Journal of Energy & Environment 4 (2005) 49 – 61

Particle Swarm Optimization for Economic Dispatch of Generating Units with Valve-Point Loading

K. Mahadevan*, P. S. Kannan and S. Kannan***

**Department of Electrical Engg., A. K. College of Engineering, Krishnankoil-626190, India*
*** Department of Electrical Engg., Thiagarajar College of Engineering, Madurai-625015, India*
** Department of Electrical Engg., A. K. College of Engineering, Krishnankoil-626190, India*
Email: mahadevand@yahoo.com
(Received on 7 Dec 2004, revised on 2 Apr 2005)

Abstract

Economic Dispatch (ED) is one of the main functions of power system operation and control which determines the optimal real power settings of generating units with an objective of minimizing the total fuel cost. In all practical cases, the fuel cost of generators can be represented as a quadratic function of real power generation. In fact, discontinuity may also be observed in thermal power plants' fuel cost curves due to valve point loading. As the conventional optimization methods require the objective functions to be in continuous differentiable form, they fail to solve these problems. This paper presents a new approach using Particle Swarm Optimization (PSO) for solving the ED problem of generating units having non-smooth fuel cost functions with ramp rate limits. The proposed method is implemented for solving an example dispatch problem in the IEEE 30-bus 6-generation unit system. The obtained results are compared with those from the evolutionary programming (EP) method and show that the PSO approach is feasible and efficient.

Nomenclature

F	the operating cost
N	the number of generating units
P_i	the power output of *i*-th generating unit
$F_i(P_i)$	the individual fuel cost function of *i*-th generating unit
P_D	the total demand of the system
P_L	transmission losses of the system
$\lvert v \rvert$ and θ	voltage magnitude and phase angles of different buses
P_k	calculated real power for *PQ* bus *k*
Q_k	calculated reactive power for *PQ* bus *k*
P_k^{net}	specified real power for *PQ* bus *k*
Q_k^{net}	specified reactive power for *PQ* bus *k*
P_m	calculated real power for *PV* bus *m*
P_m^{net}	specified reactive power for *PV* bus *m*
$P_{ik(\min)}$	minimum output of *i*-th generating unit at *k*-th bus
$P_{ik(\max)}$	maximum output of *i*-th generating unit at *k*-th bus

$V_{k(\min)}$ minimum voltage at bus k
$V_{k(\max)}$ maximum voltage at bus k
$MVAf_{p,q}^{\max}$: maximum rating of transmission line connecting buses p and q.
a_i, b_i, c_i : cost coefficients of i-th generator
d_i, f_i : positive coefficients of i-th generator to reflect valve-point loading effect

Introduction

The process of scheduling generation to minimize the operating cost is called economic dispatch (ED). In this calculation, the generation costs are represented as curves, usually piecewise linear, and the overall calculation minimizes the operating cost by finding a point where the total output of the generations equals the total power that must be delivered and where the incremental cost of power generation is equal for all generators. However, if a generator is at its upper or lower limit, that generator's incremental cost is different.

ED is an important daily optimization task in the operation of a power system. Various mathematical programming methods and optimization techniques have been applied to ED. Most of these are calculus based optimization algorithms that are based on successive linearisations, and use the first and second differentiations of objective function and its constraint equations as the search directions [1]. They usually require the heat input-electric power output characteristics of generators to be of monotonically increasing nature or of a piecewise linearity. However, large modern generating units with multi-valve steam turbines exhibit a large variation in the input-output characteristic functions. The valve-point effects, owing to wire drawing as steam admission valve starts to open, typically produce a ripple-like heat rate curve. The conventional optimization methods are not suitable to solve such a problem. Hence, more general approaches are needed without restrictions on the shape of fuel cost functions.

Dynamic programming (DP) solution is one of the approaches to solve the inherently non-linear and discontinuous ED problem [2]. However, the dimensions of the problem would become extremely large with the increase of the variables (curse of dimensionality).

Methods such as Simulated Annealing (SA), Genetic Algorithms (GA) and Evolutionary Programming (EP) have the advantage of searching the solution space more thoroughly, and avoiding premature convergence to local minima [3-5]. However, the main difficulty is their sensitivity to the choice of parameters, such as temperatures in SA, the cross over and mutation probabilities in GA and scaling factor in EP.

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart [6], is one of the modern heuristic algorithms. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous nonlinear optimization problems [6-10]. The PSO technique can generate high-quality solutions within shorter calculation time and stable convergence characteristic than other stochastic methods [7-10]. Although the PSO seems to be sensitive to the tuning of some weights or parameters, many researches are still in progress for proving its potential in solving complex power system problems [9]. Researchers including Yoshida et al [11] have presented a PSO for reactive power and voltage control (VVC) considering voltage security assessment. The feasibility of their method is compared with the reactive Tabu system (RTS) and enumeration method on practical power system, and has shown promising results. Naka et al [12] have presented the use of a hybrid PSO method for solving efficiently the practical distribution state estimation problem.

In this paper, a PSO method for solving the ED problem in power system is proposed. The proposed method considers the valve point effect of generators for actual power system operation. The feasibility of the proposed method was demonstrated for the IEEE 30-bus 6-generation unit system as compared with the EP method in terms of solution quality and computation efficiency.

Problem Formulation

The ED problem with ramp rate constraints can be described as an optimization process with the following objective function and constraints:

$$\min F = \sum_{i=1}^{N} F_i(P_i) \tag{1}$$

The fuel cost functions of the generating units are generally characterized by second-order polynomials as

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{2}$$

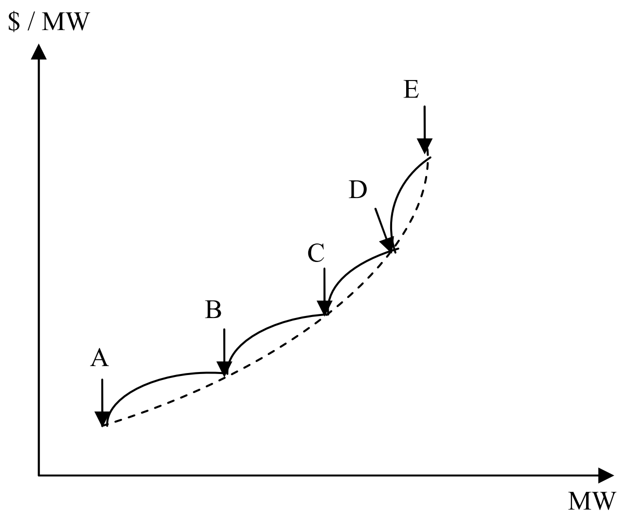
However, the generating units with multi-valve steam turbines exhibit a greater variation in the fuel cost functions. Large steam turbine generators usually have a number of steam admission valves that are opened in sequence to obtain ever-increasing output from the unit. Fig. 1 shows both input-output and incremental heat rate characteristics for a unit with steam admission valves. As the unit loading increases, the input to the unit increases. However, when a valve is just opened, the throttling losses increase rapidly so that the input heat increases rapidly and the incremental heat rate rises suddenly. This gives rise to the non-smooth type of heat input and discontinuous type of incremental heat rate characteristics. This type of characteristic should be used in order to schedule steam units accurately, but it cannot be used in traditional optimization methods because it does not meet the convex condition.

Typically, the valve-point effects, due to wire drawing as each steam admission valve starting to open, produce ripple-like heat rate curve [4]. To model this effect, a recurring rectified sinusoid contribution is added to the second-order polynomial function to represent the input-output equation. Thus (1) becomes

$$\min F = \sum_{i=1}^{N} \left(a_i + b_i P_i + c_i P_i^2 + \left| e_i \sin\left(f_i \left(P_i - P_{i(\min)} \right) \right) \right| \right) \tag{3}$$

The cost is optimized with the following power system constraint

$$\sum_{i=1}^{N} P_i - P_D - P_L = 0 \tag{4}$$



A : Primary Valve B : Secondary Valve
C : Tertiary Valve D : Quaternary Valve
E : QuinaryValve

Fig. 1 Characteristics of a steam turbine generator with steam admission valves

The power flow equation of the power network

$$g(|v,\theta|) = 0 \tag{5}$$

where,

$$g(|v,\theta|) = \begin{vmatrix} \left.\begin{matrix} P_k(|v,\theta|) - P_k^{\text{net}} \\ Q_k(|v,\theta|) - Q_k^{\text{net}} \end{matrix}\right\} \text{for each } PQ \text{ bus } k \\ P_m(|v,\theta|) - P_m^{\text{net}} \text{ for each } PV \text{ bus } m \end{vmatrix}$$

The inequality constraint on real power generation P_i of each generation i

$$P_{ik(\min)} \le P_{ik} \le P_{ik(\max)} \tag{6}$$

The inequality constraint on voltage of each PQ bus is

$$V_{k(\min)} \le V_k \le V_{k(\max)} \tag{7}$$

Power limit on transmission line

$$MVAf_{p,q} \leq MVAf_{p,q}^{\max} \tag{8}$$

Evolutionary Computation Techniques and Particle Swarm Optimization

This section gives a brief introduction of EC techniques viz., Genetic Algorithms and Evolutionary Programming and the differences with PSO.

Genetic algorithms

Genetic Algorithm (GA) is a search algorithm based on the conjecture of natural selection and genetics. The features of GA are different from other search techniques in several aspects. First, the algorithm is a multi-path that searches many peaks in parallel, and hence reducing the possibility of local minimum trapping. Secondly, GA works with a coding of parameters instead of the parameters themselves. The coding of parameter will help the genetic operator to evolve the current state into the next state with minimum computations. Thirdly, GA evaluates the fitness of each string to guide its search instead of the optimization function. The GA only needs to evaluate objective function (fitness) to guide its search. There is no requirement for derivatives or other auxiliary knowledge. Therefore, there is no need for computation of derivatives. Finally, GA explores the search space where the probability of finding improved performance is high.

Genetic Algorithm derives its behavior from a metaphor of some of the mechanisms of “Evolution in Nature” and, in essence, consists of a population of bit strings transformed by three genetic operators: selection, crossover and mutation. Each string (called chromosome) represents a possible solution to the problem being optimized and each bit (or group of bits) represents a value for some variable of the problem (gene). These solutions are classified by an evaluation function, giving better values/fitness, to better solutions. Each solution must be evaluated by the fitness function to produce a value. The pair of chromosome and fitness represents an individual. The selection operator creates a new population (or generation) by selecting individuals from the old population, biased towards the best. Crossover is the main genetic operator and consists of swapping chromosome parts between individuals. Crossover is not performed on every pair of individuals, its frequency being controlled by a crossover probability. This probability should have a larger value. The last operator is mutation and consists of changing a random part of the string representing the individual. This operator must be used with some care, with low probability.

Evolutionary programming

Evolutionary programming (EP) is a computational intelligence method in which an optimization algorithm is the main engine for the process of three steps; namely, natural selection, mutation and competition. It is a stochastic optimization strategy, which places emphasis on the behavioral linkage between parents and their offspring, rather than seeking to emulate specific genetic operators as in GAs. Crossover in GAs often destroys the essential behavioral link between each parent and its offspring. However, mutation ensures the functionally of the next generation. Therefore, EP tends to generate more effective and efficient searches. It operates on populations of real values (floating points) that represent the parameter set of the problem being solved over some finite ranges. Each representation is an individual in the EP population. The population is initialized with random individuals at the start of the EP run. The EP searches the space of possible real values for better individuals. The search is guided by fitness values returned by the environment. This gives a measure of how well adapted each individual is in terms of solving the problem, and hence determines its probability of appearing in future generation. EP uses two types of rules, named the selection rule and combination rule in its search. The selection rule is used to determine the individuals that will be represented in the next generation. It includes competition in which each individual in the combined population has to compete with some other individuals to get chance to be transcribed to the next generation. On the other hand, the combination rule operates on selected individuals to produce new individuals that appear in the next generation. The selection mechanism is based on a fitness measure or objective function values, defined on each individual in the population. The combination rule is used to introduce new individuals into the current population or to create a new population based on the current population.

Components of EP

1) Initialization: the initial population with population size n consists of individuals and is created randomly. The fitness score f_s of each individual p_s *is* obtained by a fitness function.

2) Mutation: Each p_s is mutated in order to generate a new population. The new individuals, p_{s+m} are calculated in accordance with the following equation.

$$p_{s+m} = p_{s,k} + N\left(0, \beta\left(x_{k\max} - x_{k\min}\right)\frac{f_s}{f_{\max}}\right), k = 1, 2, ..., n \tag{9}$$

where,

$p_{s,k}$	is the k th element of the s th individual
$N(\mu,\sigma^2)$	is a Gaussian random variable with mean μ and variance σ^2
$f_{\max}$	is the maximum fitness of the old population
$x_{k\max}$	is the maximum limits of the k th element
$x_{k\min}$	is the minimum limits of the k th element
β	is is the mutation scale
n	is the population size

3) Competition: Each individual p_s in the combined population has to compete with some other individuals to get chance to be transcribed to the next generation. A weight value w_s is assigned to the individual according to the competition as follows.

$$w_s = \sum_{j=1}^{P} w_j$$

where, P is the competition number, which is largely dependent on the parameters of the system such as population size. w_j is a number of 0 or 1, which represents win 1, loss 0, as p_s competes with a randomly selected individual p_r in the combined population. In the maximization problem, w_j is given by the followings:

$w_j = 1$ if $f_s > f_r$
$w_j = 0$ otherwise

where, f_r is the fitness of a randomly selected individual p_r.

When all individuals, p_s, s = 1, 2, 2m get their competition weights, they will be ranked in descending order of their corresponding value w_s by a sorting algorithm. The first m individuals are transcribed along with their corresponding fitness, f_s to be the basis for the next generation.

The aforementioned EC techniques exploit a set of potential solutions, named *population*, and detect the optimal problem solution through cooperation and competition among the individuals of the population. These techniques often find optima in complicated optimization problems faster than traditional optimization problems.

PSO method

The PSO method is a member of the wide category of Swarm Intelligence methods for solving global optimization problems. It was originally proposed by J. Kennedy as a simulation of social behavior, and it was initially introduced as an optimization method in 1995 by Eberhart and Kennedy. PSO is related with Artificial Life, and specifically to swarming theories, and also with EC techniques.

Natural creatures sometimes behave as a swarm. One of the main streams of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer. Swarm behaviour can be modeled with a few simple rules.

PSO is basically developed through artificial simulation of bird flocking in two-dimension space [6]. The position of each agent is represented by *xy* axis position and also the velocity is expressed by *vx* (the velocity of *x* axis) and *vy* (the velocity of *y* axis). Modification of the agent position is realized by the position and velocity information.

Searching procedures by PSO based on the above concept can be described as follows: bird flocking optimizes certain objective function. Each agent knows its best value so far (*pbest*) and its *xy* position. Moreover, each agent knows the best value so far in the group (*gbest*) among *pbest*s. The modified velocity of each agent can be calculated using the following information.

- the current positions (*x*,*y*)
- the current velocities (*vx*, *vy*)
- the distance between the current position and *pbest*
- the distance between the current position and *gbest*

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$v_i^{k+1} = wv_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k) \tag{10}$$

where,

v_i^k	: velocity of agent *i* at iteration *k*
w	: weighting function
c_1, c_2	: acceleration constant
rand	: random number between 0 and 1
s_i^k	: current position of agent *i* at iteration *k*

pbest$_i$: *pbest* of agent *i*
gbest : *gbest* of the group

Suitable selection of weighting function w in (11) provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. The following weighting function is usually utilized in (10):

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{11}$$

where,

$w_{\max}$: initial weight,
$w_{\min}$: final weight,
$iter_{\max}$: maximum iteration number,
$iter$: current iteration number.

Using the above equation, a certain velocity, which gradually gets close to *pbest* and *gbest* can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{12}$$

The constants $c_1$and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward the *pbest* and *gbest* positions. Low values allow particles to far from the target regions and high values result in abrupt movement toward, or past, target regions. Hence, the acceleration constants $c_1$and c_2 are often set to be 2.0 according to past experiences.

Implementation of PSO in ED Problem

The details of the implementation of PSO components are summarized as follows:

Initialisation

The real power outputs of each generator are randomly generated, called genes, comprise an individual. Each individual within the population represents a candidate solution for solving ED problem. For example, i-th individual P_{gi} can be defined as follows.

$$P_{gi} = \left[P_{i1}, P_{i2}, ...P_{in}\right], \quad i = 1, 2, ..., p \tag{13}$$

where p is the population size and n is the number of generators.

Fitness evaluation

The fitness is called directly from the objective function. The active power generations are control variable, which are self-constrained. Penalties are given for violation of power balance constraint (4).

A step-by-step procedure of the proposed method for solving ED problem is as follows:

Step 1: Initialize the parameters such as the swarm size, weighting function, acceleration constant, etc. Initial searching points (s_i^0) and velocities (v_i^0) of each individual for N generating units are usually generated randomly. These initial individuals must be feasible candidate solutions that satisfy the practical operation constraints;
Step 2: Calculate the fitness value of the each individual P_{gi} in the population pool;
Step 3: Compare each individual's fitness value with its *pbest*. The best value among the *pbest*s is denoted as *gbest*;
Step 4: The velocity *v* of each individual is updated according to (10);
Step 5: The member position *s* of each individual is modified using (12);
Step 6: If the fitness function value is better than the current *pbest*, the *pbest* value is replaced by the current value. Also, if the *pbest* is better than the current *gbest*, *gbest* is replaced by the best value and its number with the best value is stored;
Step 7: If the number of iterations reaches set value, then go to Step 8. Otherwise go to Step 2;
Step 8: The individual that generates the latest *gbest* is the optimal generation power of each unit with the minimum total generation cost.

The above strategies are illustrated in the flow chart shown in Fig.2.

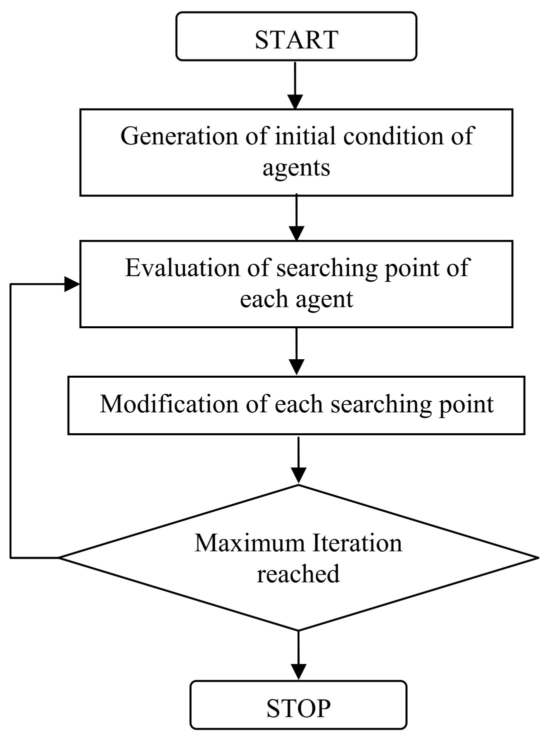


Fig. 2 Flowchart for PSO algorithm

Simulation Results

In order to compare the effectiveness of the proposed method, the IEEE 30 bus 6-unit system and the ED results for it available from the EP method [13] are considered. The simulations were carried out on a Pentium III 500 MHz processor. The program was coded in MATLAB 6.0 software.

Test system description

In this section, the modified IEEE 30-bus system taken from [13] is used to show the effectiveness of the algorithm. The network consists of six generator-buses, 21 load-buses and 43 branches. The parameters of the characteristics of the steam turbine generators are given in the Appendix. The load periods are simply divided into the valley load duration and the peak load duration. The peak loads are twice the valley loads. The PSO control parameters chosen are following:

- population size = 20
- maximum no. of iteration = 100
- w_{max} = 0.9 and w_{min} = 0.4
- acceleration constant c_1 = 2 and c_2 = 2

The comparison of results for the valley and peak load periods is reported in Table 1. The cost saved is 2.72 $/h in valley load duration and 11.55 $/h in peak load duration against EP method. A fuel cost of 1 $/MBtu is assumed for this test system.

Table 1 Comparison of results of IEEE 30-bus system

	Valley load duration		Peak load duration	
	EP [13]	PSO	EP [13]	PSO
P_1(MW)	199.007	171.3896	250	250.617
P_2(MW)	50.852	59.8113	124.982	120.247
P_3(MW)	15	20.0557	36.24	36.003
P_4(MW)	10	15.2631	70	69.8827
P_5(MW)	10	13.5524	60	59.8698
P_6(MW)	12	14.6535	58.447	61.1606
P_L(MW)	13.659	11.3257	32.869	30.9809
Fuel Cost ($/h)	819.8466	817.1211	2144.451	2132.9
CPU time (min)	3.16	0.8	5.24	1.2

The variation of cost incurred ($/h) against the number of iterations for the two load periods is shown in Fig. 3. The minimum solution is obtained within 100 iterations. As seen in Table 1, for peak load period the dispatch levels of both EP and PSO are closer, however, they are at different levels in valley load period. As both the EP and PSO are stochastic optimization techniques, starting their search from a random initial population, the solution quality is sensitivity to the choice of parameters, such as scaling factor in EP and acceleration constants c_1, c_2 and weighting function w. Also it depends on the number of iterations also.

Fig. 4 shows the variation of fuel cost during different trials of this method. In each trial, 100 iterations are set. It is observed that there is no huge variation in the fuel cost during several trial runs. Thus the consistency of PSO algorithm for this type of non-convex and non-differentiable problem is verified.

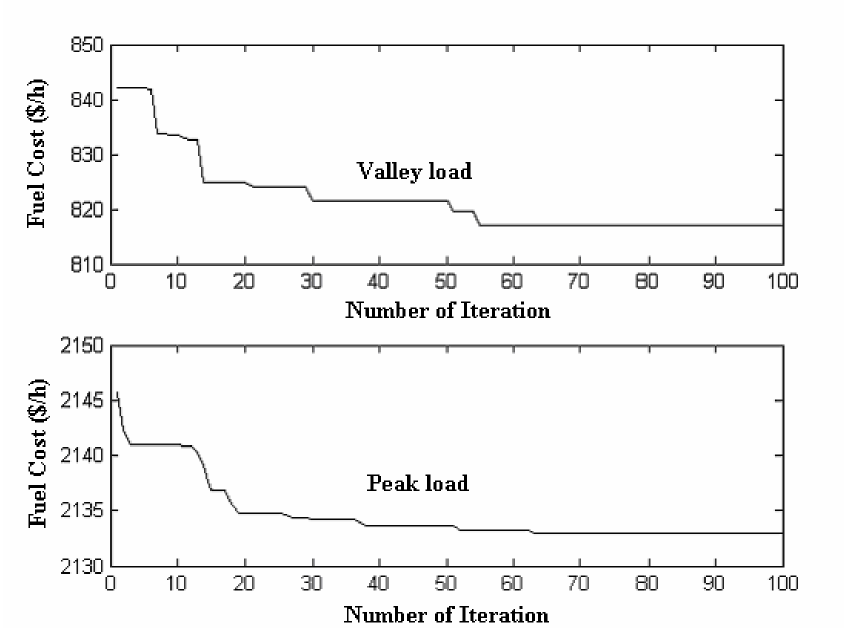


Fig. 3 Convergence characteristics of PSO

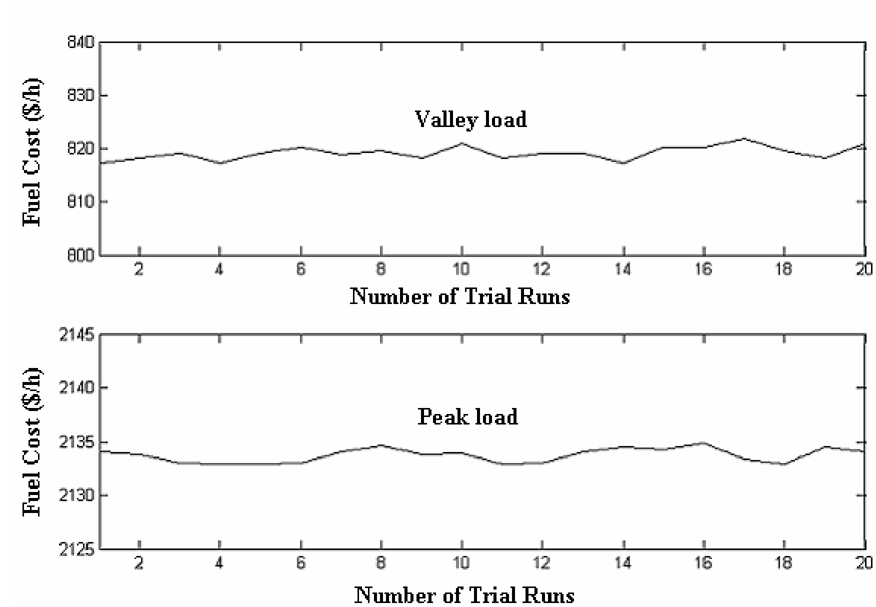


Fig. 4 Consistency nature of PSO

It is clear that PSO performs better than EP, in terms of both solution quality and execution time. The reason for the PSO to take less time is obvious. It works based on updating of velocity and position of particles given by (10) and (12) and also it does not perform the mutation and selection operations as in evolutionary processes.

Conclusion

In this paper the particle swarm optimization (PSO) method is used to solve the economic dispatch problem with non-smooth fuel cost function. The advantage of the PSO algorithm lies in its stability to handle any type of unit characteristics, whether smooth or not. The PSO algorithm approach yields solutions which are optimal or near optimal. The results obtained for the IEEE 30-bus 6-unit test system showed that the PSO algorithm was good in terms of its potential in solving ED problems and also computational time besides the high quality of solutions compared to the evolutionary programming (EP) method.

Acknowledgment

Authors thank the authorities of A. K. College of Engineering, Krishnankoil, India and Thiagarajar college of Engineering, Madurai, India for the facilities provided to carry out this work.

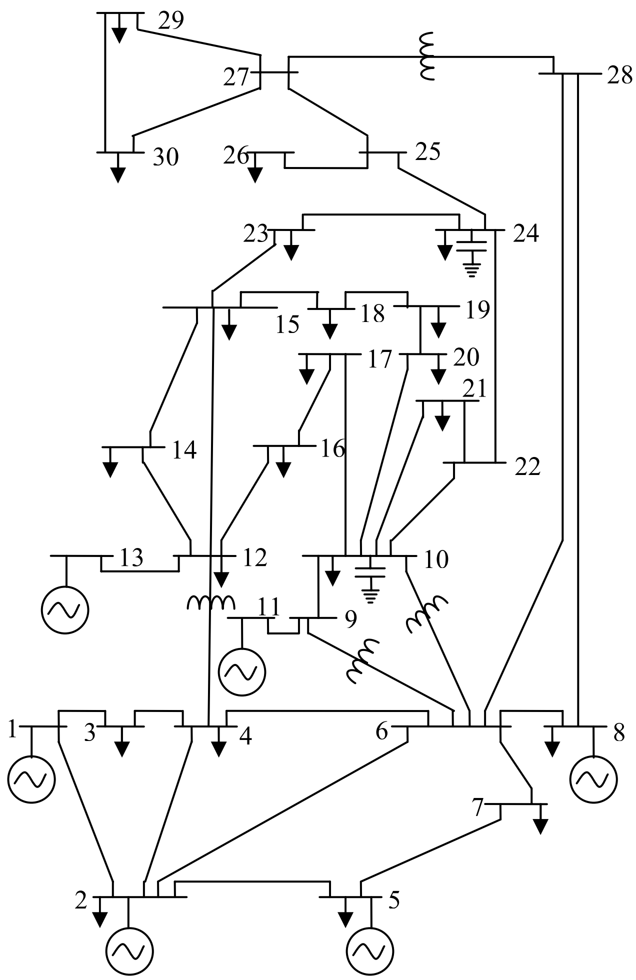
References

[1] A. J. Wood and B. F. Woolenberg “Power Generation, Operation and control,” John Wiley and Sons, Inc., 1996.
[2] Z. X. Lianf and J. D. Glover, “A zoom feature for a dynamic programming solution to economic dispatch including transmission losses,” IEEE transactions on power systems, Vol. PWRS-7, No. 2, 1992, pp 544-550.
[3] K. P. Wong and C. C. Fung, “Simulated annealing based economic dispatch algorithm,” IEE Proc.-C, Vol. 140, No. 6, 1993, pp 509-515.
[4] D. C. Walters and G. B. Sheble, “Genetic algorithm solution of economic dispatch with valve point loading,” IEEE transactions on power systems, Vol. PWRS-8, No. 3, 1993, pp 1325-1332.
[5] H. T. Yang, P. C. Yang and C. –L. Huang, “Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions,” IEEE transactions on power systems, Vol. 11, No. 1, 1993, pp 112-118.
[6] J. Kennedy, R. Eberhart, “Particle swarm optimization,” in Proc., IEEE international conference on neural networks (ICNNN ’95), Vol. IV, Perth, Australia, 1995, pp 1942-1948.
[7] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in Proc., IEEE international conference evolutionary computer, May 1998, pp 69-73.
[8] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in Proc., international conference evolutionary computer, NJ, 1999, pp 1945 -1950.
[9] R. C. Eberhart and Y. Shi, “Comparison between genetic algorithms and particle swarm optimization,” in Proc., IEEE international conference evolutionary computer, May 1998, pp 611-616.
[10] P. J. Angeline, “Using selection to improve particle swarm optimization,” in Proc., IEEE international conference evolutionary computer, May 1998, pp 84-89.
[11] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage security assessment,” IEEE transactions on power systems, Vol.15, 2000, pp 1232-1239.
[12] S. Naka, T. Gengi, T. Yura and Y. Fukuyama, “Practical distribution state estimation using hybrid particle swarm optimization,” in Proc., IEEE power eng. soc. winter meeting, Vol.2, 2001, pp 815-820.
[13] L. L. Lai, ‘Intelligent system applications in power engineering – evolutionary programming and neural networks,” John Wiley and Sons, Inc., New York, 1998.

Appendix

Parameters of the characteristics of steam turbine generators

Generator	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
Bus	1	2	5	8	11	13
P_{max} (MW)	250	160	10	70	60	80
P_{min} (MW)	50	20	15	10	10	12
a (MBtu)	0	0	0	0	0	0
b (MBtu/MW)	2.0	1.75	1.0	3.25	3.0	3.0
c (MBtu/MW^2)	0.00375	0.0175	0.0625	0.0834	0.025	0.025
e (MBtu)	15	10	10	5	5	5
f (rad/MW)	0.06283	0.08976	0.14784	0.20944	0.25133	0.1848



IEEE 30-Bus system