P* and *Q* at each bus. If all are within tolerance (typical tolerance is 0.00001 pu), a solution has been found. Otherwise, return to Step 2.

CHAPTER 5

LOAD FLOW USING PARTICLE SWARM OPTIMIZATION

5.1 Problem formulation:

Let

$P_{i,spec}$ - Specified value of real power at bus (i)

$P_{i,calc}$ - Calculated value of real power at bus (i)

$Q_{i,spec}$ - Specified value of reactive power at bus (i)

$Q_{i,calc}$ - Calculated reactive power at bus (i)

$V_{i,spec}$ - Specified value of voltage at regulated bus (m)

$V_{i,calc}$ - Calculated value of voltage at regulated bus (m)

Now

Real power mismatch

$$MP = P_{i,spec} - P_{i,calc}$$

Reactive power mismatch

$$MQ = Q_{i,spec} - Q_{i,calc} \qquad (8)$$

Voltage mismatch

$$MV = V_{i,spec} - V_{i,calc} \qquad (9)$$

Now

MPS $= MP^2$, MQS $= MQ^2$ and MVS $= MV^2$

Therefore optimizing function becomes

$$f = \text{sum (MPS)} + \text{sum (MQS)} + \text{sum (MVS)} \qquad (10)$$

5.2 Computational procedure:

The complete algorithm is divided in six parts which are described in the block diagram:

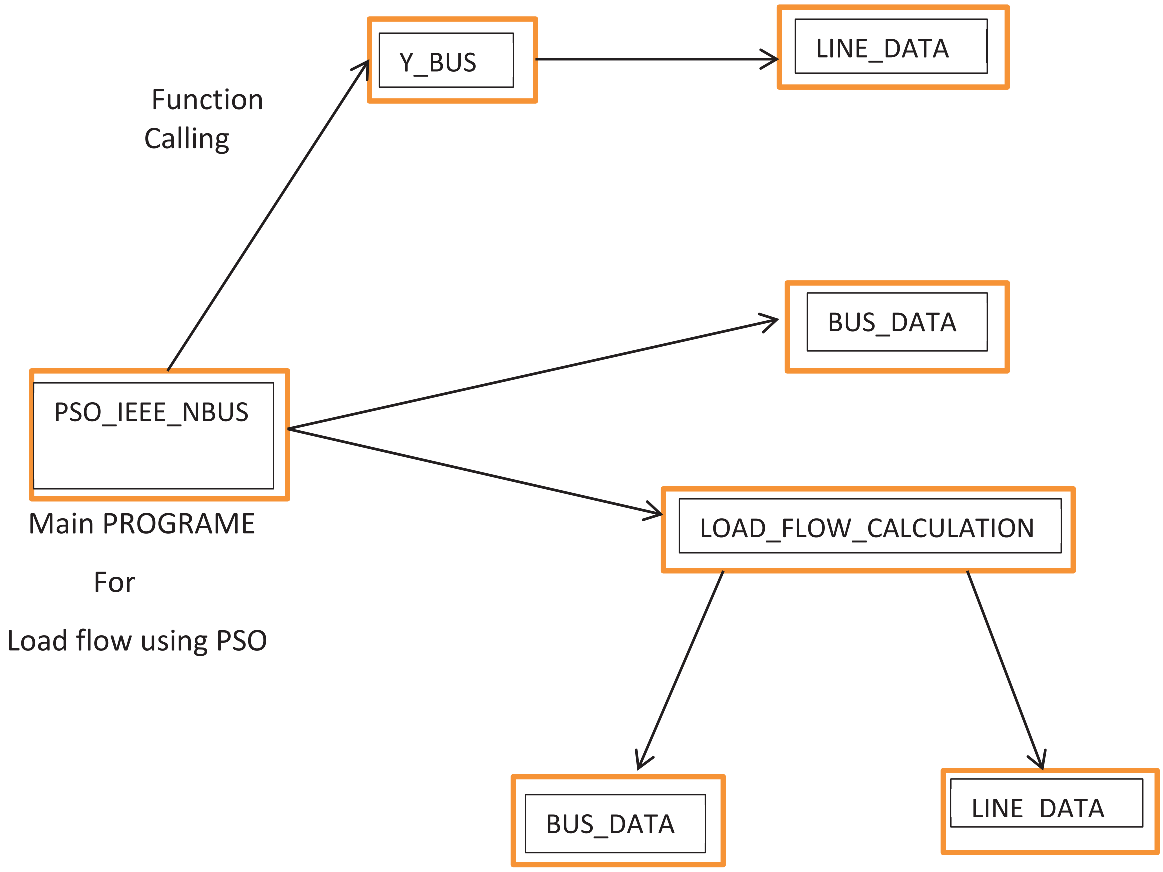

Fig 3: Block diagram for complete algorithm

Computational steps:

1. NBUS = number of buses i.e. IEEE 5,14 and 30
2. Calling Ybus function
   I. Conductance matrix G = real (Y)
   II. Susceptance matrix B = imag (Y)
3. Setting base MVA
4. Calling bus data function
   I. Pg = Real power generation.
   II. Qg = Reactive power generation.
   III. Pl = Real power at load bus.
   IV. Ql = Reactive power at load bus.
   V. Qmin = Minimum reactive power limit at regulated bus.
   VI. Qmax = maximum reactive power limit at regulated bus.
5. PSO PARAMETER INITILIZATION
   I. P = number of particles
   II. It = maximum number of iteration
   III. Rp and Rg are constants
   IV. T = tolerance factor
6. Creating storage matrix for variables
   I. f (p,it) values of function for each particles for each iteration
   II. th (p,it,nbus) values of load angle at each bus taking slack bus as reference
   III. v (p,it,nbus) values of voltage magnitude at each bus

IV. vth(p,it,nbus) velocity of theta for each particles

V. vv(p,it,nbus) velocity of voltage for each particles

VI. vp(p,nbus) personal best velocity of each particles

VII. vg(p,nbus) personal best velocity of each particles

7. initializing the variables

I. v = random values between 0.9 and 1.1

II. th = random values between 0.3 and -0.3

III. vv = random values between 0.5 and -0.5

IV. vth = random values between 0.5 and -0.5

8. initializing the index for checking reactive power limit

PVIND (p, nbus) = index for checking the reactive power limit at voltage controlled bus

Let Q calc = calculated value of reactive power regulated bus

Table 6: index for checking reactive power limit

s. no	Condition	PVIND
1	Q calc < Q min	1
2	Q calc > Q max	2
3	Q min < Q calc < Q max	0

9. Computing the initial values for all particles

I. Objective function value

a. Calculating the value of real power P and reactive power Q at each bus

b. Calculating the value of objective function f

c. Checking the reactive power at pv buses and updating PVIND

II. Calculating initial personal best values of voltage and load angle

III. Calculating initial global best values of voltage and load angle

10. For i= 1 to maximum number of iteration

    I. Initialize inertia weight

    II. Compute the velocity of each particle and update the value of load angle and value of voltage for regulated bus by checking PVIND and updating the value according to table.

    III. Compute for all the particles

        a. Calculating the value of real power P and reactive power Q at each bus

        b. Calculating the value of objective function f

        c. Checking the reactive power at pv buses and updating PVIND

    IV. STOPPING CRITERIA

    Among all the particles If the optimizing function value is less than Some defined minimum value based on tolerance limit then break Out of the iterative loop

11. Update the values of voltage and angle

    i. Bus voltage updating

    ii. Bus angle updating

12. Calling load flow calculation PROGRAME to calculate line loss and power injected at each bus

13. Calculate slack bus power

14. END

BLOCK DIAGRAM FOR IMPLEMENTING LOAD FLOW USING PSO

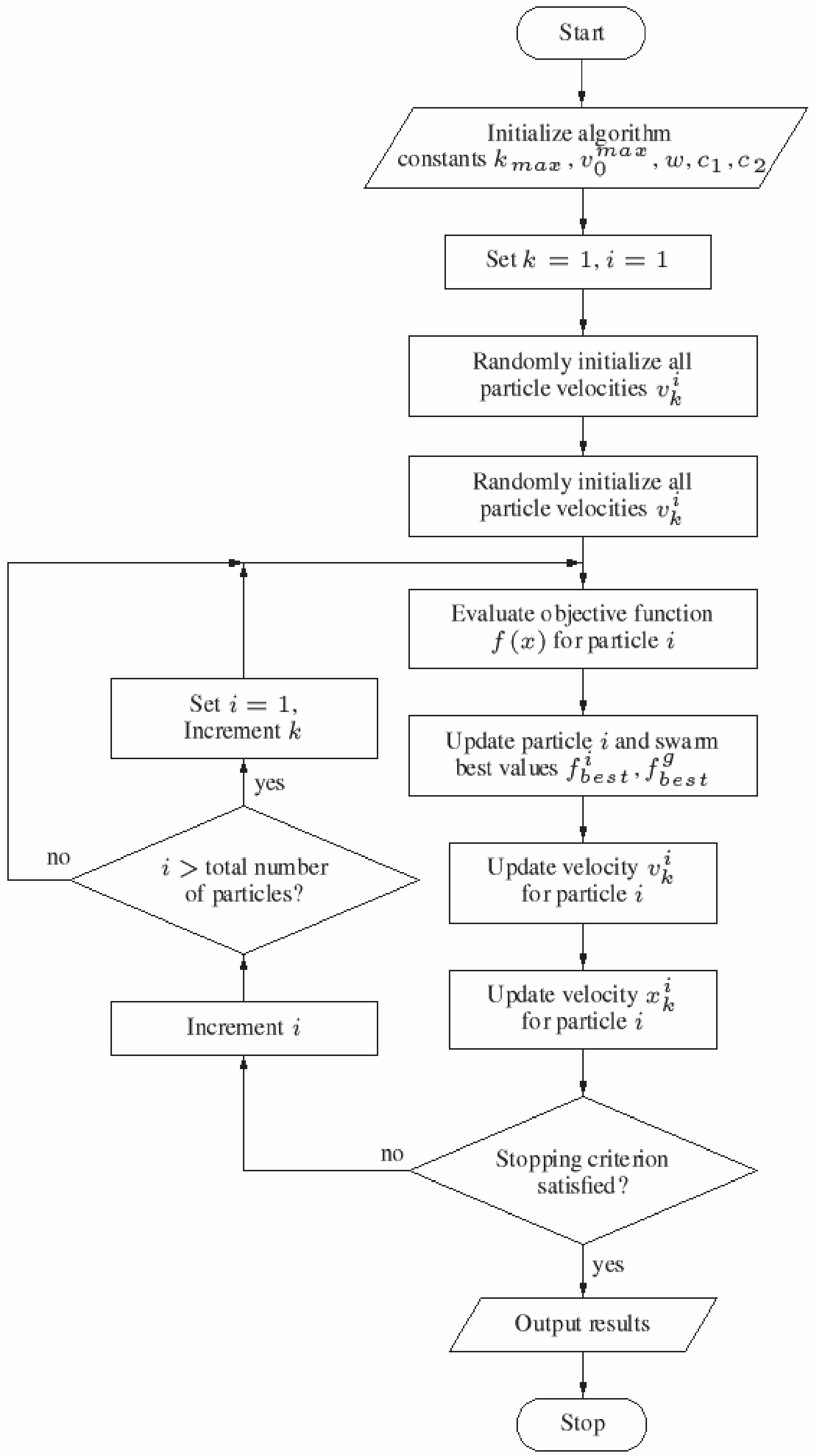

FIG 4: Load flow using PSO

5.3 COMPUTATIONAL RESULTS

5.3.1 IEEE 5 BUS SYSTEM

MATLAB code is developed for the above algorithm. Bus data, line data are taken from appendix. MATLAB code is run and the results are obtained which are as follows:

TABLE 7: LOAD FLOW RESULT OF IEEE 5 BUS SYSTEM USING PSO

Bus	V	Angle	Injection		Generation		Load	
No	pu	Degree	MW	MVar	MW	Mvar	MW	MVar
1	1.0200	0.0000	65.475	30.108	65.475	30.108	0.000	0.000
2	0.9515	-4.5352	-60.000	-30.000	0.000	0.000	60.000	30.000
3	1.0400	1.1342	100.000	51.791	100.000	51.791	0.000	0.000
4	0.9201	-8.7952	-40.000	-10.000	0.000	0.000	40.000	10.000
5	0.9876	-2.5527	-60.000	-20.000	0.000	0.000	60.000	20.000
Total			5.475	21.899	165.475	81.899	160.000	60.000

TABLE 8: LOAD FLOW RESULT OF IEEE 5 BUS SYSTEM USING NR

Bus	V	Angle	Injection		Generation		Load	
No	pu	Degree	MW	MVar	MW	Mvar	MW	MVar
1	1.0200	0.0000	65.150	32.916	65.150	32.916	0.000	0.000
2	0.9548	-3.9413	-60.000	-30.000	-0.000	0.000	60.000	30.000
3	1.0400	2.0008	100.000	47.684	100.000	47.684	0.000	0.000
4	0.9235	-8.0078	-40.000	-10.000	0.000	0.000	40.000	10.000
5	0.9931	-2.0726	-60.000	-20.000	-0.000	-0.000	60.000	20.000
Total			5.150	20.599	165.150	80.599	160.000	60.000

TABLE 9: LINE POWER FLOW AND LOSSES IN IEEE 5 BUS SYSTEM USING PSO

From	To	P	Q	From	To	P	Q	Line Loss	
Bus	Bus	MW	MVar	Bus	Bus	MW	MVar	MW	MVar
1	2	22.345	12.637	2	1	-21.711	-10.104	0.633	2.534
1	4	26.938	12.083	4	1	-25.681	-7.057	1.257	5.027
1	5	25.119	10.745	5	1	-24.760	-9.310	0.359	1.435
2	3	-55.339	-25.843	3	2	57.399	34.083	2.060	8.240
2	4	17.202	3.771	4	2	-16.859	-2.401	0.343	1.370
3	5	37.742	18.876	5	3	-36.919	-15.583	0.823	3.293
Total Loss								5.475	21.899

TABLE 10: LINE POWER FLOW AND LOSSES IN IEEE 5 BUS SYSTEM USING NR

From	To	P	Q	From	To	P	Q	Line Loss	
Bus	Bus	MW	MVar	Bus	Bus	MW	MVar	MW	MVar
1	2	19.800	12.264	2	1	-19.279	-10.178	0.521	2.086
1	4	24.805	11.743	4	1	-23.719	-7.399	1.086	4.344
1	5	20.544	8.909	5	1	-20.303	-7.945	0.241	0.964
2	3	-57.321	-23.698	3	2	59.431	32.139	2.110	8.441
2	4	16.600	3.876	4	2	-16.281	-2.601	0.319	1.275
3	5	40.569	15.545	5	3	-39.697	-12.055	0.873	3.490
Total Loss								5.150	20.599

TABLE 11: COMPARISION BETWEEN PSO AND NR FOR IEEE 5 BUS SYSTEM

S NO.		PSO	NR
1	Number of iterations	45	3
2	Total power loss (MW)	5.475	5.150
3	Computational time(sec)	0.33678	0.72

5.3.2 IEEE 14 BUS SYSTEM

MATLAB code is developed for the above algorithm. Bus data, line data are taken from appendix. MATLAB code is run and the results are obtained which are as follows:

TABLE 12: LOAD FLOW RESULT OF IEEE 14 BUS SYSTEM USING PSO

Bus	V	Angle	Injection		Generation		Load	
No	pu	Degree	MW	MVar	MW	Mvar	MW	MVar
1	1.0600	0.0000	230.075	30.383	230.075	30.383	0.000	0.000
2	1.0440	-6.2962	18.300	24.600	40.000	37.300	21.700	12.700
3	1.0100	-9.2596	-94.200	-31.890	0.000	-12.890	94.200	19.000
4	1.0204	-7.0783	-47.800	3.900	0.000	0.000	47.800	-3.900
5	1.0329	-6.6486	-7.600	-1.600	0.000	0.000	7.600	1.600
6	1.0743	-9.2394	-11.200	9.000	0.000	16.500	11.200	7.500
7	1.0401	-5.7283	0.000	0.000	0.000	0.000	0.000	0.000
8	1.0903	-4.3425	0.000	24.000	0.000	24.000	0.000	0.000
9	1.0104	-7.7342	-29.500	-16.600	0.000	0.000	29.500	16.600
10	1.0012	-8.1160	-9.000	-5.800	0.000	0.000	9.000	5.800
11	1.0450	-9.2686	-3.500	-1.800	0.000	0.000	3.500	1.800
12	1.0572	-7.8813	-6.100	-1.600	0.000	0.000	6.100	1.600
13	1.0384	-9.7470	-13.500	-5.800	0.000	0.000	13.500	5.800
14	1.0082	-7.1097	-14.900	-5.000	0.000	0.000	14.900	5.000
Total			11.075	21.793	270.075	95.293	259.000	73.500

TABLE 13: LOAD FLOW RESULT OF IEEE 14 BUS SYSTEM USING NR

Bus	V	Angle	Injection		Generation		Load	
No	pu	Degree	MW	MVar	MW	Mvar	MW	MVar
1	1.0600	0.0000	232.476	-17.890	232.476	-17.890	0.000	0.000
2	1.0450	-4.9786	18.300	29.866	40.000	42.566	21.700	12.700
3	1.0100	-12.7220	-94.200	6.856	0.000	25.856	94.200	19.000
4	1.0164	-10.2784	-47.800	3.900	-0.000	-0.000	47.800	-3.900
5	1.0224	-8.8126	-7.600	-1.600	-0.000	0.000	7.600	1.600
6	1.0300	-14.8755	-11.200	23.720	-0.000	31.220	11.200	7.500
7	1.0216	-13.5406	-0.000	-0.000	-0.000	-0.000	0.000	0.000
8	1.0600	-13.5406	-0.000	23.111	-0.000	23.111	0.000	0.000
9	1.0017	-15.2812	-29.500	-16.600	-0.000	-0.000	29.500	16.60
10	0.9989	-15.5164	-9.000	-5.800	0.000	0.000	9.000	5.800
11	1.0106	-15.3282	-3.500	-1.800	0.000	0.000	3.500	1.800
12	1.0135	-15.7847	-6.100	-1.600	0.000	0.000	6.100	1.600
13	1.0076	-15.8333	-13.500	-5.800	0.000	0.000	13.500	5.800
14	0.9853	-16.6226	-14.900	-5.000	0.000	-0.000	14.900	5.000
Total			13.476	31.363	272.476	104.863	259.00	73.50

TABLE 14: LINE FLOW AND LOSSES IN IEEE 14 BUS SYSTEM USING PSO

From	To	P	Q	From	To	P	Q	Line Loss	
Bus	Bus	MW	MVar	Bus	Bus	MW	MVar	MW	MVar
1	2	28.622	-53.887	2	1	-22.201	55.848	6.422	1.961
1	5	57.385	2.268	5	1	-55.799	4.279	1.586	6.547
2	3	30.264	11.485	3	2	-29.812	-9.582	0.452	1.903
2	4	11.613	10.212	4	2	-11.486	-9.825	0.127	0.387
2	5	5.424	4.914	5	2	-5.396	-4.829	0.028	0.085
3	4	-21.829	2.829	4	3	22.147	-2.016	0.318	0.812
4	5	-25.756	-22.034	5	4	25.903	22.499	0.147	0.465
4	7	-11.958	-9.463	7	4	11.958	9.930	0.000	0.467
4	9	2.122	1.857	9	4	-2.122	-1.815	0.000	0.042
5	6	19.903	-16.508	6	5	-19.903	18.087	0.000	1.579
6	11	6.380	12.759	11	6	-6.213	-12.408	0.167	0.351
6	12	-5.691	10.056	12	6	5.833	-9.760	0.142	0.296
6	13	17.989	20.485	13	6	-17.563	-19.646	0.426	0.839
7	8	-15.569	-29.436	8	7	15.569	31.242	0.000	1.806
7	9	33.438	28.703	9	7	-33.438	-26.728	-0.000	1.975
9	10	10.603	6.972	10	9	-10.553	-6.839	0.050	0.133
9	14	-3.540	3.990	14	9	3.575	-3.943	0.035	0.047
10	11	1.056	-23.177	11	10	-0.616	24.209	0.441	1.031
12	13	13.124	-4.304	13	12	-12.747	4.645	0.377	0.3
13	14	-7.452	13.009	14	13	7.809	-12.283	0.356	0.725
Total Loss								11.075	21.793

TABLE 15: LINE FLOW AND LOSSES OF LOAD FLOW OF IEEEE 14 BUS USING NR

From	To	P	Q	From	To	P	Q	Line Loss	
Bus	Bus	MW	MVar	Bus	Bus	MW	MVar	MW	MVar
1	2	156.761	-17.410	2	1	-152.470	30.510	4.291	13.101
1	5	75.715	5.249	5	1	-72.945	6.185	2.770	11.434
2	3	73.246	5.951	3	2	-70.922	3.839	2.324	9.790
2	4	55.999	1.087	4	2	-54.329	3.978	1.669	5.065
2	5	41.525	1.338	5	2	-40.625	1.410	0.900	2.749
3	4	-23.278	5.903	4	3	23.656	-4.937	0.379	0.967
4	5	-61.344	5.629	5	4	61.834	-4.082	0.490	1.547
4	7	28.254	-1.734	7	4	-28.254	3.356	-0.000	1.622
4	9	15.963	3.381	9	4	-15.963	-1.947	0.000	1.433
5	6	44.135	-0.733	6	5	-44.135	5.431	0.000	4.697
6	11	7.275	6.575	11	6	-7.189	-6.395	0.086	0.180
6	12	7.873	2.908	12	6	-7.791	-2.738	0.082	0.170
6	13	17.788	8.806	13	6	-17.542	-8.323	0.246	0.484
7	8	0.000	-22.273	8	7	-0.000	23.111	0.000	0.837
7	9	28.254	18.917	9	7	-28.254	-17.698	0.000	1.219
9	10	5.349	1.296	10	9	-5.339	-1.270	0.010	0.026
9	14	9.368	1.750	14	9	-9.253	-1.505	0.115	0.245
10	11	-3.661	-4.530	11	10	3.689	4.595	0.028	0.065
12	13	1.691	1.138	13	12	-1.682	-1.130	0.009	0.008
13	14	5.724	3.653	14	13	-5.647	-3.495	0.078	0.158
Total Loss								13.476	55.797

TABLE 16: COMPARISION BETWEEN PSO AND NR FOR IEEE 14 BUS SYSTEM

S NO.		**PSO**	**NR**
1	Number of iterations	51	5
2	Total power loss (MW)	11.075	13.476
3	Computational time(sec)	1.128009	0.866

5.3.1 IEEE 30 BUS SYSTEM

MATLAB code is developed for the above algorithm. Bus data, line data are taken from appendix. MATLAB code is run and the results are obtained which are as follows:

TABLE 17: LOAD FLOW RESULT OF IEEE 30 BUS SYSTEM USING PSO

Bus	V	Angle	Injection		Generation		Load	
No	pu	Degree	MW	MVar	MW	Mvar	MW	MVar
1	1.0600	0.0000	264.168	125.366	264.168	125.366	0.000	0.000
2	1.0450	-3.8849	18.300	22.153	40.000	34.853	21.700	12.700
3	1.0137	-6.2626	-2.400	-1.200	0.000	0.000	2.400	1.200
4	1.0097	-6.9089	-7.600	-1.600	0.000	0.000	7.600	1.600
5	1.0100	-8.6974	-94.200	-28.356	0.000	-9.356	94.200	19.000
6	1.0133	-7.3524	0.000	0.000	0.000	0.000	0.000	0.000
7	1.0100	-9.2153	-22.800	-10.900	0.000	0.000	22.800	10.900
8	1.0100	-7.8719	-30.000	-37.685	0.000	-7.685	30.000	30.000
9	0.9941	-8.8187	0.000	0.000	0.000	0.000	0.000	0.000
10	1.0478	-8.1936	-5.800	-2.000	0.000	0.000	5.800	2.000
11	1.0972	-12.4705	0.000	24.000	0.000	24.000	0.000	0.000
12	1.0380	-6.4905	-11.200	-7.500	0.000	0.000	11.200	7.500
13	1.0717	-5.9838	0.000	24.000	0.000	24.000	0.000	0.000
14	1.0227	-8.0667	-6.200	-1.600	0.000	0.000	6.200	1.600
15	1.0280	-6.1052	-8.200	-2.500	0.000	0.000	8.200	2.500
16	1.0480	-6.8954	-3.500	-1.800	0.000	0.000	3.500	1.800
17	1.0732	-8.2672	-9.000	-5.800	0.000	0.000	9.000	5.800
18	1.0657	-12.0342	-3.200	-0.900	0.000	0.000	3.200	0.900
19	1.0624	-9.0495	-9.500	-3.400	0.000	0.000	9.500	3.400
20	1.0549	-8.0334	-2.200	-0.700	0.000	0.000	2.200	0.700
21	1.0102	-6.9763	-17.500	-11.200	0.000	0.000	17.500	11.200
22	1.0350	-10.4937	0.000	0.000	0.000	0.000	0.000	0.000
23	1.0034	-6.3861	-3.200	-1.600	0.000	0.000	3.200	1.600

24	1.0330	-11.0507	-8.700	-6.700	0.000	0.000	8.700	6.700
25	1.0040	-8.3528	0.000	0.000	0.000	0.000	0.000	0.000
26	1.0447	-9.7274	-3.500	-2.300	0.000	0.000	3.500	2.300
27	1.0313	-4.1818	0.000	0.000	0.000	0.000	0.000	0.000
28	1.0258	-6.8211	0.000	0.000	0.000	0.000	0.000	0.000
29	0.9749	-11.0570	-2.400	-0.900	0.000	0.000	2.400	0.900
30	1.0143	-10.4435	-10.600	-1.900	0.000	0.000	10.600	1.900
Total			20.768	64.978	304.168	191.178	283.400	126.200

TABLE 18: LOAD FLOW RESULT OF IEEE 30 BUS SYSTEM USING NR

Bus	V	Angle	Injection		Generation		Load	
No	pu	Degree	MW	MVar	MW	Mvar	MW	MVar
1	1.0600	0.0000	260.928	-17.118	260.928	-17.118	0.000	0.000
2	1.0430	-5.3474	18.300	35.066	40.000	47.766	21.700	12.700
3	1.0217	-7.5448	-2.400	-1.200	-0.000	0.000	2.400	1.200
4	1.0129	-9.2989	-7.600	-1.600	0.000	0.000	7.600	1.600
5	1.0100	-14.1542	-94.200	16.965	-0.000	35.965	94.200	19.000
6	1.0121	-11.0880	0.000	0.000	0.000	0.000	0.000	0.000
7	1.0035	-12.8734	-22.800	-10.900	-0.000	-0.000	22.800	10.900
8	1.0100	-11.8039	-30.000	0.691	0.000	30.691	30.000	30.000
9	1.0507	-14.1363	0.000	0.000	0.000	0.000	0.000	0.000
10	1.0438	-15.7341	-5.800	17.000	-0.000	19.000	5.800	2.000
11	1.0820	-14.1363	0.000	16.270	0.000	16.270	0.000	0.000
12	1.0576	-14.9416	-11.200	-7.500	0.000	-0.000	11.200	7.500
13	1.0710	-14.9416	0.000	10.247	0.000	10.247	0.000	0.000
14	1.0429	-15.8244	-6.200	-1.600	-0.000	0.000	6.200	1.600
15	1.0384	-15.9101	-8.200	-2.500	-0.000	0.000	8.200	2.500
16	1.0445	-15.5487	-3.500	-1.800	-0.000	-0.000	3.500	1.800
17	1.0387	-15.8856	-9.000	-5.800	-0.000	0.000	9.000	5.800
18	1.0282	-16.5425	-3.200	-0.900	-0.000	0.000	3.200	0.900

19	1.0252	-16.7273	-9.500	-3.400	-0.000	0.000	9.500	3.400
20	1.0291	-16.5363	-2.200	-0.700	0.000	-0.000	2.200	0.700
21	1.0293	-16.2462	-17.500	-11.200	-0.000	0.000	17.500	11.200
22	1.0353	-16.0738	0.000	-0.000	0.000	-0.000	0.000	0.000
23	1.0291	-16.2528	-3.200	-1.600	0.000	0.000	3.200	1.600
24	1.0237	-16.4409	-8.700	-2.400	-0.000	4.300	8.700	6.700
25	1.0202	-16.0539	0.000	0.000	0.000	0.000	0.000	0.000
26	1.0025	-16.4712	-3.500	-2.300	-0.000	-0.000	3.500	2.300
27	1.0265	-15.5558	0.000	-0.000	0.000	-0.000	0.000	0.000
28	1.0109	-11.7436	0.000	0.000	0.000	0.000	0.000	0.000
29	1.0067	-16.7777	-2.400	-0.900	-0.000	-0.000	2.400	0.900
30	0.9953	-17.6546	-10.600	-1.900	0.000	0.000	10.600	1.900
Total			17.528	20.921	300.928	147.121	283.400	126.200

TABLE 19: LINE FLOW AND LOSSES IN IEEE 30 BUS SYSTEM USING PSO

From	To	P	Q	From	To	P	Q	Line Loss	
Bus	Bus	MW	MVar	Bus	Bus	MW	MVar	MW	MVar
1	2	127.064	-10.350	2	1	-124.287	18.667	2.777	8.317
1	3	74.558	13.161	3	1	-72.252	-4.733	2.306	8.428
2	4	35.475	10.451	4	2	-34.761	-8.276	0.714	2.175
3	4	30.596	0.380	4	3	-30.476	-0.035	0.120	0.345
2	5	46.837	9.172	5	2	-45.852	-5.036	0.985	4.136
2	6	38.687	7.165	6	2	-37.864	-4.666	0.824	2.499
4	6	15.375	-13.060	6	4	-15.328	13.225	0.048	0.165
5	7	6.875	-2.708	7	5	-6.851	2.770	0.025	0.062
6	7	38.054	-7.731	7	6	-37.662	8.935	0.392	1.204
6	8	22.525	1.523	8	6	-22.465	-1.314	0.060	0.209
6	9	12.671	9.688	9	6	-12.671	-9.183	0.000	0.504
6	10	2.893	-6.477	10	6	-2.893	6.741	0.000	0.264
9	11	33.399	-48.182	11	9	-33.399	55.415	0.000	7.234

9	10	-10.332	-48.454	10	9	10.332	51.186	0.000	2.732
4	12	-3.208	-11.968	12	4	3.208	12.327	0.000	0.359
12	13	-7.026	-24.977	13	12	7.026	25.851	0.000	0.875
12	14	11.758	0.723	14	12	-11.599	-0.394	0.159	0.330
12	15	-1.161	8.552	15	12	1.207	-8.462	0.046	0.090
12	16	1.129	-5.761	16	12	-1.099	5.824	0.030	0.064
14	15	-9.311	7.868	15	14	9.625	-7.584	0.314	0.284
16	17	6.930	-16.504	17	16	-6.689	17.065	0.240	0.561
15	18	35.774	-32.619	18	15	-33.394	37.465	2.380	4.846
18	19	-35.090	21.304	19	18	36.038	-19.387	0.948	1.917
19	20	-18.628	21.198	20	19	18.868	-20.718	0.240	0.480
10	20	-2.560	-2.414	20	10	2.571	2.438	0.011	0.024
10	17	-9.018	-27.975	17	10	9.273	28.640	0.255	0.665
10	21	-4.479	54.978	21	10	5.443	-52.902	0.964	2.076
10	22	27.250	-3.681	22	10	-26.749	4.713	0.501	1.032
21	23	-24.060	41.077	23	21	24.318	-40.553	0.258	0.524
15	23	6.984	9.051	23	15	-6.860	-8.801	0.124	0.250
22	24	4.653	-1.796	24	22	-4.626	1.838	0.027	0.042
23	24	21.368	-20.154	24	23	-20.237	22.468	1.131	2.313
24	25	-7.096	13.506	25	24	7.507	-12.788	0.411	0.718
25	26	-0.363	-10.434	26	25	0.639	10.845	0.275	0.411
25	27	-33.178	5.551	27	25	34.405	-3.209	1.227	2.343
28	27	-12.708	-1.196	27	28	12.708	1.789	0.000	0.594
27	29	29.149	0.317	29	27	-27.393	3.001	1.756	3.318
27	30	16.400	-4.762	30	27	-15.522	6.414	0.878	1.653
29	30	-5.318	-5.636	30	29	5.470	5.922	0.152	0.286
8	28	-10.901	-4.405	28	8	10.987	4.676	0.086	0.271
6	28	-20.413	-15.313	28	6	20.520	15.693	0.107	0.380
Total Loss								20.768	64.978

TABLE 20: LINE FLOW AND LOSSES IN IEEE 30 BUS SYSTEM USING NR

From Bus	To Bus	P MW	Q MVar	From Bus	To Bus	P MW	Q MVar	Line Loss MW	Line Loss MVar
1	2	173.143	-18.108	2	1	-167.964	33.617	5.179	15.509
1	3	87.785	6.248	3	1	-84.669	5.140	3.116	11.388
2	4	43.619	5.194	4	2	-42.607	-2.113	1.011	3.081
3	4	82.269	-3.772	4	3	-81.412	6.235	0.858	2.463
2	5	82.293	4.033	5	2	-79.347	8.342	2.945	12.374
2	6	60.353	1.403	6	2	-58.406	4.503	1.946	5.906
4	6	72.272	-17.521	6	4	-71.631	19.753	0.641	2.231
5	7	-14.853	11.796	7	5	15.015	-11.387	0.162	0.409
6	7	38.195	-1.201	7	6	-37.815	2.370	0.381	1.169
6	8	29.490	-3.214	8	6	-29.387	3.574	0.103	0.361
6	9	27.799	-18.485	9	6	-27.799	20.698	0.000	2.213
6	10	15.882	-5.306	10	6	-15.882	6.781	-0.000	1.475
9	11	-0.000	-15.799	11	9	0.000	16.270	0.000	0.470
9	10	27.799	7.041	10	9	-27.799	-6.222	0.000	0.819
4	12	44.147	-16.795	12	4	-44.147	21.983	0.000	5.18
12	13	-0.000	-10.119	13	12	0.000	10.247	0.000	0.128
12	14	7.790	2.390	14	12	-7.717	-2.238	0.073	0.152
12	15	17.639	6.705	15	12	-17.429	-6.290	0.211	0.415
12	16	7.518	3.420	16	12	-7.460	-3.299	0.058	0.121
14	15	1.517	0.638	15	14	-1.511	-0.633	0.006	0.005
16	17	3.960	1.499	17	16	-3.946	-1.468	0.014	0.032
15	18	6.291	1.829	18	15	-6.249	-1.742	0.043	0.087
18	19	3.049	0.842	19	18	-3.043	-0.830	0.006	0.012
19	20	-6.457	-2.570	20	19	6.473	2.601	0.016	0.031
10	20	8.749	3.471	20	10	-8.673	-3.301	0.076	0.170
10	17	5.067	4.367	17	10	-5.054	-4.332	0.013	0.035
10	21	18.286	11.764	21	10	-18.135	-11.439	0.151	0.325

10	22	5.780	3.107	22	10	-5.751	-3.048	0.029	0.059
21	23	0.635	0.239	23	21	-0.635	-0.239	0.000	0.000
15	23	4.449	2.593	23	15	-4.424	-2.544	0.025	0.050
22	24	5.751	3.048	24	22	-5.706	-2.977	0.045	0.071
23	24	1.859	1.183	24	23	-1.853	-1.171	0.006	0.012
24	25	-1.142	1.748	25	24	1.149	-1.734	0.008	0.014
25	26	3.544	2.366	26	25	-3.500	-2.300	0.044	0.066
25	27	-4.694	-0.632	27	25	4.717	0.677	0.024	0.045
28	27	17.998	-3.529	27	28	-17.998	4.791	0.000	1.262
27	29	6.189	1.667	29	27	-6.103	-1.505	0.086	0.162
27	30	7.091	1.661	30	27	-6.930	-1.358	0.161	0.303
29	30	3.703	0.605	30	29	-3.670	-0.542	0.033	0.063
8	28	-0.613	-0.241	28	8	0.614	0.242	0.000	0.001
6	28	18.670	-3.094	28	6	-18.611	3.304	0.059	0.209
Total Loss								17.528	68.888

TABLE 21: COMPARISION FOR IEEE 30 BUS SYSTEM

S NO.		**PSO**	**NR**
1	Number of iterations	105	6
2	Total power loss (MW)	20.768	17.528
3	Computational time(sec)	7.93	1

TABLE 22: COMPARISION OF LINE LOSSES IN IEEE 5, 14 AND 30 BUS SYSTEM

	Real power loss in line (MW)		Reactive power demand (MVAR)	
	PSO	NR	PSO	NR
IEEE 5 BUS	**5.475**	**5.150**	**21.895**	**20.599**
IEEE 14 BUS	**11.075**	**13.476**	**21.793**	**55.797**
IEEE 30 BUS	**20.768**	**17.528**	**64.978**	**68.88**

5.6 DISCUSSION:

Real power loss:

- **IEEE 5 bus:** Real power loss in line computed from PSO has been found to be slightly more than that of obtained from NR method.
- **IEEE 14 bus:** Real power loss in line computed from PSO has been found to be less than that of obtained from NR method.
- **IEEE 30 bus:** Real power loss in line computed from PSO has been found to be more than that of obtained from NR method.

Reactive power demand in line:

- **IEEE 5 bus:** Reactive power demand in line computed from PSO has found to be slightly more than that of obtained from NR method.
- **IEEE 14 BUS:** Reactive power demand in line computed from PSO has found to be very less than that of obtained from NR method.
- **IEEE 30 bus:** Reactive power demand in line computed from PSO has found to be slightly less than that of obtained from NR method.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSIONS

New technique of solving load flow is developed. Basic PSO has found to be the simplest one to implement the load flow.

In this work parameter dependency in PSO algorithm is obtained using different inertia weight by changing different parameters like tolerance factor, retardation factor, cognitive factors etc. The new algorithm is applied to solve load flow of IEEE 5, 14 and 30 bus systems.

The results obtained are compared with Newton-Raphson method and proved the validity of the algorithm as a new tool to calculate load flow with ease.

As regards computational time, conventionally it seems that PSO would take much more computational time as compared to NR method. But in this work it is observed that for 5 Bus system, computational time for PSO is lesser than NR method. For 14 bus system computational time for PSO is little more than NR method. Further for 30 bus system computational time is much more than NR method.

However it is pertinent to mention that computational time for PSO can reduced considerably by tuning the PSO parameters more accurately.

6.2 FUTURE SCOPE

PSO has many area of research. Number of iterations and computational time can be reduced. Improvement in algorithm can be done like out of number of particles performing iteration we can reject the particles which are far from the optimum point and take only the best particles for next iteration. Further improvement in velocity update can be done for faster convergence. Algorithm can be made a generalized one to take data for any number of bus and compute the load flow. Further the optimal load flow can be done to compute the cost.

REFERENCES

[1] J. Kennedy, R.C.Eberhart, et al., "Particle swarm optimization", In Proceedings of IEEE international conference on neural networks,ivolume 4, pages 1942–1948. 1995

[2] Amgad A. EL-Dib, Hosam K.M. Youssef, Member, M.M.EL-Metwally and Z. Osrnan, "Load Flow Solution Using Hybrid Particle Swarm Optimization" in International conference on ICEEC,Pages: 742 – 746,Year-2004

[3] Yamille del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi and Ronald G Harley" Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems"in IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12, NO. 2,APRIL 2008

[4] R.Mageshvaran1, I. Jacob Raglend2, V.Yuvaraj3, "Implementation of Non-Traditional Optimization Techniques (PSO, CPSO, HDE) for the Optimal Load Flow Solution" in IEEE Region 10 Conference on TENCON,Pages: 1 – 6 ,Year:2008

[5] M. R. AlRashidi, et al," A Survey of Particle Swarm Optimization Applications in Electric Power Systems" in IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 4, AUGUST 2009

[6] Ibrahim Oumarou, Prof. Daozhuo Jiang, Prof. Cao Yijia, "Particle Swarm Optimization Applied to Optimal Power flow solution" Fifth International Conference on Natural Computation"Year:2009

[7] Lin Lu, Qi Luo, Jun-yong Liu, Chuan LongAn "Improved Particle Swarm Optimization Algorithm" in IEEE International Conference on Granular Computing, Pages: 486 – 490,year:2008.

[8] Camila P. Salomon, Germano Lambert-Torres, Helga G. Martins, Cláudio Ferreira, CláudioI. A. Costa "Load Flow Computation via Particle Swarm Optimization" in 9th IEEE/IAS International Conference on Industry Applications, year: 2009

[9] A.Arunya Revathi and Dr.N.S.Marimuthu "Application of Modified Particle Swarm Optimization for Load Flow Problem" Journal of Electrical Engineering, volume 8,edition 3,article-7,year-2008

[10] C.Kumar and Dr. Ch. Padmanabha Raju in "Constrained Optimal Power Flow using Particle Swarm Optimization" in International Journal of Emerging Technology and Advanced ISSN 2250-2459, Volume 2, Issue 2, February 2012

[11] Camila Paes Salomon, Germano Lambert-Torres Luiz Eduardo Borges da Silva, Maurilio Pereira Coutinho and Carlos Henrique Valerio de Moraes "A HYBRID PARTICLE SWARM OPTIMIZATION APPROACH FOR LOAD-FLOW COMPUTATION" in International Journal of Innovative Computing, Information and Control ICIC International 2013 ISSN 1349-4198 Volume 9, Number 11, November 2013

[12] Bhupender Sharma, Shivani Sehgal, Ajay Nain "Particle Swarm Optimization and Genetic Algorithm based Optimal Power Flow Solutions" in International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 2, Issue 7, July 2013

[13] Deepak Saini and Neeraj Saini "A Study of Load Flow Analysis Using Particle Swarm Optimization" in Int. Journal of Engineering Research and Applications ISSN 2248 -9622 Vol. 5, Issue 1(Part 1, pp.125-131

[14] H.R. Li and Y.L. Gao., "Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation", In —2009 Second International Conference on Information and Computing Science, pages 66–69. IEEE, 2009.

[15] Y. Gao, X. An, and J. Liu., "A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation", In Computational Intelligence and Security, 2008. CIS'08. International Conference on, volume 1, pages 61–65. IEEE, 2008.

[16] James Kennedy and Tim Blackwell Riccardo Poli, "Particle swarm optimization An overview," Swarm Intelligence, vol. 1, no. 1, pp. 33–57, 2007.

[17] Dr. N.K. Jain, Ms. Uma Nangia, Dr. C.L. Wadhwa "Investigaion on Multiobjective Optimal Load Flow Study by Sequential Goal Programming" IE(I) Journal-EL Vol 77, August 1996.

[18] Dr. N.K. Jain, Ms. Uma Nangia, Dr. C. L. Wadhwa "Multiobjective Optimal Load Flow Based on Ideal Distance Minimization In 3D Space" Electrical Power and Energy Systems 23(2001) 847-855 (ELSEVIER)

[19] Singiresu S. Rao "Engineering Optimization Theory and Practice (Third Enlarged Edition)" New Age International Publishers ISBN:978-81-224-2723-3.

[20] Nangia, U., N.K. Jain , C.L. Wadhwa, 'Multiobjective optimal load flow based on ideal distance minimization in 3D space', Electrical power and energy systems (23), 2001,pp. 847- 855

[21] Nangia, Uma, Jain, N.K., Wadhwa, C.L., 'Comprehensive Comparison of Various Multiobjective Techniques', Engineering Intelligent Systems Journal, Vol. 11, No. 3, Sept.,2003, pp. 123-132.

[22] C.L. WADHWA "electrical power system" fifth edition, new age international Publication,2009

[23] JOHN J GRAINGER, WILLIAM D. STEVENSON,JR. "power system analysis" TMH Publication,13th reprint 2010

APPENDIX I

i) IEEE 5 BUS SYSTEM:

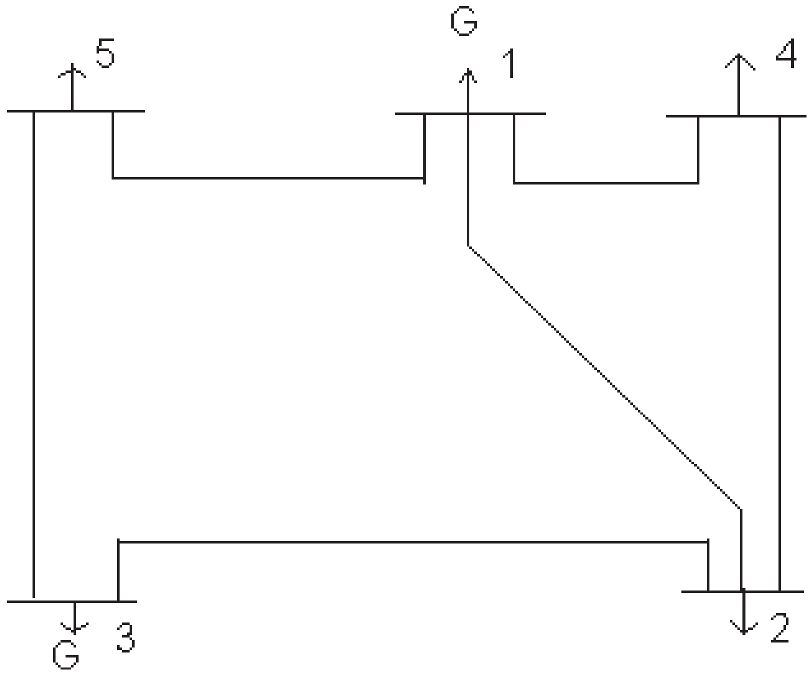

Fig. (I-A): BUS-CODE DIAGRAM 5 BUS SYSTEM

TABLE (I-A)
LINE DATA OR IMPEDANCE DATA (5 BUS SYSTEM)

LINE DESIGNATION	*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 (I-B)
BUS DATA or OPERATING CONDITIONS (5 BUS SYSTEM)

	GENERATION	GENERATION	LOAD	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 (I-C)
REGULATED BUS DATA (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 $0.9 <= V_i <= 1.05$

ii) IEEE 14 BUS SYSTEM:

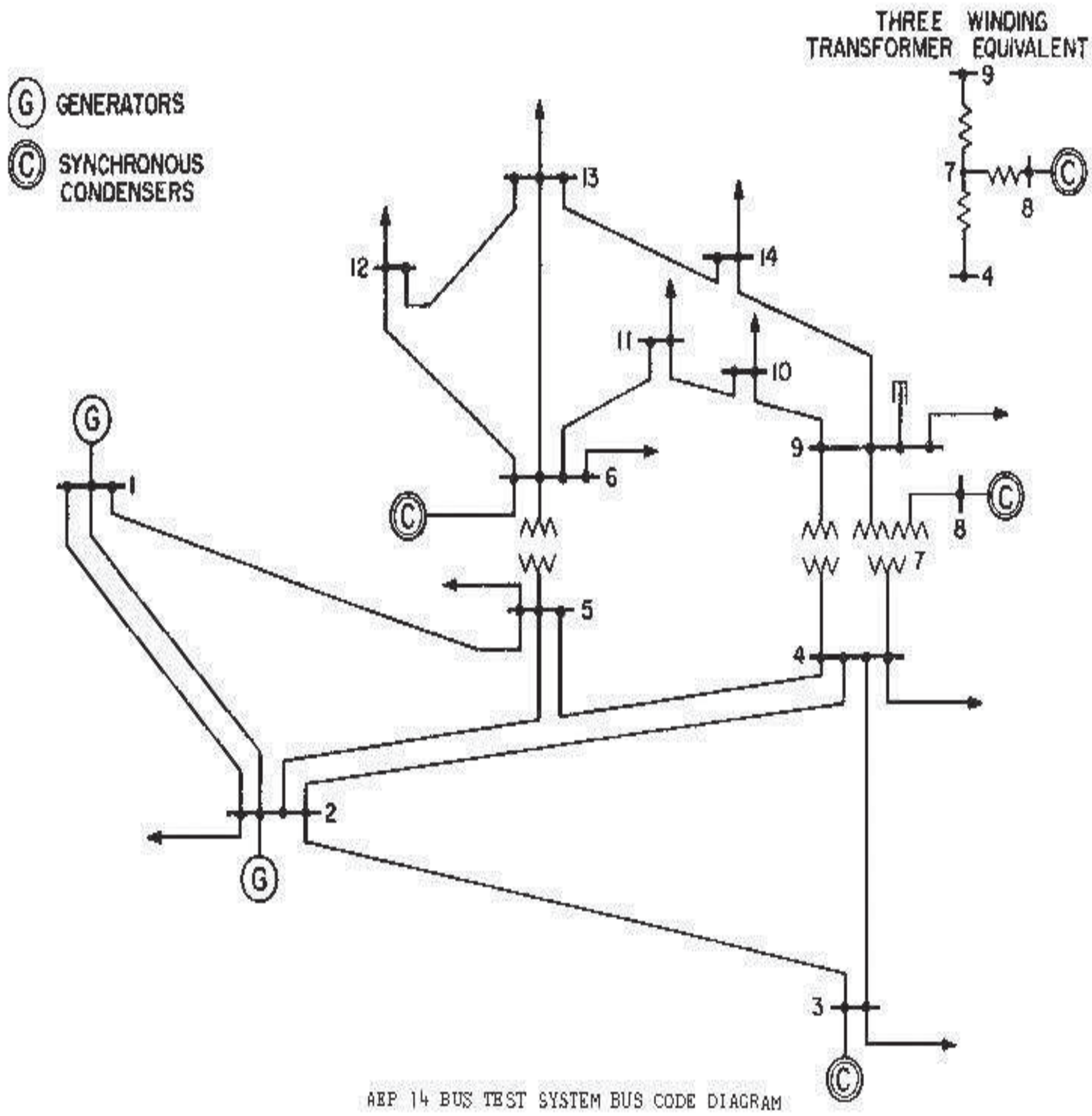

Fig. (I-B): BUS-CODE DIAGRAM 14 BUS SYSTEM

TABLE (I-D)
IMPEDANCE AND LINE-CHARGING DATA (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 the total charging of line.

TABLE (I-E)
BUS DATA OR OPERATING CONDITIONS (14 BUS SYSTEM)

			Generation	Generation	Load	Load
Bus No.	Magnitude p.u.	Phase Angle deg.	MW	MVAR	MW	MVAR
1*	1.06	0	0	0	0	0
2	1	0	40	0	21.7	12.7
3	1	0	0	0	94.2	19.0
4	1	0	0	0	47.8	-3.9
5	1	0	0	0	7.6	1.6
6	1	0	0	0	11.2	7.5
7	1	0	0	0	0	0

8	1	0	0	0	0	0
9	1	0	0	0	29.5	16.6
10	1	0	0	0	9.0	5.8
11	1	0	0	0	3.5	1.8
12	1	0	0	0	6.1	1.6
13	1	0	0	0	13.5	5.8
14	1	0	0	0	14.9	5.0

* Slack Bus

TABLE (I-F)
REGULATED BUS DATA (14 BUS SYSTEM)

Bus No.	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

iii) IEEE 30 BUS SYSTEM:

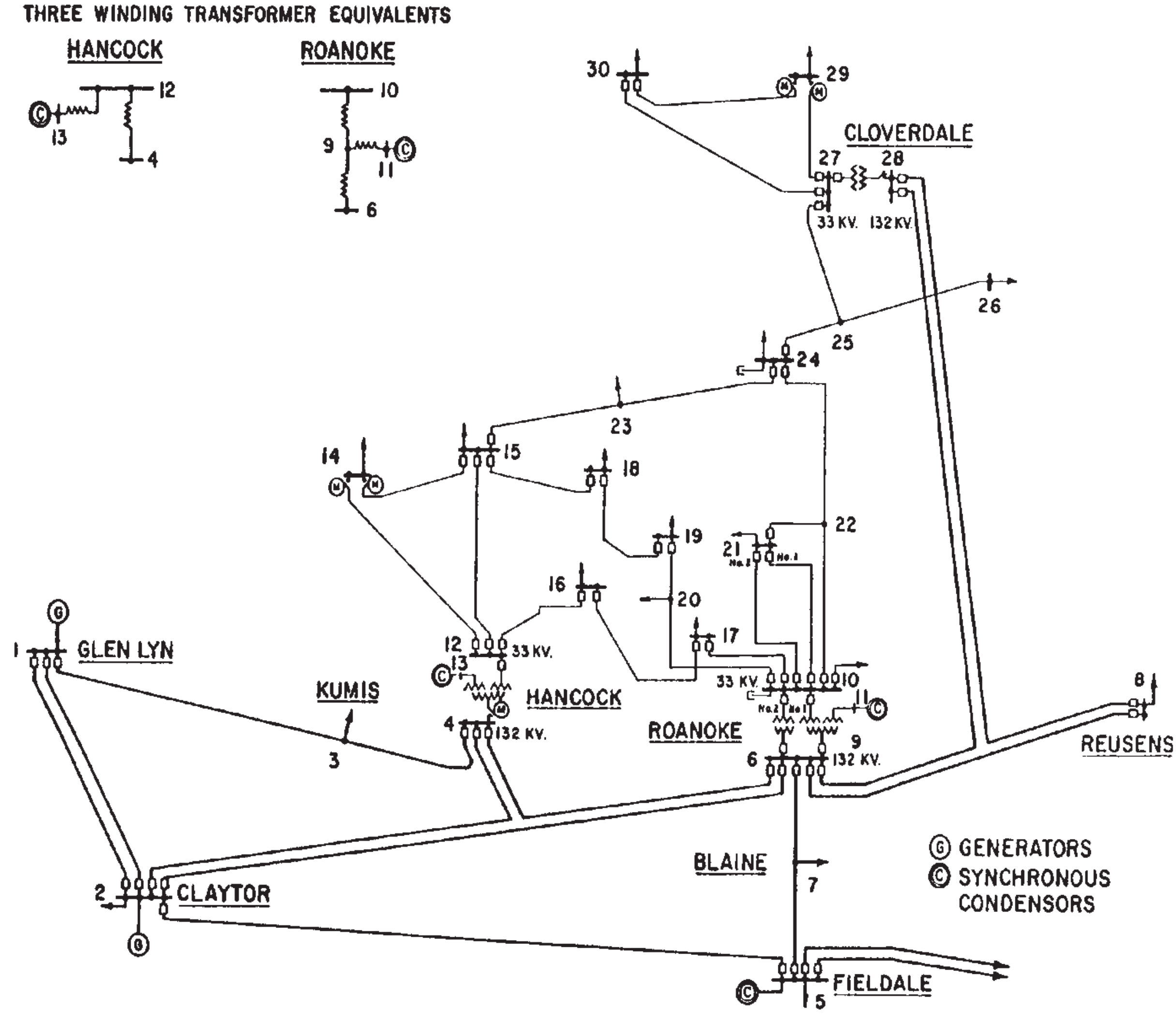

FIG (I-C): BUS-CODE DIAGRAM 30 BUS SYSTEM

TABLE (I-G)
IMPEDANCE AND LINE-CHARGING DATA (30 BUS SYSTEM)

Line Designation	Resistance p.u.*	Reactance p.u.*	Line Charging	Tap Setting
1-2	0.0192	0.0575	0.0264	1
1-3	0.0452	0.1852	0.0204	1
2-4	0.0570	0.1737	0.0184	1
3-4	0.0132	0.0379	0.0042	1
2-5	0.0472	0.1983	0.0209	1
2-6	0.0581	0.1763	0.0187	1
4-6	0.0119	0.0414	0.0045	1

5-7	0.0460	0.1160	0.0102	1
6-7	0.0267	0.0820	0.0085	1
6-8	0.0120	0.0420	0.0045	1
6-9	0	0.2080	0	0.978
6-10	0	0.5560	0	0.969
9-11	0	0.2080	0	1
9-10	0	0.1100	0	1
4-12	0	0.2560	0	0.932
12-13	0	0.1400	0	1
12-14	0.1231	0.2559	0	1
12-15	0.0662	0.1304	0	1
12-16	0.0945	0.1987	0	1
14-15	0.2210	0.1997	0	1
16-17	0.0824	0.1923	0	1
15-18	0.1070	0.2185	0	1
18-19	0.0639	0.1292	0	1
19-20	0.0340	0.0680	0	1
10-20	0.0936	0.2090	0	1
10-17	0.0324	0.0845	0	1
10-21	0.0348	0.0749	0	1
10-22	0.0727	0.1499	0	1
21-22	0.0116	0.0236	0	1
15-23	0.1000	0.2020	0	1
22-24	0.1150	0.1790	0	1
23-24	0.1320	0.2700	0	1
24-25	0.1885	0.3292	0	1
25-26	0.2544	0.3800	0	1
25-27	0.1093	0.2087	0	1

27-28	0	0.3960	0	0.968
27-29	0.2198	0.4153	0	1
27-30	0.3202	0.6027	0	1
29-30	0.2399	0.4533	0	1
8-28	0.0636	0.2000	0.0214	1
6-28	0.0169	0.0599	0.0065	1

*Impedance and line-charging susceptance in p.u. on a 100 MVA base. Line charging one-half of total charging line.

TABLE (I-H)
BUS DATA OR OPERATING CONDITIONS (30 BUS SYSTEM)

			Generation	Generation	Load	Load
Bus No.	Magnitude p.u.	Phase Angle Degrees	MW	MVAR	MW	MVAR
1*	1.06	0	0	0	0	0
2	1	0	40	0	21.7	12.7
3	1	0	0	0	2.4	1.2
4	1	0	0	0	7.6	1.6
5	1	0	0	0	94.2	19.0
6	1	0	0	0	0	0
7	1	0	0	0	22.8	10.9
8	1	0	0	0	30.0	30.0
9	1	0	0	0	0	0
10	1	0	0	0	5.8	2.0
11	1	0	0	0	0	0
12	1	0	0	0	11.2	7.5
13	1	0	0	0	0	0
14	1	0	0	0	6.2	1.6
15	1	0	0	0	8.2	2.5
16	1	0	0	0	3.5	1.8

17	1	0	0	0	9.0	5.8
18	1	0	0	0	3.2	0.9
19	1	0	0	0	9.5	3.4
20	1	0	0	0	2.2	0.7
21	1	0	0	0	17.5	11.2
22	1	0	0	0	0	0
23	1	0	0	0	3.2	1.6
24	1	0	0	0	8.7	6.7
25	1	0	0	0	0	0
26	1	0	0	0	3.5	2.3
27	1	0	0	0	0	0
28	1	0	0	0	0	0
29	1	0	0	0	2.4	0.9
30	1	0	0	0	10.6	1.9

* Slack Bus

TABLE (I-I)
REGULATED BUS DATA (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

TABLE (I-J)

TRANSFORMER DATA (30 BUS SYSTEM)

Transformer Designation	Tap Setting*
4-12	0.932
6-9	0.978
6-10	0.969
28-27	0.968

* Off-nominal turns ratio, as determined by the actual transformer-tap positions and the voltage bases. In the case of nominal turns ratio, this would equal 1.

TABLE (I-K)
STATIC CAPACITOR DATA (30 BUS SYSTEM)

Bus Number	Susceptance* p.u.
10	0.19
24	0.043

* Susceptance in p.u. on 100 MVA base.

APPENDIX II

MATLAB Program for Load flow using particle swarm optimization

```
clc
tic
nbus =30;                      % IEEE-5
Y = ybusppg(nbus);            % Calling ybusppg.m to get Y-Bus Matrix..
busd = busdatas(nbus);
BMva=100;                         % Calling busdatas..                  % Base
MVA..
Pg = busd(:,5)/BMva;          % gen  erated real power
Qg = busd(:,6)/BMva;          % generated reactive power.
Pl = busd(:,7)/BMva;          % load real power
Ql = busd(:,8)/BMva;          % load reactive power
Qlim1 = busd(:,9)/BMva;
Qlim2 = busd(:,10)/BMva;
P = Pg - Pl;                  % Pi = PGi - PLi..
Q = Qg - Ql;                  % Qi = QGi - QLi..
Psp = P        ;              % P Specified..
Qsp = Q        ;             % q specified
for i=[2,5,8,11,13]
Qmin(i)=Qlim1(i)-Ql(i);
Qmax(i)=Qlim2(i)-Ql(i);
end
%Qmin=Qlim1;
%Qmax=Qlim2;
G = real(Y)   ;         % Conductance matrix..
B = imag(Y)  ;          % Susceptance matrix..
kv=10;
%----------------------PSO PARAMETERS INITIALIZATION ---------------%
p=60;                    % no of particle
it=500;                  % no of iteration
rp=1;
T=2;                      %tolerance factor
rg=1;
rf=1;
f=zeros(p,it);
fp=zeros(p,1);
thp=zeros(p,nbus);
thg=zeros(p,nbus);
vp=zeros(p,nbus);
vg=zeros(p,nbus);
v=zeros(p,it,nbus);
th=zeros(p,it,nbus);
vth=zeros(p,it,nbus);
vv=zeros(p,it,nbus);
a=0.4;
b=-0.4;
vth(:,1,:)=a+(b-a)*rand(p,nbus); %initial velocity of theta vector%
a=0.5;
b=-0.5;
vv(:,1,:)=a+(b-a)*rand(p,nbus); %initial velocity of voltage vector%
vth(:,:,1)=0;
vv(:,:,1)=0;
a=-0.3;
b=0;
```

```
th(:,1,:)=a+(b-a)*rand(p,nbus);
a=.95;
b=1.1;
v(:,1,:)=a+(b-a)*rand(p,nbus);
% volage  asuumption
v(:,:,1)=1.060;
v(:,:,2)=1.045;
v(:,:,5)=1.010;
v(:,:,8)=1.010;
v(:,:,11)=1.082;
v(:,:,13)=1.071;
th(:,:,1)=0;
thp(:,1)=0;
thg(:,1)=0;
vp(:,:,1)=1.060;
vp(:,:,2)=1.045;
vp(:,:,5)=1.010;
vp(:,:,8)=1.010;
vp(:,:,11)=1.082;
vp(:,:,13)=1.071;
vg(:,:,1)=1.060;
vg(:,:,2)=1.045;
vg(:,:,5)=1.010;
vg(:,:,8)=1.010;
vg(:,:,11)=1.082;
vg(:,:,13)=1.071;
%----------------------initial value of objective function-------------%
% Calculate P and Q
PVIND=zeros(p,nbus);
MV=zeros(p,nbus);
 for j=1:p
 P = zeros(nbus,1);
 Q = zeros(nbus,1);
 MPS=zeros(p,1);
 MQS=zeros(p,1);
 MVS=zeros(p,1);

for i = 2:nbus
        for k = 1:nbus
            P(i)   =   P(i)   +   v(j,1,i)*   v(j,1,k)*(G(i,k)*cos(th(j,1,i)-
th(j,1,k)) + B(i,k)*sin(th(j,1,i)-th(j,1,k)));
        end
end

for i = 2:nbus
        for k = 1:nbus
            Q(i)   =   Q(i)   +   v(j,1,i)*   v(j,1,k)*(G(i,k)*sin(th(j,1,i)-
th(j,1,k)) - B(i,k)*cos(th(j,1,i)-th(j,1,k)));
        end

end

 % real power mismatch
  MP=P-Psp;
  MPS=MP.^2;
  %reactive power mismatch in regulated bus
  for jk=[2,5,8,11,13]
    if Q(jk)<Qmin(jk);
      Qsp(jk)=Qmin(jk);
      PVIND(j,jk)=1;
```

```
  else PVIND(j,jk)=0;
  end
      if Q(jk)>Qmax(jk);
          Qsp(jk)=Qmax(jk);
          PVIND(j,jk)=2;
  else PVIND(j,jk)=0;
      end
  end

  MQ=Q-Qsp;
  MQS=MQ.^2;

   %voltage mismatch
     if PVIND(j,2)==0;
      MV(j,2)=v(j,1,2)-1.045;
     end
       if PVIND(j,5)==0;
      MV(j,5)=v(j,1,5)-1.01;
       end
       if PVIND(j,8)==0;
      MV(j,8)=v(j,1,8)-1.01;
       end

       if PVIND(j,11)==0;
      MV(j,11)=v(j,1,11)-1.082;
       end
     if PVIND(j,13)==0;
      MV(j,13)=v(j,1,13)-1.071;
     end

     MVS=MV.^2;
     %objective function value
  f(j,1)=sum(MPS)+sum(MQS)+kv*sum(MVS(j,:));

 end

%Initial personal best values
for i=1:p
    for k=2:nbus
        thp(i,k)=th(i,1,k);
    end
    for k=2:nbus
        vp(i,k)=v(i,1,k);
    end
end
%for Initial Global best values updation
fmin=min(f(:,1));
for k=1:p
    if f(k,1)==fmin
        gb=k;
    else
    end
end
%Initial global best value

for k=1:p
    for j=2:nbus
        thg(k,j)=th(gb,1,j);
         end
       for j=2:nbus
```

```
        vg(k,j)=v(gb,1,j);
        end
end
fgm = min(f(:,1));
Q3=zeros(p,it,nbus);
for i=1:it
   %for inertia weight W
    wmax=.4;
    wmin=.3;
    w=wmax-((wmax-wmin)*i/it);
      %velocity update
      %position update
for j=1:p
        for k=2:nbus

        vth(j,(i+1),k)    =    w*vth(j,i,k)    +    rp*(thp(j,k)-th(j,i,k))    +
rg*(thg(j,k)-th(j,i,k));
        th(j,(i+1),k) = th(j,i,k) + rf*vth(j,(i+1),k);
        end


        for q=2:nbus
        vv(j,(i+1),q)  =  w*vv(j,i,q)  +  rp*(vp(j,q)-v(j,i,q))  +  rg*(vg(j,q)-
v(j,i,q));
        v(j,(i+1),q) = v(j,i,q) + rf*vv(j,(i+1),q);
        end

            if PVIND(j,2)==0
                v(j,(i+1),2)=1.045;
            end
            if PVIND(j,5)==0
                v(j,(i+1),5)=1.010;
            end
            if PVIND(j,8)==0
                v(j,(i+1),8)=1.010;
            end
            if PVIND(j,11)==0
                v(j,(i+1),11)=1.082;
            end
            if PVIND(j,13)==0
                v(j,(i+1),13)=1.071;
            end

end
%objective function value
for j=1:p
 P = zeros(nbus,1);
 Q = zeros(nbus,1);
 MPS=zeros(p,1);
 MQS=zeros(p,1);

     for m = 2:nbus
        for k = 1:nbus
            P(m)          =          P(m)          +          v(j,(i+1),m)*
v(j,(i+1),k)*(G(m,k)*cos(th(j,(i+1),m)-th(j,(i+1),k))                    +
B(m,k)*sin(th(j,(i+1),m)-th(j,(i+1),k)));

        end
     end
        for m = 2:nbus
```

```
        for k = 1:nbus
            Q(m)            =            Q(m)            +            v(j,(i+1),m)*
v(j,(i+1),k)*(G(m,k)*sin(th(j,(i+1),m)-th(j,(i+1),k))                            -
B(m,k)*cos(th(j,(i+1),m)-th(j,(i+1),k)));

        end
        end

   % real power mismatch
  MP=P-Psp;
  MPS=MP.^2;
  %reactive power mismatch in regulated bus
  for jk=[2,5,8,11,13]
      Q3(j,i,jk)=Q(jk);
    if Q(jk)<Qmin(jk);
      Qsp(jk)=Qmin(jk);
      Q3(j,i,jk)=Qsp(jk);
      PVIND(j,jk)=1;
  else PVIND(j,jk)=0;

  end
      if Q(jk)>Qmax(jk);
          Qsp(jk)=Qmax(jk);
          Q3(j,i,jk)=Qsp(jk);
          PVIND(j,jk)=2;
  else PVIND(j,jk)=0;

      end
  end

  MQ=Q-Qsp;
  MQS=MQ.^2;

   %voltage mismatch
     if PVIND(j,2)==0;
      MV(j,2)=v(j,i+1,2)-1.045;

     end
       if PVIND(j,5)==0;
      MV(j,5)=v(j,i+1,5)-1.01;

       end
       if PVIND(j,8)==0;
      MV(j,8)=v(j,i+1,8)-1.01;

       end
     if PVIND(j,11)==0;
      MV(j,11)=v(j,i+1,11)-1.082;

     end
      if PVIND(j,13)==0;
      MV(j,13)=v(j,i+1,13)-1.071;

     end
     MVS=MV.^2;

  %objective function value
  f(j,(i+1))=sum(MPS)+sum(MQS)+kv*sum(MVS(j,:));
 end
```

```
          %personal best values updation
for j=1:p
 P = zeros(nbus,1);
 Q = zeros(nbus,1);
 MPS=zeros(p,1);
 MQS=zeros(p,1);
        for t =2:nbus
        for k = 1:nbus
            P(t) = P(t) + vp(j,t)* vp(j,k)*(G(t,k)*cos(thp(j,t)-thp(j,k)) +
B(t,k)*sin(thp(j,t)-thp(j,k)));

        end
        end
        for t = 2:nbus
        for k = 1:nbus
            Q(t) = Q(t) + vp(j,t)* vp(j,k)*(G(t,k)*sin(thp(j,t)-thp(j,k)) -
B(t,k)*cos(thp(j,t)-thp(j,k)));

        end
        end

  % real power mismatch
  MP=P-Psp;
  MPS=MP.^2;
  %reactive power mismatch in regulated bus
  for jk=[2,5,8,11,13]
       if Q(jk)<Qmin(jk);
      Qsp(jk)=Qmin(jk);
      PVIND(j,jk)=1;
  else PVIND(j,jk)=0;
      end
      if Q(jk)>Qmax(jk);
          Qsp(jk)=Qmax(jk);
          PVIND(j,jk)=2;
  else PVIND(j,jk)=0;
      end
  end

  MQ=Q-Qsp;
  MQS=MQ.^2;

   %voltage mismatch
     if PVIND(j,2)==0;
      MV(j,2)=v(j,i+1,2)-1.045;

     end
       if PVIND(j,5)==0;
      MV(j,5)=v(j,i+1,5)-1.01;

       end
       if PVIND(j,8)==0;
      MV(j,8)=v(j,i+1,8)-1.01;

       end
     if PVIND(j,11)==0;
      MV(j,11)=v(j,i+1,11)-1.082;

     end
      if PVIND(j,13)==0;
```

```
     MV(j,13)=v(j,i+1,13)-1.071;

    end
    MVS=MV.^2;

 %objective function value

 fp(j)=sum(MPS)+sum(MQS)+kv*sum(MVS(j,:));
end
%personal best value updation
        for k=1:p
            for m=2:nbus
               if f(k,i)<fp(k)
               thp(k,m)=th(k,i,m);
             else
            end

        end

      end

        for k=1:p
             for m=2:nbus
             if f(k,i)<fp(k)
             else
             vp(k,m)=v(k,i,m);
            end

        end
      end

  %for Global best values updation
   if min(f(:,(i+1)))<fgm
       fgm=min(f(:,(i+1)));
   else
   end

 for m=2:nbus

        for k=1:p
        if f(k,i)==fgm
            for l=1:p
                thg(l,m) = th(k,i,m);      %global best values
               end
        else
      end

  end
 end
      for m=2:nbus

        for k=1:p
        if f(k,i)==fgm
            for l=1:p
                  vg(l,m) = v(k,i,m);           %global best values
            end
        else
        end
```

```
   end
       end
%stopping criteria
    gb1=gb;
for k=1:p
    if f(k,i+1)==fgm
        gb1=k;
    else
    end
end
   if (f(gb1,i+1)-f(gb1,i)<=(10)^(-T))
            break


    end
 %f(gb1,i)
end
i
vth(gb1,(i+1),2);
vv(gb1,(i+1),2) ;
bus=zeros(nbus,10) ;
for k3=1:nbus
    bus(k3,3)=v(gb1,i,k3);
end
%bus angle updation
for k3=1:nbus
    bus(k3,4)=th(gb1,i,k3);
end
for k3=[2,5,8,11,13]
    bus(k3,6)=Q3(gb1,i,k3)*BMva;
end
V = bus(:,3) ;            % Specified Voltage..
x=bus(:,6);
del = bus(:,4);   % Voltage Angle..
toc
%load flow function calling
loadflow(nbus,V,del,BMva,x);
```