International Journal of Electrical and Power Engineering 6 (5-6): 140-148, 2012
ISSN: 1990-7958
© Medwell Journals, 2012

Dynamic Economic Load Dispatch of Thermal Power System Genetic Algorithm

[1]W.M. Mansour, [1]M.M. Salama, [2]S.M. Abdelmaksoud and [2]H.A. Henry
[1]Department of Electrical Power Systems, [2]Department of Electrical Engineering,
Shoubra Faculty of Engineering, Benha University, Cairo, Egypt

Abstract: Economic Load Dispatch (ELD) problem is one of the most important problems to be solved in the operation and planning of a power system. The main objective of the economic load dispatch problem is to determine the optimal schedule of output powers of all generating units so as to meet the required load demand at minimum operating cost while satisfying system equality and inequality constraints. This study presents an application of Genetic Algorithm (GA) for solving the ELD problem to find the global or near global optimum dispatch solution. The proposed approach has been evaluated on 26-bus, 6-unit system with considering the generator constraints, ramp rate limits and transmission line losses. The obtained results of the proposed method are compared with those obtained from the Conventional Lambda Iteration Method and Particle Swarm Optimization (PSO) Technique. The results show that the proposed approach is feasible and efficient.

Key words: Economic load dispatch, ramp rate limits, particle swarm optimization, genetic algorithm, Egypt

INTRODUCTION

With the development of modern power systems, Economic Load Dispatch (ELD) problem has received an increasing attention. The primary objective of ELD problem is to minimize the total generation cost of units while satisfying all units and system equality and inequality constraints (Wood and Wollenberg, 1984). In this problem, the generation costs are represented as curves and the overall calculation minimizes the operating cost by finding the point where the total output power of the generators equals the total power that must be delivered. In the traditional ELD problem, the cost function for each generator has been represented approximately by a single quadratic function and is solved using mathematical programming based optimization techniques such as Lambda Iteration, Gradient, Newton, Linear and Dynamic Programming Methods (Salama, 1999; Report, 1971). All these methods assume that the cost curve is continuous and monotonically increasing. In these methods, computational time increases with the increase of the dimensionality of the ELD problem. The most common optimization techniques based upon artificial intelligence concepts such as evolutionary programming (Sinha *et al*., 2003), simulated annealing (Wong and Fung, 1993), artificial neural networks (Nanda *et al*., 1997), tabu search (Lin *et al*., 2002), Particle Swarm Optimization (PSO) (Gaing, 2004; Park *et al*., 2005; Jeyakumar *et al*., 2006) and genetic algorithm (Walters and Sheble, 1993; Tippayachai *et al*., 2002; Bakirtzis *et al*., 1994; Sheble and Brittig, 1995; Chen and Chang, 1995; Song *et al*., 1995) have been given attention by many researchers due to their ability to find an almost global or near global optimal solution for ELD problems with operating constraints. Major problem associated with these techniques is that appropriate control parameters are required. Sometimes these techniques take large computational time due to improper selection of the control parameters. The GA is a stochastic global search and optimization method that mimics the metaphor of natural biological evolution such as selection, crossover and mutation (Goldberg, 1990). GA is started with a set of candidate solutions called population (represented by chromosomes). At each generation, pairs of chromosomes of the current population are selected to mate with each other to produce the children for the next generation. The chromosomes which are selected to form the new offspring are selected according to their fitness. In general, the chromosomes with higher fitness values have higher probability to reproduce and survive to the next generation. While, the chromosomes with lower fitness values tend to be discarded. This process is repeated until a termination condition is reached (for example, maximum number of generations).

MATERIALS AND METHODS

Formulation of an ELD problem with generator constraints: The primary objective of the ELD problem is to minimize the total fuel cost of thermal power plant

Corresponding Auhtor: W.M. Mansour, Department of Electrical Power Systems, Shoubra Faculty of Engineering, Benha University, Cairo, Egypt

subjected to the operating constraints of a power system. In general, the ELD problem can be formulated mathematically as a constrained optimization problem with an objective function of the form:

$$F_T = \sum_{i=1}^{n} F_i(P_i) \qquad (1)$$

Where:

F_T	=	The total fuel cost of the system ($ h^{-1})
n	=	The total number of generating units
$F_i(P_i)$	=	The operating fuel cost of generating unit I ($ h^{-1})

Generally, the fuel cost function of the generating unit is expressed as a quadratic function as given in Eq. 2:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \qquad (2)$$

Where:

P_i	=	The real output power of unit i (MW)
a_i-c_i	=	The cost coefficients of generating unit I

The minimization of the ELD problem is subjected to the following constraints:

Real power balance constraint: For power balance, an equality constraint should be satisfied. The total generated power should be equal to the total load demand plus the total line losses. The active power balance is given by:

$$\sum_{i=1}^{n} P_i = P_D + P_L \qquad (3)$$

Where:

P_D	=	The total load demand (MW)
P_L	=	The total line losses (MW)

The total transmission line loss is assumed as a quadratic function of output powers of the generator units (Momoh *et al*., 1999) that can be approximated in the equation:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j \qquad (4)$$

Where:

B_{ij}	=	The transmission loss coefficient matrix
P_i, P_j	=	The power generation of ith and jth units

Generator power limit constraint: The generation output power of each unit should lie between minimum and maximum limits. The inequality constraint for each generator can be expresses as:

$$P_{i,\min} \le P_i \le P_{i,\max} \qquad (5)$$

Where $P_{i,\min}$ and $P_{i,\max}$ are the minimum and maximum power outputs of generator i (MW), respectively. The maximum output power of generator is limited by thermal consideration and minimum power generation is limited by the flame instability of a boiler.

Ramp rate limit constraint: The generator constraints due to ramp rate limits of generating units are given as: As generation increases:

$$P_{i(t)} - P_{i(t-1)} \le UR_i \qquad (6)$$

As generation decreases:

$$P_{i(t-1)} - P_{i(t)} \le DR_i \qquad (7)$$

Therefore, the generator power limit constraints can be modified as:

$$\begin{aligned} &\max\left(P_{i,\min}, P_{i(t-1)} - DR_i\right) \le P_{i(t)} \\ &\le \min\left(P_{i,\max}, P_{i(t-1)} + UR_i\right) \end{aligned} \qquad (8)$$

From Eq. 8, the limits of minimum and maximum output powers of generating units are modified as:

$$P_{i,\min,\text{ramp}} = \max\left(P_{i\min}, P_{i(t-1)} - DR_i\right) \qquad (9)$$

$$P_{i,\max,\text{ramp}} = \min\left(P_{i\max}, P_{i(t-1)} + UR_i\right) \qquad (10)$$

Where:

$P_{i(t)}$	=	The output power of generating unit i (MW) in the time interval (t)
$P_{i(t-1)}$	=	The output power of generating unit i (MW) in the previous time interval (t - 1)
UR_i	=	The up ramp limit of generating unit i (MW/time-period)
DR_i	=	The down ramp limit of generating unit i (MW/time-period)

The ramp rate limits of the generating units with all possible cases are shown in Fig. 1.

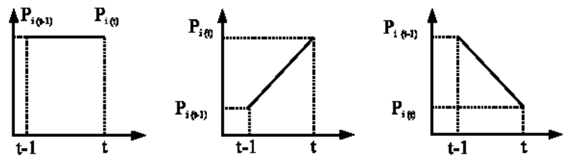

Fig. 1: Ramp rate limits of the generating units

Overview of Particle Swarm Optimization (PSO): Particle Swarm Optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. The PSO algorithm searches in parallel using a group of random particles. Each particle in a swarm corresponds to a candidate solution to the problem. Particles in a swarm approach to the optimum solution through its present velocity, its previous experience and the experience of its neighbors. In every generation, each particle in a swarm is updated by two best values. The first one is the best solution (best fitness), it has achieved so far. This value is called Pbest. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. Each particle moves its position in the search space and updates its velocity according to its own flying experience and neighbor's flying experience. After finding the two best values, the particle update its velocity according to Eq. 11:

$$V_i^{k+1} = \omega \times V_i^k + C_1 \times R_1 \times \left(Pbest_i^k - P_i^k\right) + C_2 \times R_2 \times \left(gbest^k - P_i^k\right) \tag{11}$$

Where:

V_i^k	=	The velocity of particle i at iteration k
P_i^k	=	The position of particle i at iteration k
ω	=	The inertia weight factor
C_1, C_2	=	The acceleration coefficients
R_1, R_2	=	Positive random numbers between 0 and 1
$Pbest_i^k$	=	The best position of particle i at iteration k
$gbest^k$	=	The best position of the group at iteration k

The constants C_1 and C_2 represent the weighting of the stochastic acceleration terms that pull each particle toward Pbest and gbest positions. Low values allow particles to roam far from the target regions while high values result in abrupt movement toward or past, target regions. Hence, the acceleration constants were often set to be 2.0 according to past experiences. Suitable selection of inertia weight in Eq. 11 provides a balance between local and global searches.

A low value of inertia weight implies a local search while a high value leads to global search. As originally developed, the inertia weight factor often decreases often is decreased linearly from about 0.9-0.4 during a run. It was proposed in Shi and Eberhart (1998). In general, the inertia weight ω is set according to Eq. 12:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iter_{max}} \times Iter \tag{12}$$

Where:

ω_{min}	=	Minimum value of inertia weight factor
ω_{max}	=	Maximum value of inertia weight factor
$Iter_{max}$	=	The maximum iteration number
Iter	=	The current iteration number

The current position (searching point in the solution space) can be modified by Eq. 13:

$$P_i^{k+1} = P_i^k + V_i^{k+1} \tag{13}$$

Implementation of PSO for solving ELD problem: The step by step procedure of the PSO Technique for solving ELD problem is as follows:

Step 1: Select the parameters of PSO such as population size (N), acceleration constants (C_1 and C_2), minimum and maximum value of inertia weight factor (ω_{min} and ω_{max}).

Step 2: Initialize a population of particles with random positions and velocities. These initial particles must be feasible candidate solutions that satisfy the practical operation constraints.

Step 3: Evaluate the fitness value of each particle in the population using the objective function given in Eq. 2.

Step 4: Compare each particle's fitness with the particles Pbest. If the current value is better than Pbest then set pbest equal to the current value.

Step 5: Compare the fitness with the population overall previous best. If the current value is better than gbest then set gbest equal to the current value.

Step 6: Update the velocity of each particle according to Eq. 11.

Step 7: The position of each particle is modified using Eq. 13.

Step 8: Go to step 9 if the stopping criteria is satisfied, usually a sufficiently good fitness or a maximum number of iterations. Otherwise go to step 3.

Step 9: The particle that generate the latest gbest is the optimal generation power of each unit with the minimum total cost of generation.

The procedure of Particle Swarm Optimization (PSO) Technique can be summarized in the flow chart shown in Fig. 2.

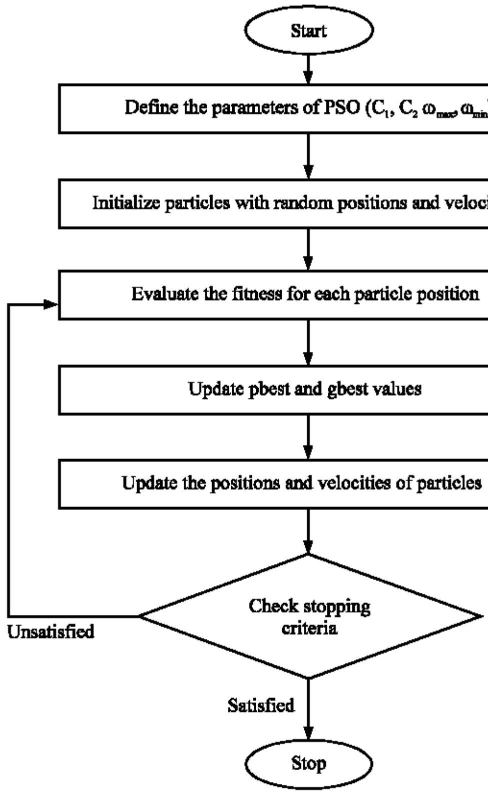
Start
Define the parameters of PSO (C_1, C_2 ω_{max}, ω_{min})
Initialize particles with random positions and velocities
Evaluate the fitness for each particle position
Update pbest and gbest values
Update the positions and velocities of particles
Check stopping criteria
Unsatisfied
Satisfied
Stop

Fig. 2: Flow chart of PSO Technique

Genetic Algorithm (GA): The GA is a method for solving optimization problems that is based on natural selection, the process that drives biological evolution. The general scheme of GA is initialized with a population of candidate solutions (called chromosomes). Each chromosome is evaluated and given a value which corresponds to a fitness level in problem domain. At each generation, the GA selects chromosomes from the current population based on their fitness level to produce offspring. The chromosomes with higher fitness levels have higher probability to become parents for the next generation while the chromosomes with lower fitness levels to be discarded. After the selection process, the crossover operator is applied to parent chromosomes to produce new offspring chromosomes that inherent information from both sides of parents by combining partial sets of genes from them. The chromosomes or children resulting from the crossover operator will now be subjected to the mutation operator in final step to form the new generation. Over successive generations, the population evolves toward an optimal solution. The features of GA are different from other traditional methods of optimization in the following respects (Mimoun *et al*., 2006):

- GA does not require derivative information or other auxiliary knowledge
- GA work with a coding of parameters instead of the parameters themselves. For simplicity, binary coded is used in this study
- GA search from a population of points in parallel, not a single point
- GA use probabilistic transition rules, not deterministic rules

Genetic algorithm operators: At each generation, GA uses three operators to create the new population from the previous population.

Selection or reproduction: Selection operator is usually the first operator applied on the population. The chromosomes are selected based on the Darwin's Evolution Theory of survival of the fittest. The chromosomes are selected from the population to produce offspring based on their values. The chromosomes with higher fitness values are more likely to contributing offspring and are simply copied on into the next population. The commonly used reproduction operator is the proportionate reproduction operator. The ith string in the population is selected with a probability proportional to F_i where F_i is the fitness value for that string. The probability of selecting the ith string is:

$$P_{ri} = \frac{F_i}{\sum\limits_{j=1}^{n} F_j} \quad (14)$$

where, n is the population size, the commonly used selection operator is the Roulette-wheel Selection Method. Since, the circumference of the wheel is marked according to the string fitness, the roulette-wheel mechanism is expected to make F_i copies of the ith string in the mating pool. The average fitness of the population:

$$F_{avg} = \frac{\sum\limits_{i=1}^{n} F_i}{n} \quad (15)$$

Crossover or recombination: The basic operator for producing new chromosomes in the GA is that of crossover. The crossover produce new chromosomes have some parts of both parent chromosomes. The simplest form of crossover is that of single point crossover. In single point crossover, two chromosomes strings are selected randomly from the mating pool. Next, the crossover site is selected randomly along the string length and the binary digits are swapped between the two strings at crossover site.

Mutation: The mutation is the last operator in GA. It prevents the premature stopping of the algorithm in a local solution. This operator randomly flips or alters one or more bits at randomly selected locations in a chromosome from 0-1 or vice versa.

Parameters of GA: The performance of GA depends on choice of GA parameters such as:

Population size (N): The population size affects the efficiency and performance of the algorithm. Higher population size increases its diversity and reduces the chances of premature converge to a local optimum but the time for the population to converge to the optimal regions in the search space will also increase. On the other hand, small population size may result in a poor performance from the algorithm. This is due to the process not covering the entire problem space. A good population size is about 20-30, however sometimes sizes 50-100 are reported as best.

Crossover rate: The crossover rate is the parameter that affect the rate at which the process of cross over is applied. This rate generally should be high, about 80-95%.

Mutation rate: It is a secondary search operator which increases the diversity of the population. Low mutation rate helps to prevent any bit position from getting trapped at a single value whereas high mutation rate can result in essentially random search. This rate should be very low.

Termination of the GA: The generational process is repeated until a termination condition has been satisfied. The common terminating conditions are: fixed number of generations reached, a best solution is not changed after a set number of iterations or a cost that is lower than an acceptable minimum.

GA applied to ELD problem: The step by step algorithm of the proposed method is explained as follow:

Step 1: Read the system input data, namely fuel cost curve coefficients, power generation limits, ramp rate limits of all generators, power demands and transmission loss coefficients.

Step 2: Select GA parameters such as population size, length of string, probability of crossover, probability of mutation and maximum number of generations.

Step 3: Generate randomly a population of binary string.

Step 4: The generated string is converted in feasible range by using Eq. 16:

$$P_{gi} = P_{i\,min} + \left(\frac{P_{i\,max} - P_{i\,min}}{2^L - 1} \right) . P_{m(i)} \qquad (16)$$

Where:

L = The string length

$P_{m(I)}$ = The decimal value of ith generating unit in the string

Step 5: Calculate the fitness value for each string in the population.

Step 6: The chromosomes with lower cost function are selected to become parents for the next generation.

Step 7: Perform the crossover operator to parent chromosomes to create new offspring chromosomes.

Step 8: The mutation operator is applied to the new offspring resulting from the crossover operation to form the new generation.

Step 9: If the number of iterations reaches the maximum then go to step 10 otherwise, go to step 5.

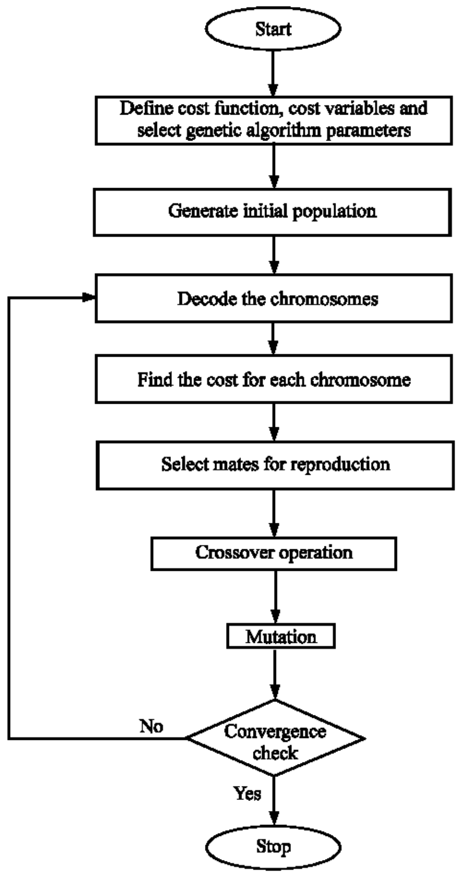
Start
Define cost function, cost variables and select genetic algorithm parameters
Generate initial population
Decode the chromosomes
Find the cost for each chromosome
Select mates for reproduction
Crossover operation
Mutation
No
Convergence check
Yes
Stop

Fig. 3: Flow chart of genetic algorithm

Step 10: The string that generates the minimum total generation cost is the solution of the problem.

The procedure of Genetic Algorithm (GA) can be summarized in the flow chart shown in Fig. 3.

RESULTS AND DISCUSSION

To verify the effectiveness of the proposed algorithm, a six unit thermal power generating plant was tested. The proposed algorithm has been implemented in Matlab language. The proposed algorithm is applied to 26 buses, 6 generating units with generator constraints, ramp rate limits and transmission losses (Saadat, 1999). The results obtained from the proposed method will be compared with the outcomes obtained from the conventional Lambda Iteration and PSO Method in terms of the solution quality and computation efficiency. The fuel cost data and ramp rate limits of the six thermal generating units were shown in Table 1. The load demand for 24 h is shown in Table 2. B-loss coefficients of 6-units system is given in Eq. 17. Table 3 shows the optimal scheduling of all generating units, power loss and total fuel cost for 24 h by using PSO Technique. Table 4 shows the optimal scheduling of all generating units, power loss and total fuel cost for 24 h by using genetic algorithm and Table 5 shows the total fuel cost comparison between Lambda Iteration, PSO and GA Method. Figure 4-9 show the relation between fuel cost of each unit and 24 h by the Lambda Iteration, PSO and GA Method. Some parameters must be assigned for the use of GA to solve the ELD problems as follows:

- Population size = 20
- Maximum number of generations = 100
- Crossover probability = 0.8
- Mutation probability = 0.05

And the parameters used in PSO technique to solve the ELD problem as follows:

- Population size = 20
- Maximum number of iterations = 100
- Acceleration constants $C_1 = 2.0$ and $C_2 = 2.0$
- Inertia weight parameters $\omega_{max} = 0.9$ and $\omega_{min} = 0.4$

$$B_{ij} = 10^{-3}\begin{bmatrix} 1.7 & 1.2 & 0.7 & -0.1 & -0.5 & -0.2 \\ 1.2 & 1.4 & 0.9 & 0.1 & -0.6 & -0.1 \\ 0.7 & 0.9 & 3.1 & 0.0 & -1.0 & -0.6 \\ -0.1 & 0.1 & 0.0 & 0.24 & -0.6 & -0.8 \\ -0.5 & -0.6 & -0.1 & -0.6 & 12.9 & -0.2 \\ -0.2 & -0.1 & -0.6 & -0.8 & -0.2 & 15.0 \end{bmatrix} \quad (17)$$

Figure 4-9 show the relation between fuel cost for each unit and 24 h by Lambda Iteration, PSO and GA

Table 1: Fuel cost coefficients and ramp rate limits of 6 thermal units system

Unit	a_i (\$/MW)	b_i (\$/M/W)	c_i (\$)	$P_{i,min}$ (MW)	$P_{i,max}$ (MW)	UR_i (MW/H)	DR_i (MW/H)
1	0.0070	7.0	240	100	500	80	120
2	0.0095	10.0	200	50	200	50	90
3	0.0090	8.5	220	80	300	65	100
4	0.0090	11.0	200	50	150	50	90
5	0.0080	10.5	220	50	200	50	90
6	0.0075	12.0	190	50	120	50	90

Table 2: Load demand for 24 h of 6-units system

Time (h)	Load (MW)	Time (h)	Load (MW)	Time (h)	Load (MW)	Time (h)	Load (MW)
1	955	7	989	13	1190	19	1159
2	942	8	1023	14	1251	20	1092
3	935	9	1126	15	1263	21	1023
4	930	10	1150	16	1250	22	984
5	935	11	1201	17	1221	23	975
6	963	12	1235	18	1202	24	960

Table 3: Output powers, power losses and total fuel cost for 24 h by PSO Method of 6-units system

Time (h)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	P_6 (MW)	Loss (MW)	Fuel cost (\$)
1	381.5	120.8	210.4	86.5	112.1	50.0	6.53	11410.86
2	375.6	118.3	208.2	84.9	111.2	50.0	6.35	11248.50
3	372.1	116.8	207.0	84.5	110.6	50.0	6.25	11161.44
4	369.6	115.8	206.1	84.3	110.2	50.0	6.17	11099.41
5	372.1	116.8	207.0	84.5	110.6	50.0	6.25	11161.44
6	384.9	122.2	211.6	87.7	113.0	50.0	6.64	11511.17
7	394.9	126.2	215.8	92.0	116.8	50.0	7.00	11838.94
8	399.0	133.7	222.1	96.2	122.7	56.4	7.38	12270.52
9	420.7	145.6	239.2	114.8	140.7	73.2	8.57	13599.88
10	427.7	148.1	243.1	118.8	143.3	77.7	8.85	13914.45
11	443.1	154.9	247.8	127.5	151.0	85.9	9.50	14588.85
12	452.3	160.5	251.5	133.1	155.4	91.9	9.95	15042.84
13	439.1	153.2	246.5	125.5	150.2	84.4	9.37	14442.65
14	456.1	162.7	254.3	136.3	157.9	93.6	10.18	15257.49
15	458.8	164.5	255.7	138.9	159.2	95.9	10.32	15419.10
16	455.6	162.5	254.1	136.2	157.9	93.6	10.16	15244.01
17	447.6	158.8	250.4	129.9	153.7	90.0	9.78	14855.29
18	443.5	155.0	248.0	127.6	151.1	86.0	9.97	14602.16
19	430.7	149.6	244.1	120.3	144.2	78.8	8.15	14032.85
20	414.3	141.9	233.4	109.2	133.8	67.2	8.38	13157.51
21	399.0	133.7	222.1	96.2	122.7	56.4	7.38	12270.52
22	393.7	125.3	214.9	90.9	115.9	50.0	6.94	11775.78
23	390.2	124.3	213.6	89.0	114.3	50.0	6.82	11662.16
24	383.5	121.5	211.0	87.4	112.9	50.0	6.60	11473.52
Total fuel cost (\$)								313041.40

Table 4: Output powers, power losses and total fuel cost for 24 h by GA of 6-units system

Time (h)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	P_6 (MW)	Loss (MW)	Fuel cost (\$)
1	378.4	118.4	210.7	85.4	118.4	50.0	6.58	11411.42
2	373.2	116.0	207.8	84.7	116.4	50.0	6.38	11249.19
3	371.0	114.8	206.1	83.7	115.3	50.0	6.28	11162.06
4	369.3	113.8	205.1	82.9	114.8	50.0	6.21	11099.99
5	371.0	114.8	206.1	83.7	115.3	50.0	6.28	11162.06
6	381.3	119.8	212.1	86.5	119.7	50.0	6.69	11511.66
7	388.9	125.0	217.1	90.8	123.9	50.0	7.06	11838.99
8	395.8	132.8	222.0	97.7	126.3	55.5	7.38	12270.42
9	422.5	147.3	239.5	114.3	138.0	72.7	8.57	13599.96
10	427.1	153.0	243.5	118.8	140.0	76.2	8.86	13914.33
11	438.9	161.9	252.0	128.5	145.8	83.1	9.51	14588.41
12	446.2	166.8	257.8	134.6	150.4	89.0	9.95	15042.04

Table 4: Continue

Time (h)	P1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	P_6 (MW)	Loss (MW)	Fuel cost ($)
13	436.9	160.1	250.1	126.2	144.2	81.5	9.37	14442.43
14	450.2	169.5	260.1	136.7	152.7	91.7	10.18	15256.81
15	452.5	171.9	261.6	139.6	154.2	93.2	10.33	15418.34
16	450.0	169.4	259.9	136.5	152.6	91.5	10.16	15243.36
17	443.0	164.9	255.7	131.9	148.2	86.7	9.76	14854.86
18	439.1	162.1	252.2	128.7	145.9	83.2	9.52	14601.71
19	428.9	154.9	244.5	120.8	141.1	77.4	8.96	14032.65
20	411.9	142.0	236.1	108.4	135.3	66.2	8.20	13157.46
21	395.8	132.8	222.0	97.7	126.3	55.5	7.38	12270.42
22	387.9	124.1	215.9	89.8	123.0	50.0	6.99	11775.86
23	385.4	122.1	214.6	88.1	121.5	50.0	6.87	11662.42
24	380.3	119.0	211.7	86.2	119.2	50.0	6.65	11474.06
Total fuel cost ($)								313040.90

Table 5: Total fuel cost comparison between proposed GA, Lambda Iteration and PSO Method of 6-units system

Method	Total fuel cost ($)
Lambda Iteration	313045.50
Particle Swarm Optimization (PSO)	313041.40
Genetic Algorithm (GA)	313040.90

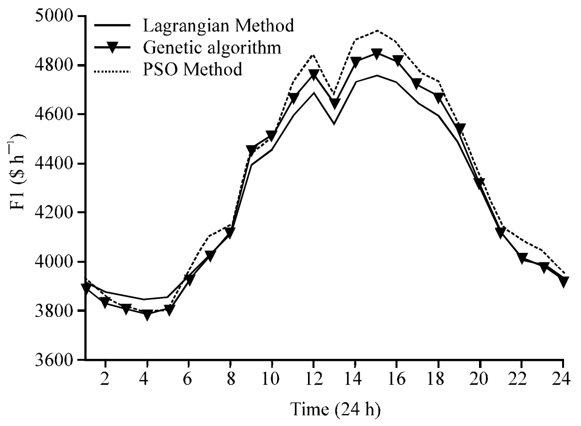

Fig. 4: Fuel cost of unit 1 vs. 24 h by the three used methods

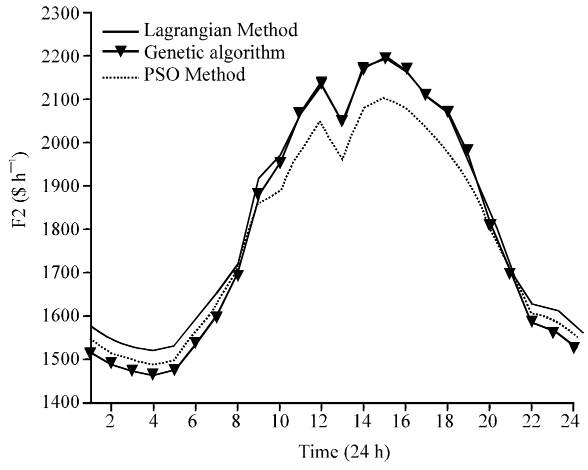

Fig. 5: Fuel cost of unit 2 vs. 24 h by the three used methods

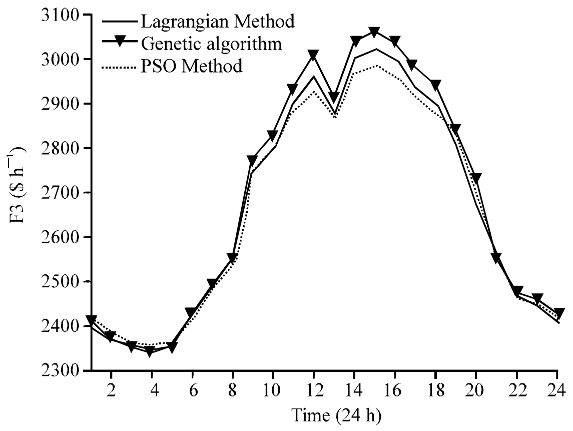

Fig. 6: Fuel cost of unit 3 vs. 24 h by the three used methods

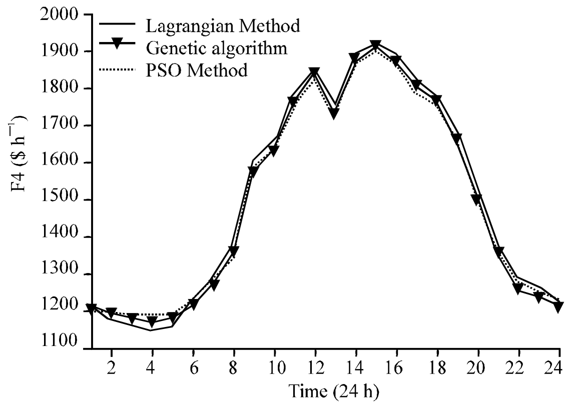

Fig. 7: Fuel cost of unit 4 vs. 24 h by the three used methods

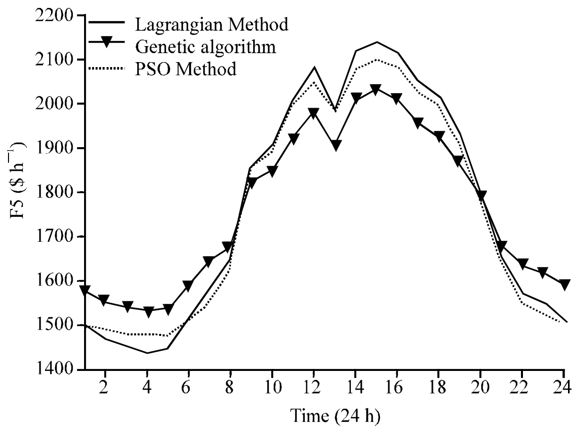

Fig. 8: Fuel cost of unit 5 vs. 24 h by the three used methods

method. From Fig. 4-9, it can see that the fuel cost for each generating unit obtained by the three techniques is different due to different distribution of powers on 6-units but the total fuel cost per hour is the same.

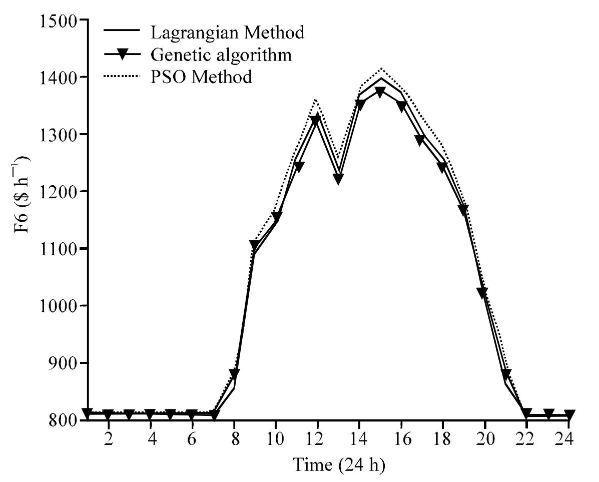

Fig. 9: Fuel cost of unit 6 vs. 24 h by the three used methods

CONCLUSION

In this study, Genetic Algorithm (GA) is used to solve the ELD problem. The proposed algorithm has been successfully implemented for solving the ELD problem of a power system consists of 6 units with different constraints such as real power balance, generator power limits and ramp rate limits. From the tabulated results, it is clear that the total fuel cost obtained by GA is comparatively less compared to other methods. GA approach gives high quality solutions with fast convergence characteristic compared to the Lambda Iteration Method. The Lambda Iteration Method is also applicable but it can converge to the minimum generation cost after so many iterations. So, the computational time of the Lambda Iteration Method is much greater than the proposed algorithm. Simulation results demonstrate that the proposed method is powerful and practical tool for obtaining global minimum or near global minimum of total fuel cost.

REFERENCES

Bakirtzis, A., V. Petridis and S. Kazarlis, 1994. Genetic algorithm solution to the economic dispatch problem. IEE Proc. Gene. Trans. Distribut., 141: 377-382.

Chen, P.H. and H.C. Chang, 1995. Large-scale economic dispatch approach by genetic algorithm. IEEE Trans. Power Syst., 10: 1919-1926.

Gaing, Z.L., 2004. Constrained dynamic economic dispatch solution using particle swarm optimization. Proceedings of the IEEE Power Engineering Society General Meeting, June 6-10, 2004, Denver, CO, USA., pp: 153-158.

Goldberg, D.E., 1990. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co. Inc., Boston, MA.

Jeyakumar, D.N., T. Jayabarathi and T. Raghunathan, 2006. Particle swarm optimization for various types of economic dispatch problems. Int. J. Elec. Power Energy Syst., 28: 36-42.

Lin, W.M., F.S. Cheng and M.T. Tsay, 2002. An improved tabu search for economic dispatch with multiple minima. IEEE Trans. Power Syst., 17: 108-112.

Mimoun, Y., M. Rahli and L.A. Koridak, 2006. Economic power dispatch using evolutionary algorithm. J. Elect. Eng., 57: 211-217.

Momoh, J.A., R. Adapa and M.E. El-Hawary, 1999. A review of selected optimal power flow literature to 1993. Part 1. Non linear and quadratic programming approaches. IEEE Trans. Power Syst., 14: 96-104.

Nanda, J., A. Sachan, L. Pradhan, M.L. Kothari, A.K. Rao, L.L. Lai and M. Prasad, 1997. Application of artificial neural network to economic load dispatch. Proceedings of the 4th International Conference on Advances in Power System Control, Operation and Management, Volume: 2, November 11-14, 1997, Hong Kong, pp: 707-711.

Park, J.B., K.S. Lee, J.R. Shin and K.Y. Lee, 2005. A particle swarm optimization for economic dispatch with nonsmooth cost functions. IEEE Trans. Power Syst., 20: 34-42.

Report, I.C., 1971. Present practices in the economic operation of power systems. IEEE Trans. Power Appa. Syst., PAS-90: 1768-1775.

Saadat, H., 1999. Power System Analysis. McGraw-Hill Book Co. Inc., New York.

Salama, M.M., 1999. Economic control for generation in thermal power system. Energy Convers. Manage., 40: 669-681.

Sheble, G.B. and K. Brittig, 1995. Refined genetic algorithm-economic dispatch example. IEEE Trans. Power Syst., 10: 117-124.

Shi, Y. and R.C. Eberhart, 1998. Parameters selection in particle swarm optimization. Evol. Programming VII, 1147: 591-600.

Sinha, N., R. Chakrabarti and P.K. Chattopadhyay, 2003. Evolutionary programming techniques for economic load dispatch. IEEE Trans. Evolutionary Comput., 7: 83-94.

Song, Y.H., F. Li, R. Morgen and D.T.Y. Cheng, 1995. Comparison studies of genetic algorithms in power system economic dispatch. Power Syst. Technol., 19: 28-33.

Tippayachai, J., W. Ongsakul and I. Ngamroo, 2002. Parallel micro genetic algorithm for constrained economic dispatch. IEEE Trans. Power Syst., 17: 790-797.

Walters, D.C. and G.B. Sheble, 1993. Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans. Power Syst., 8: 1325-1332.

Wong, K.P. and C.C. Fung, 1993. Simulated annealing based economic dispatch algorithm. IEE Proc., 140: 509-515.

Wood, A.J. and B.F. Wollenberg, 1984. Power Generation, Operation and Control. John Wiley and Sons, New York, USA.