PARTICLE SWARM OPTIMIZATION ALGORITHMS FOR ECONOMIC LOAD DISPATCH PROBLEM

[1]OM PRAKASH, [2]DR. BHAVIN SEDANI

[1]Assistant Professor, ECE Department, JJT University, Jhunjhunu, India.
[2]Head & Associate Professor, Electronics and Communication Engineering Department, V.V.P. Engineering College, Rajkot-360005

om4096@gmail.com,bhavin_s_sedani@yahoo.com

ABSTRACT: *The aim of this paper is to give fundamental insight into the particle swarm optimization algorithm. The applicability of this method is demonstrated by solving different benchmark test problems like constraint and unconstraint optimization problems. Economic load dispatch is an important optimization task in power system operation for generation allocation among the committed units with the objective of dividing the power economically, whereas satisfying various system constraints. The application of particle swarm optimization approach is also presented to solve economic load dispatch problem of electric power system in this work.*

KEYWORDS: Particle swarm optimization, constraint optimization problems, unconstraint optimization problem and economic load dispatch problem.

Introduction

The last three decades have witnessed the development in efficient and effective stochastic optimizations. In contrast to the traditional adaptive stochastic search algorithms, evolutionary computation (EC) techniques exploit a set of potential solutions, namely a population, and detect the optimal solution through cooperation and competition among the individuals of the population. These techniques often detect optima in difficult optimization problems faster than traditional methods [7]. One of the most powerful swarm intelligence-based optimization techniques, named PSO, was introduced by Kennedy and Eberhart [3, 5]. PSO is inspired by the swarming behavior of animals, and human social behavior. During the last decade many studies focused on this method and almost all of them, strongly confirmed the abilities of this newly proposed optimization technique [3, 5, 9, 14], e.g. fast convergence, finding global optimum in presence of several local optima, simple programming and adaptability with constrained problems. Some author attempted to enhance the algorithm by developing new variations such as variable inertia coefficient, constriction factor [9], maximum velocity limit, parallel optimization, deflection, repulsion, stretching [5], mutation [14,16] etc. Particle swarm optimization was invented by Russ Eberhart and James Kennedy in 1995 through simplifying a social simulation model which was originally developed to simulate the process of birds seeking food. The PSO algorithm is a population-based evolutionary algorithm. Like other evolutionary algorithms, each individual (called particle in PSO) in the population represents a candidate solution to the problem to be solved. Unlike other evolutionary algorithms each individual/particle has a velocity parameter associated with it in addition to its position parameter in the solution space, which is the only parameter that an individual in other evolutionary algorithms has. Each particle "flies" through the solution space with a velocity which is dynamically changed according to its own flying experience and its companion's flying experience. It is this velocity changing rule through which all the particles communicate and share information among themselves. Furthermore, it is this sharing and communicating mechanism that enables particles to fly towards better and better search areas while at the same time to risk to be stuck into local minima.

The search process or flying trajectories of particles are complicated and nonlinear. To search for good enough solutions, especially for the multi-modal optimization problems, the search process needs to have the ability to converge at some time while diverging at other times in order to have the ability to find good enough solutions and to be able to avoid to be stuck in un-wanted local minima. Therefore, it is critical to have a capability to monitor the search process of PSO in order to first understand the PSO search process and then design a better algorithm or even have possibilities to control the search process later.

A straightforward approach to measure the diversity of PSO is to use the standard deviation of the fitness values of all the population particles. Population fitness values are attributes of the PSO behaviors and not the PSO particles themselves directly. Therefore, this kind of diversity measurement is simple but it is an indirect measurement of the population diversity. The diversity of PSO has been looked at from different perspectives. Each particle in a PSO has an *n*-dimensional velocity associated with it in addition to its position as in other evolutionary algorithms. Therefore, diversities depend on particles' positions and velocities instead of only the position diversity as in other evolutionary algorithms. Velocity diversity has velocity speed diversity and velocity directional diversity. The velocity speed tells how fast a particle is flying and the velocity direction tells where a particle is flying towards [12].

Particle swarm optimization algorithms

The Particle swarm optimization algorithm is an optimization and search technique based on the principles of social behavior of animals. The method was developed in 1995 by James Kennedy and Russell Eberhart. PSO mimics the collective intelligent behavior of "unintelligent" creatures. PSO is very good at finding good enough solutions for a large range of problems, such as constrained optimization problems, multi-objective optimization problems, etc. The original PSO algorithm is very simple in concept and easy in implementation.

Initialization

The initial swarm is generally created with all particles randomly distributed throughout the design space, each with a random initial velocity vector. Eq. (1) is used for obtaining the random initial position and Eq. (2) for velocity vector, it can be formulated as:

$$x_{id}^{(0)} = x_i^{min} + r_1\left(x_i^{max} - x_i^{min}\right);\ i = 1,2,\dots,n;\ d = 1,2,\dots,m \quad (1)$$

$$v_{id}^{(0)} = \frac{x_i^{min} + r_2\left(x_i^{max} - x_i^{min}\right)}{\Delta t};\quad i = 1,2,\dots,n;\ d = 1,2,\dots,m. \quad (2)$$

where,

- $x_{id}^{(0)}$ represents the *d*th position value of the *i*th particle at time step is zero.
- $v_{id}^{(0)}$ represents the rate of the *d*th position value change (velocity) for particle *i*at time step is zero.
- $r_1$and r_2 are random number within the range of [0,1].
- $x_i^{min}$is lower bound of the position.
- $x_i^{max}$is upper bound of the position.
- Δt is step size.
- *n*is the number of variables.
- *m* is the size of the swarm (number of particle in swarm).

Parameters of particle swarm optimization are also initializing.c_1 and c_2 are positive constants, known as thrust parameter. Generalized value of c_1 and c_2 is 2 and $c_1 + c_2 \leq 4$. $w$is the inertia of the particles. Upper and lower bounds are usually specified on v_i to avoid too rapid movement of particles in the search space; that is, the various range of the *dth*velocity is [-Vmax, Vmax]. In this thesis upper and lower bonds are formulated as:

$$v_i^{min} = -\alpha\, x_i^{min};\ \ i = 1,2,\dots,n. \quad (3a)$$

$$v_i^{max} = \alpha\, x_i^{max};\ i = 1,2,\dots,n. \quad (3b)$$

where,

- v_i^{min}lower bond of velocity.
- v_i^{min}upper bond of velocity.
- $\alpha$is the arbitrary constant with in the range [0,1], generally taken as 0.5 in this thesis. For x_i^{min}, if x_i^{min} is positive then α is positive and vice versa. For x_i^{max} α is always positive.
- $x_i^{min}$is lower bound of the position.
- $x_i^{max}$is upper bound of the position.

Updating position and velocity

New velocity and Position can be updated using Eqn. (4) and Eqn. (5) respectively. These equations are formulated as [11]:

$$v_{id}^{(t+1)} = wv_{id}^{t} + c_1r_3\left(x_i^{lb} - x_{id}^{t}\right) + c_2r_4(x_i^{gb} - x_{id}^{t};i = ,2,\dots n\ ;d = 1,2,\dots m.(4)$$

where

- c_1 and c_2 are positive constants,
- $r_3$and r_4 are random number within the range of [0,1].
- $w$is the inertia of the particles.
- $x_{id}^{(t)}$represents the *d*th position value of the *i*th particle at time step *t.*
- x_i^{lb}represents the *d*th position value of the best previous position (the position giving the best fitness value) of the *i*th particle at the time step *t.*
- x_i^{gb}represents the index of the best particle among all the particles.
- $v_{id}^{(t)}$represents the rate of the *d*th position value change of the *i*th particle at time step *t*.

- $v_{id}^{(t+1)}$ represents the rate of the *d*th position value change (velocity) for particle *i*at time step *t+1*

$$x_{id}^{(t+1)} = x_{id}^{t} + v_{id}^{(t+1)}, i = 1,2, \dots n; d = 1,2, \dots m \qquad (5)$$

where,

- $x_{id}^{(t)}$represents the *d*th position value of the *i*th particle at time step *t.*
- $x_{id}^{(t+1)}$represents the *d*th position value of the *i*th particle at time step *t+1.*
- $v_{id}^{(t+1)}$represents the rate of the *d*th position value change (velocity) for particle *i*at time step *t+1.*
- *n*is the number of variables.
- *m* is the size of the swarm (number of particle in swarm).

Usually, all the w will have the same value for simplicity but the inertia weight can be dynamically adjusted according to the current and historical performance of the particles, which will improve the PSO's performance since the search process of a PSO algorithm is nonlinear and complicated. A simple and straightforward approach is to linearly decrease inertia weigh over the course of PSO. Other PSO parameters can be fixed and/or even can be dynamically changed to affect the search process in the hope of having a more diverse or better performed PSO particles.

Equation (4) and Eqn. (5) are the equations governing the flying trajectory of particles and tells change of the velocity. In order not to violate the physical law, the velocity cannot be changed abruptly and shall be changed from the current velocity, which is reflected by the first part of the Eq. (4) as a "flying" particle's momentum. The other two parts of the Eq. (4) reflect the learning and collaboration capability of a particle. The second part reflects a particle's self-learning capability or self-cognition, that is, a particle learns from its own flying experience. The third part reflects particle's collaboration capability, that is, a particle learns from "flying" experience of its neighboring particles. The position of a "flying" particle is adjusted according to the Eq. (5).

There are two most commonly used versions of PSOs, global version and local version. In a global version PSO, a single and unique global best is shared by all particles in the whole population. In a local version PSO, each particle in the population may have different global best which is the best performed particle within the particle's own neighborhood. In both global and local version PSO, particles fly through the search space with dynamically changed velocities according to the Eq. (4). The neighborhood of each particle is generally defined as its topologically nearest particles at each side instead of Euclidean neighborhood. The global version PSO can be considered as a special case of a local version PSO if the whole population is considered as each particle's neighborhood. It has been claimed that the global version PSO converges fast, but with potential to converge to the local minimum, while the local version PSO might have more chances to find better solutions slowly.

Inertia weight

The inertia weight, w, controls the momentum of the particle by weighing the contribution of the previous velocity–basically controlling how much memory of the previous flight direction will influence the new velocity. The equation of inertia weight is formulated as

$$W = \frac{(w^{max} - w^{min}) * IT}{IT^{max}} \qquad (6)$$

where

- wis the inertia of the particle.
- w^{max}represents upper limit of inertia weight.
- w^{min}represents lower limit of inertia weight.
- ITrepresents current number of iteration.
- IT^{max}represents maximum number of iteration.

The inertia weight *w* is in the range [0.4, 0.9] and declines linearly in iteration as described in equation (6). For w > 1, velocities increase over time causing divergent behavior. Particles fail to change direction in order to move back towards promising areas. For w < 0, particles decelerate until their velocities reach zero.

Convergence criterion

Changes in the objective function are monitored for a specified number of consecutive design iteration. If the maximum change in the objective function is less than a predefined allowable change, convergence is assumed.

Optimization problem formulation

Economic load dispatch (ELD) is an important topic in the operation of power plants which helps to build up effective generating management plans. The ELD problem has non-smooth cost function with equality and inequality constraints which make it difficult to be effectively solved [6]. Real cost functions are more complex than conventional second order cost functions when multi-fuel operations, valve-point effects, accurate curve fitting, etc., are considering in deregulated changing market [13].

The ELD problem may be expressed by minimizing the fuel cost of generator units under constraints. Depending on load variations, the output of generators has to be changed to meet the balance between loads and generation of a power system. The power system model consists of *n* generating units already connected to the system [4]. The fuel-cost function without valve-point loadings of the generating units is given by

$$F(x_i) = a_i x_i^2 + b_i x_i + c_i \quad Rs/h \qquad (7)$$

A cost function is obtained based on the ripple curve for more accurate modeling which contains higher order nonlinearity and discontinuity due to the valve point effect and should be refined by a sine function. The ELD problem can be expressed as [1, 2, 15]:

Minimize operation cost;

$$F = \sum_{i=1}^{n} (c_i + b_i x_i + a_i x_i^2) + \left| e_i \sin\left(f_i\left(x_i^{min} - x_i\right)\right)\right| {}^{Rs}/_{h} \qquad (8)$$

Subjects to;

i. Power balance constraints

$$\sum_{i=1}^{n} x_i - P_D - P_L = 0 \qquad (8a)$$

and,

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i B_{ij} x_j + \sum_{i=1}^{n} B_{oi} x_i + B_{oo} \qquad (8b)$$

ii. ii. Generating capacity constraints

$$x_i^{min} \le x_i \le x_i^{max};\ i = 1,2,\dots,n \qquad (8c)$$

where,

- a_i, b_i and c_i are the cost coefficients of the i^{th} generator
- n is the number of generators
- x_i is the real power output of the i^{th} generator (MW)
- $F(x_i)$ is the operating cost of unit i (Rs/h)
- P_L is the transmission losses (MW)
- x_i^{max} is the maximum generation output of the i^{th}generator
- x_i^{min} is the minimum generation output of the i^{th} generator
- B_{ij}, B_{oi}andB_{oo} are the B-coefficients
- $P_D$is the total demand (MW).

The key factor in solving an ELD problem is how to handle the several constraints relating to the problem. Over the last few decades, kinds of approaches had been proposed to handle the constraints. These can be grouped into four categories: ideas that preserve the feasibility of solutions, penalty-based approaches, methods that clearly distinguish between feasible and unfeasible solutions, and hybrid techniques [9]. In this thesis the penalty function is adopted to address the constraints in an ELD problem. The introduction of the penalty term enables to transform a constrained optimization problem into an unconstrained one. As a result, the fuel cost function is written as:

$$F_m(x_i, r_k) = \sum_{i=1}^{n} F_i(x_i) + r_k \,.\, h^2 \qquad (9)$$

where,

The value of the penalty coefficient r_k is checked at each iteration, and h is the equality constrained defined as

$$h = \sum_{i=1}^{n} x_i - P_D - P_L \qquad (10)$$

Most of these methods were based on penalty formulations that transform Eq. (8b) into an unconstrained function $F_m(x_i, r_k)$ as shown in Eq. (9), which consisting of a sum of the objective and the constraints weighted by penalties, and use PSO to minimize$F_m(x_i, r_k)$.

Algorithm

The step-wise procedure of the Economic load dispatch using Particle swarm optimization algorithm is outlined below:

1. *Initialization:* Generate swarm size; initialize initial position and velocity vector of the particle. Equations for position and velocity vector are formulated as:

$$x_{id}^{(0)} = x_i^{min} + r_1\left(x_i^{max} - x_i^{min}\right); \quad i = 1,2,\dots,n; d = 1,2,\dots,m;$$

$$v_{id}^{(0)} = \frac{x_i^{min} + r_2\left(x_i^{max} - x_i^{min}\right)}{\Delta t}; \quad i = 1,2,\dots,n;\ d = 1,2,\dots,m$$

Initial parameters are also initialized.

2. *Swarm's manipulation:* The particles, except the best of them regulate their velocities in accordance with the equation

$$v_{id}^{(t+1)} = wv_{id}^{(t)} + c_1 r_3 \left(x_i^{lb} - x_{id}^{(t)}\right) + c_2 r_4 (x_i^{gb} - x_{id}^{(t)});$$

$$i = 1,2,\dots,n\,;\, d = 1,2,\dots,m$$

3. *Best particle's manipulation:* The best particle in the swarm updates its velocity using a random coordinator calculated between its position and the position of a randomly chosen particle in the swarm.

4. *Velocity bounds' oscillations:*Check if the bounds of velocities are enforced, if the bounds are violated then they are replaced by the respective limits.

5. *Position update:* The positions of particles are updated using equation

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)}, i = 1,2,\dots,n; d = 1,2,\dots,m$$

6. *Position limits:*Check if the limits of particles' positions (generators prohibited operating zones) are enforced

$$x_i^{min} \le x_i \le x_i^{max};\ i = 1,2,\dots,n \quad .$$

7. *Evaluation:* Objective function is evaluated to minimize fuel cost. Objective function is formulated as:

$$F_m(x_i, r_k) = \sum_{i=1}^{n} F_i(x_i) + r_k\,.\,h^2$$

8. *Update search intervals and swarm size:* If there is no improvement in the global best then swarm size is increased and search intervals are updated.

9. *Stopping criteria:* The PSO algorithm will be terminated if the maximum number of allowed iterations is achieved.

10. *Global optimal solution:* Choose the optimal solution as the global best achievement.

Results and discussion

The PSO algorithm is developed in Fortran 77, objects oriented language and has been successfully tested on benchmark test problems, which involve constraints on state variables and control variables. The obtained results are widely affected by the search area boundaries, number of iteration and tuning of parameter. The different benchmark test examples undertaken for study are presented in the ensuing section.

Unconstraint optimization problem

To evaluate the performance of the PSO algorithm, a series of experiments are conducted on the well known benchmark functions. All benchmark functions used for experiments are summarized here [8, 10]:

Test Problem 1 (Sphere). This problem is defined by

$$\mathrm{f}(x) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x}_\mathrm{i}^2$$

where*n* is the dimension of the problem. The global minimizer is $x* = (0, \dots ,0)$ with $f(x*) = 0$.

Test Problem 2 (Rastrigin). This problem is defined by

$$\mathrm{f}(x) = 10n + \sum_{i=1}^{\mathrm{n}} (\mathrm{x}_\mathrm{i}^2 - 10\cos(2\pi \mathrm{x}_\mathrm{i}))$$

where*n* is the dimension of the problem. The global minimizer is $x* = (0, \dots ,0)$ with $f(x*) = 0$.

Test Problem 3(Generalized Rosenbrock). This problem is defined by

$$\mathrm{f}(x) = \sum_{i=1}^{n-1} 100\Big((\mathrm{x}_{\mathrm{i}+1} - \mathrm{x}_\mathrm{i}^2)^2 + (x_i - 1)^2\Big)$$

where*n* is the dimension of the problem. The global minimizer is $x* = (1, \dots ,1)$ with $f(x*) = 0$.

Test Problem 4(A quadratic function). This problem is defined by

$$\mathrm{f}(x_1, x_2) = (\mathrm{x}_1 + 2\mathrm{x}_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

The global minimizer is $x* = (1,3)$ with $f(x*) = 0$.

Test Problem 5 (Beal's function). This problem is defined by

$$f(x_1, x_2) = [1.5 - \mathrm{x}_1(1 - \mathrm{x}_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2$$
$$+[2.625 - x_1(1 - x_2^3)]^2$$

The global minimizer is $x* = (3, 0.5)$ with $f(x*) = 0$.

Particle swarm optimization is used to find the control variables to minimize the function. For each test problem swarm size is taken as 50, dimension *n* is taken as 2 and $c_1 = c_2 = 2$; w is varied from 0.9 to 0.4. error (ε) is taken according to problem shown in Table 1. The results obtained for different test problem are given in Table 2. The effect of iterations while obtaining the result has been realized through Fig. 1 to Fig. 5.

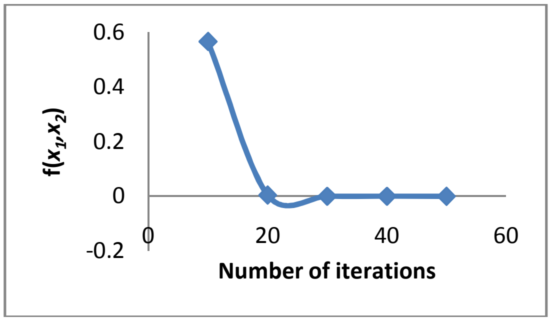


Fig. 1: Variation of function with respect to iterations for test problem 1

Table 1: Parameters for the unconstrained optimization problems

S. No	Problem	(x_i^{min}, x_i^{max})	ε
1	TP1	(-100,100)	10^{-6}
2	TP2	(-30,30)	10^{-6}
3	TP3	(-5.12,5.12)	10^{-7}
4	TP4	(-30,30)	10^{-6}
5	TP5	(-15,15)	10^{-5}

Table 2: Results of PSOA for the unconstrained problems

S. No.	Problem	Ideal Result			Obtained Result		
		x_1	x_2	$f(x_1,x_2)$	x_1	x_2	$f(x_1,x_2)$
1	TP1	0	0	0.00	0.0011	0.0010	$2.26*10^{-6}$
2	TP2	0	0	0.00	0.000138	-0.000014	$3.81*10^{-6}$
3	TP3	1	1	0.00	0.995800	0.992083	0.000013
4	TP4	1	3	0.00	0.752117	3.4999086	0.00001
5	TP5	3	0.5	0.00	2.990513	0.496756	0.00003

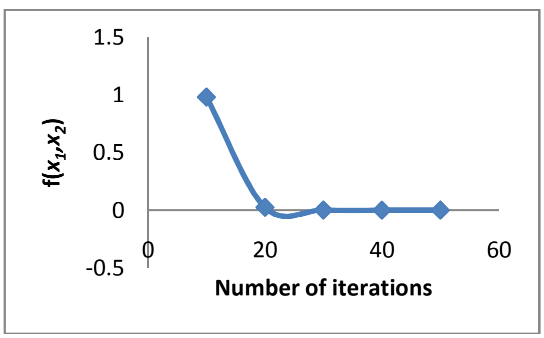


Fig. 2: Variation of function with respect to iterations for test problem 2

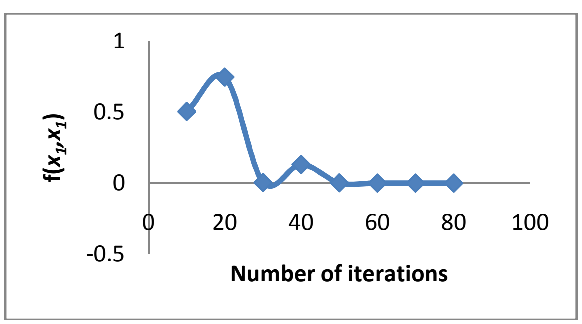


Fig. 3: Variation of function with respect to iterations for test problem 3

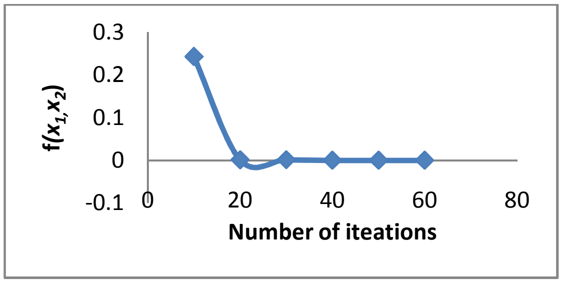


Fig. 4: Variation of function with respect to iterations for test problem 4

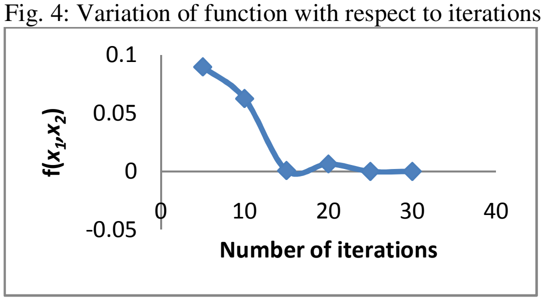


Fig. 5: Variation of function with respect to iterations for test problem 5

Equality constraints optimization problem

Applicability of PSO has been tested by considering a problem with one equality constraint and with some limits on the values of variables, x_1 and x_2. A control problem with given state boundary conditions and equality constraint, is stated below:

Minimize $f(x_1,x_2) = x_1^2-(x_2-1)^2$ (11a)

Subject to $h(x_1,x_2) = x_2 - x_1^2 = 0$, (11b)

$-1< x_i<1$; $i = 1, 2$ (11c)

$X = [x_1, x_2]$

Augmented penalty function is written as:

$A(x_1,x_2,r_k) = x_1^2 - (x_2-1)^2 + r_k(x_2 - x_1^2)^2$ (12a)

$-1< x_i<1$; $i = 1, 2$ (12b)

Particle swarm optimization is applied to find the control variables to minimize the function. Swarm size is taken as 30, plenty parameter is set to 100 and error goal is 10^{-5} for this problem. The results obtained for different number of iteration are given in Table 3.

Table 3: Results of equality constraint optimization problem

S. No.	Iterations	x_1	x_2	$f(x_1,x_2)$
1	5	0.384777	0.161084	0.86881
2	10	0.549719	0.317520	0.79147
3	15	0.711679	0.511302	0.74763
4	20	0.690876	0.482266	0.74781
5	25	0.709950	0.509364	0.74760

The best solution is obtained. The obtained variables are $x_1 = 0.709950$, $x_2 = 0.509364$ and the objective function is $f(x_1,x_2) = 0.74760$. The effect of iterations while obtaining the result has been realized through Fig. 6. It is concluded that smaller number of iteration terminates prematurely.

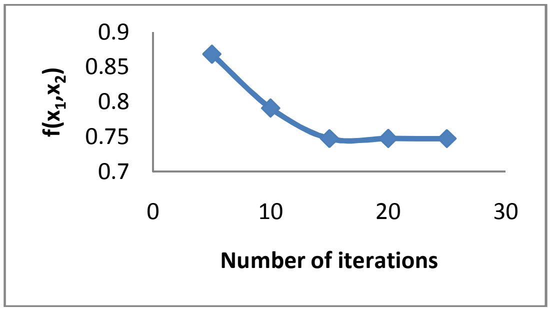


Fig. 6: Variation of function with respect to iterations for equality constraint optimization problem

Inequality constraints optimization problem

Another test problem is defined below is solved by Particle swarm optimization which have two inequality constraints.

Maximize f (x_1,x_2) = 170-14x_1-.22x_2 (13a)

Subject to g_1 (x_1,x_2) =20-4x_1- x_2 ≥ 0 (13b)

$g_2(x_1,x_2) = 73 - 2x_1 - 12x_2 \geq 0$, (13c)

$0 \leq x_i \leq 6$;i =1, 2 (13d)

$X = [x_1, x_2]^T$

Augmented penalty function is written as:

$A(x_1, x_2, r_k) = (170 - 14x_1 - 22x_2) + r_k < 20 - 4x_1 - x_2 >^2$
$+r_k < 73 - 2x_1 - 12x_2 >^2$ (14a)

$$< 20 - 4x_1 - x_2 >= \begin{cases} 20 - 4x_1 - x_2 & if\ 20 - 4x_1 - x_2 < 0 \\ 0 & if\ 20 - 4x_1 - x_2 \geq 0 \end{cases} \quad (14b)$$

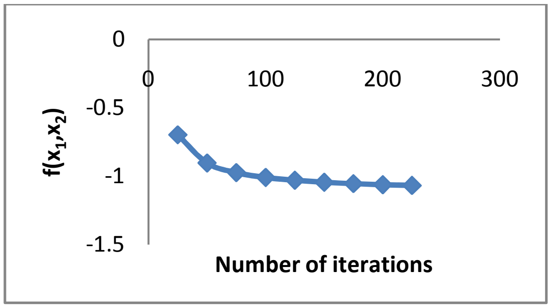
$$< 73 - 2x_1 - 12x_2 >= \begin{cases} 73 - 2x_1 - 12x_2 & if\ 73 - 2x_1 - 12x_2 < 0 \\ 0 & if\ 73 - 2x_1 - 12x_2 \geq 0 \end{cases} \quad (14c)$$

The results obtained are given in Table 4. The variation of objective function with respect to iterations is given in Fig.7. Higher number of iterationsgives better result.

Table 4: Results of inequality constraint optimization problem

S. No	Iteration	x_1	x_2	$f(x_1, x_2)$
1	25	3.607857	5.463054	-0.69719
2	50	3.622584	5.463054	-0.90337
3	75	3.627593	5.463054	-0.97349
4	100	3.630115	5.463054	-1.00881
5	125	3.631635	5.463054	-1.03008
6	150	3.632650	5.463054	-1.04430
7	175	3.633377	5.463054	-1.05447
8	200	3.633923	5.463054	-1.06211
9	225	3.634347	5.463054	-1.06804

The best solution is obtained by using particle swarm optimization. The obtained variable are x_1= 3.634347, x_2= 5.463054 and $f(x_1, x_2)$ = -1.06804. The effect of number of iterations while obtaining the result has been realized through Fig. 7. Swarm size is taken as 200, plenty parameter is set to 100 and error goal is 10^{-5} for this Problem



.
Fig. 7: Variation of function with respect to iterations for inequality constraint optimization problem

Economic load dispatch problem

The economic load dispatch problem can be described as an optimization (minimization) process with the objective:

$$Minimized f(X) = \sum_{i=1}^{n} F_i(x_i) \qquad (15a)$$

Subject to:

(i) Power balance constraints:

(ii)

$$P_D = \sum_{i=1}^{n} x_i \qquad (15b)$$

In this thesis, the transmission losses are disregarded.

(iii) Generating capacity constraints:

$x_i^{min} \leq x_i \leq x_i^{max}$;i = 1, 2, …, n (15c)

where

$F_i(x_i)$ is the fuel cost function of the i^{th}unit

P_D is the system load demand and

x_i^{min} and x_i^{max} are the minimum and maximum power outputs of the i^{th}unit.

Augmented penalty function is written as:

$$A(x_i, r_k) = \sum_{i=1}^{n} F_i(x_i) + r_k\left(P_D - \sum_{i=1}^{n} x_i\right)$$

In this section, the PSO has been tested on two sample electric power systems consisting 3 and 13-generators. The iteration was varied and the penalty multiplier r_k is taken as 100 for all selected methods.

3- Generators

Electric power system of three generators with valve-point loading effect has been studied. In this case, the load demand, P_D is taken as 850MW. Operating cost coefficients for each generator are given in Table 5,

Table 5: Fuel-cost coefficients: 3-generators.

Generator number	**Generator limits**		**Fuel cost coefficients**				
	x_i^{min} (MW)	x_i^{max} (MW)	a_i (Rs/h)	b_i (Rs/MWh)	c_i (Rs/MW²h)	e_i (Rs/h)	f_i (MW^{-1})
1	100	600	0.001562	7.92	561	300	0.0315
2	50	200	0.004820	7.97	78	150	0.063
3	100	400	0.001940	7.85	310	200	0.042

Swarm size is taken as 50, plenty parameter is set to 100 and error is 10^{-5} for this problem, c_1 and c_2 are taken as 2. The results are represented in Table 6, which show that the Particle swarm optimization succeeded in finding the satisfactory solution. Figure 8 represents variation of function with respect to iterations.

Table 6: Results of economic load dispatch problem for 3-generators

S. No.	**Iterations**	***f*(X)**
1	25	8315.885
2	50	8316.604
3	75	8436.338
4	100	8234.867
5	125	8234.185
6	150	8234.867

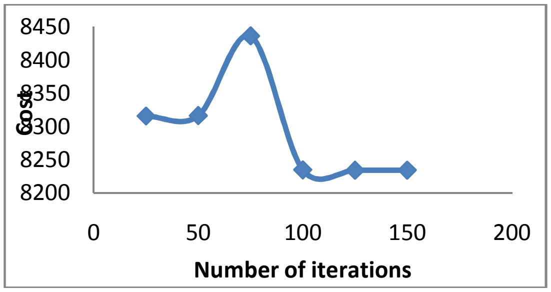


Fig. 8: Variation of cost with respect to number of iterations for 3-generator power system.

13- Generators:

A 13-generators electric power system considering valve-point loading effect has been studied in this test. In this case, the load demand, P_D is taken as 1800MW. 13- generators result are given in Tables 8. Although the acquired best solution is not guaranteed to be the global solution, the Particle swarm optimization succeeded in finding the satisfactory solution. The respective operating cost coefficients for each generator are given in Table 7.

Table 7: Fuel-cost coefficients: 13-generators**.**

Generator number	**Generator limits**		**Fuel cost coefficients**				
	x_i^{min} **(MW)**	x_i^{max} **(MW)**	a_i **(Rs/h)**	b_i **(Rs/MWh)**	c_i **(Rs/MW²h)**	e_i **(Rs/h)**	f_i **(MW^{-1})**
1	00	680	0.00028	8.10	550	300	0.035
2	00	360	0.00056	8.10	309	200	0.042
3	00	360	0.00056	8.10	307	200	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

Swarm size is taken as 200, plenty parameter is set to 100, error is 10^{-4} for this problem, c_1 and c_2 are taken as 2. The obtained results are presented Table 9. Economic load dispatch problem for 13-generators data is graphically shown in Figure 9.

Table 8: Results of economic load dispatch problem for 13-generators

S. No.	**Iterations**	***f*(X)**
1	50	18581.34
2	100 100	18655.89
3	150	18633.17
4	200	18574.99
5	250	18641.51

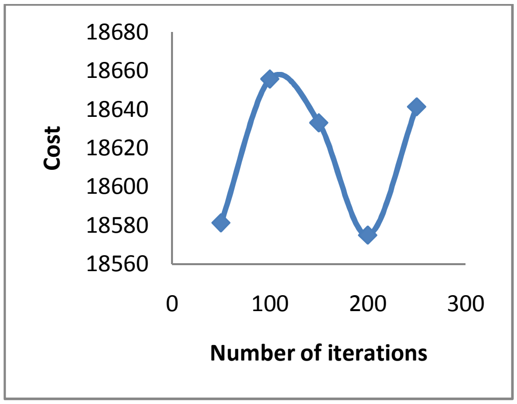


Fig. 9: Variation of cost with respect number of iterations for 13-generator power System.

CONCLUSION AND FUTURE SCOPE

PSO greatly enhances the searching ability and efficiently manages the system constraints. It makes problem easier because the probability of finding a solution by chance is large. This work presents an application of Particle swarm optimization based algorithm to solve the unconstrained and constrained optimization problems. An algorithm in Fortran 77 object oriented programming language has been developed for the solution of ELD problem, in present work the economic dispatch optimization problem has been solved using PSO for 3-Generator and 13-Generator power system. The proposed technique improves the quality of the solution and reduces the computation time.

The algorithms are tested on number of sample problems. Numerical results for a test case show that the PSO-algorithms are capable of finding very nearly global solutions within a reasonable time. Depending on the problem and required solution quality, PSO algorithms exhibit a stagnation tendency with different degrees of severity. This tendency is smaller when using large population sizes and larger when using small population sizes. However, even though the probability of solving the problem increases, first hitting times are normally delayed. Independent restarts can improve the performance of various PSO algorithms. In some cases, configurations that favour an exploitative behaviour can outperform those that favour an exploratory one if optimal restart policies are used. However, the optimal restart policy is algorithm- and problem dependent and therefore cannot be defined a priori. When a limited number of function evaluations are allowed, configurations that favour an exploitative behaviour obtain the best results. When solution quality is the most important aspect, algorithms with exploratory properties are the best performing. By varying the inertia weight schedule, it is possible to control the convergence speed of the time-varying inertia weight variants. In the case of the time-decreasing inertia weight variant, faster schedules induce a faster convergence speed, albeit at the cost of increasing the algorithm's stagnation tendencies. In the time-increasing inertia weight variant, slow schedules provide the best performance both in terms of speed and quality.

Insights on experimental results ideally guide toward the definition of new better performing algorithms. Several interesting directions need to be explored in the future work. For example, to study in future work the approach to more effectively incorporate the impact of constraints into the inherent search mechanism of PSO which, at the same time, remains the advantageous features of PSO such as simplicity, fast convergence and easy implementation. Evaluating the performance of the improved algorithm on a wider variety of benchmark functions is also an interesting work. In addition, a lot of work needs to be done to further improve the global optimization ability of algorithm.

REFERENCES

[1] Dhillon, J.S. and Kothari, D.P. [2004], Power System Optimization, Prentice Hall of India.

[2] D
ao-Hyun Choi and Se-Young Oh [2000], A new mutation rule for Evolutionary Programming motivated from back propagation learning, IEEE transaction on Evolutionary Programming, vol.4, no.2, pp.188-190.

[3] E
berhart R and Kennedy J. [1995], A new optimizer using particle swarm theory, Proceedings of 6th Symposium on MicroMachine and Human Science, Nagoya, Japan, pp. 39–43.

[4] Kennedy, J., Eberhart, R. C. and Shi, Y. [2001], Swarm intelligence. San Francisco: Morgan Kaufmann Publishers.

[5] Kennedy, J. and Eberhart, R.C. [1995], Particle swarm optimization.InProceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948.

[6] Meng, Ke, Wang, H. G., Dong, Z. Y. and Wong, K. P. [2010], Quantum-Inspired Particle Swarm Optimization for Valve-Point Economic Load Dispatch, IEEE transactions on power systems, vol. 25, no. 1, pp.215-222.

[7] Parasopoulos, K.E., Vrahatis, M.N. [2004]. On the computation of all global minimizers through particle swarm optimization. IEEE Transactions on Evolutionary Computation vol. 8, no. 3, pp.211-224.

[8] Petalas Y.G., Parsopoulos K.E., Vrahatis M.N. [2007], Memetic particle swarm optimization, Annals of Operations Research vol. 156, no.1, pp. 99–127, DOI 10.1007/s10479-007-0224-y.

[9] Parsopoulos, K. E., and Vrahatis, M. N. [2002], Particle swarm optimization method for constrained optimization problems. In P. Sincak, J. Vascak, V. Kvasnicka& J. and Pospichal (Eds.), intelligent technologies–theory and application (New trends in intelligent technologies). Frontiers in artificial intelligence and applications, vol. 76, pp. 214–220. Amsterdam: IOS Press, ISBN:1-58603-256-9.

[10] Rao, S.S. [1999], Engineering Optimization Theory and Practice, 3rd ed New Age International (P) Ltd. Publishers.

[11] Shi, Y., and Eberhart, R.C. [1998], Parameter selection in particle swarm optimization. In: Porto, V.W., Saravanan,N.,Waagen, D., Eiben, A.E. (eds.) Evolutionary Programming, vol. 7, pp. 591–600. Springer-Verlag,Berlin.

[12] Shi Y. and Eberhart R. [2009], Monitoring of particle swarm optimization, Front. Computer science, China, vol. 3, no. 1, pp. 31-37.

[13] Saber A. Y., Chakraborty S., Razzak S.M. Abdur and Senjyu T. [2009], Optimization of economic load dispatch of higher order general cost polynomials and its sensitivity using modified particle swarm optimization, Electric Power Systems Research, vol. 79, no. 1, pp. 98–106.

[14] Tan, R.R. [2007], Hybrid evolutionary computation for the development of pollution prevention and control strategies. Journal of Cleaner Production vol. 15, no. 10, pp. 902–906.

[15] Vlachogiannis, J. G. and Lee, K. Y. [2009], Economic Load Dispatch—A Comparative Study on Heuristic Optimization Techniques With an Improved Coordinated Aggregation-Based PSO,IEEE Transactions On Power Systems, vol. 24, no. 2, pp. 991-1001.

[16] Zhang .J, Wang .J,and Yue .C; [2012]Small Population Based Particle Swarm Optimization for short term Hydrothermal Scheduling. IEEE transaction on Power System, vol.27, no.1, pp.142-152.