World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:4, No:3, 2010

Application of Computational Intelligence Techniques for Economic Load Dispatch

S.C. Swain, S. Panda, A.K. Mohanty, C. Ardil

***Abstract*—**This paper presents the applications of computational intelligence techniques to economic load dispatch problems. The fuel cost equation of a thermal plant is generally expressed as continuous quadratic equation. In real situations the fuel cost equations can be discontinuous. In view of the above, both continuous and discontinuous fuel cost equations are considered in the present paper. First, genetic algorithm optimization technique is applied to a 6-generator 26-bus test system having continuous fuel cost equations. Results are compared to conventional quadratic programming method to show the superiority of the proposed computational intelligence technique. Further, a 10-generator system each with three fuel options distributed in three areas is considered and particle swarm optimization algorithm is employed to minimize the cost of generation. To show the superiority of the proposed approach, the results are compared with other published methods.

***Keywords*—**Economic Load Dispatch, Continuous Fuel Cost, Quadratic Programming, Real-Coded Genetic Algorithm, Discontinuous Fuel Cost, Particle Swarm Optimization.

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

I. Introduction

ECOMONIC load dispatch is defined as the process of allocating generation levels to the generating units in the mix, so that the system load is supplied entirely and most economically [1]. The objective of the economic dispatch problem is to calculate the output power of every generating unit so that all demands are satisfied at minimum cost, while satisfying different technical constraints of the network and the generators. 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 of the generators equals the total power that must be delivered. It is an important daily optimization task in the operation of a power system [2].

S. Swain is working as an Assistant Professor in the Electrical Engineering Department, School of Technology, KIIT University, Bhubaneswar, Orissa, India (e-mail:scs_132@rediffmail.com).

S. Panda is working as a Professor in the Department of Electrical and Electronics Engineering, NIST, Berhampur, Orissa, India, Pin: 761008. (e-mail: panda_sidhartha@rediffmail.com)

A.K. Mohanty is working as a Professor Emeritus in the Electrical Engineering Department, School of Technology, KIIT University, Bhubaneswar, Orissa, India, (e-mail: dhisi1@rediffmail.com)..

C. Ardil is with National Academy of Aviation, AZ1045, Baku, Azerbaijan, Bina, 25th km, NAA (e-mail: cemalardil@gmail.com).

Several optimization techniques have been applied to solve the ED problem. To solve economic dispatch problem effectively, most algorithms require the incremental cost curves to be of monotonically smooth increasing nature and continuous [3-6]. For the generating units, which actually having non-monotonically incremental cost curves, the conventional method ignores or flattens out the portions of the incremental cost curve that are not continuous or monotonically increasing. Hence, inaccurate dispatch result may be obtained. To obtain accurate dispatch results, the approaches without restriction on the shape of fuel cost functions are necessary [7-8]. Most of conventional methods suffer from the convergence problem, and always get trap in the local minimum. Moreover, some techniques face the dimensionality problem especially when solving the large-scale system.

In recent years, one of the most promising research fields has been "Evolutionary Techniques", an area utilizing analogies with nature or social systems. Evolutionary techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable objective functions. Several modern heuristic tools have evolved in the last two decades that facilitate solving optimization problems that were previously difficult or impossible to solve. These tools include evolutionary computation, simulated annealing, tabu search, particle swarm, etc. Recently, genetic algorithm (GA) and particle swarm optimization (PSO) techniques appeared as promising algorithms for handling the optimization problems [9]. These techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable cost functions.

Genetic Algorithm (GA) can be viewed as a general-purpose search method, an optimization method, or a learning mechanism, based loosely on Darwinian principles of biological evolution, reproduction and ''the survival of the fittest'' [10]. GA maintains a set of candidate solutions called population and repeatedly modifies them. At each step, the GA selects individuals from the current population to be parents and uses them to produce the children for the next generation. In general, the fittest individuals of any population tend to reproduce and survive to the next generation, thus improving successive generations. However, inferior individuals can, by chance, survive and also reproduce. GA is well suited to and has been extensively applied to solve complex design optimization problems because it can handle

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

both discrete and continuous variables, non-linear objective and constrain functions without requiring gradient information. It has been reported in the literature that Real-Coded Genetic Algorithm (RCGA) is more efficient in terms of CPU time and offers higher precision with more consistent results [11-14].

Particle Swarm Optimization (PSO) is inspired by the ability of flocks of birds, schools of fish, and herds of animals to adapt to their environment, find rich sources of food, and avoid predators by implementing an information sharing approach. PSO technique was invented in the mid 1990s while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a sociocognitive study investigating the notion of collective intelligence in biological populations [15]. In PSO, a set of randomly generated solutions propagates in the design space towards the optimal solution over a number of iterations based on large amount of information about the design space that is assimilated and shared by all members of the swarm [16].

Techniques such as PSO and GA are inspired by nature, and have proved themselves to be effective solutions to optimization problems. It has been reported in the literature that, both PSO and GA optimization techniques can be used for optimization problems giving almost similar results [9]. This paper presents the applications of both GA and PSO to economic load dispatch problems. Both continuous and discontinuous fuel cost equations are considered in the present paper. First, RCGA optimization technique is applied to a 6-generator 26-bus test system having continuous fuel cost equations and the results are compared to conventional quadratic programming method to show its superiority. Further, PSO is employed to minimize the cost of generation of a 10-generator system each with discontinuous fuel cost equations and the results are compared with other published methods.

II. Problem Statement

The basic economic dispatch problem can described mathematically as a minimization of problem of minimizing the total fuel cost of all committed plants subject to the constraints [1].

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

Subject to the constraints

$$\sum_{i=1}^{N} P_i - P_D - P_L = 0 \tag{2}$$

$$P_{i\min} \le P_i \le P_{i\max}\,,\ i = 1,2,\ldots\ldots,N \tag{3}$$

Where

F = Total operating cost

N = Number of generating units

P_i = Power output of i th generating unit

$F_i\,(P_i)$ = Fuel cost function of i th generating unit

P_D = Total load demand

P_L = Total losses

$P_{i\,min}$ = Minimum out put power limit of i th generating unit

$P_{i\,max}$ = Maximum out put power limit of i th generating unit

The total fuel cost is to be minimized subject to the constraints. The transmission loss can be determined form B_{mn} coefficients.

The conditions for optimality can be obtained by using Lagrangian multipliers method and Kuhn tucker conditions as follows:

$$2a_i P_i + b_i = \lambda(1 - 2\sum_{j=1}^{N} B_{ij})\,,\ i = 1,2,\ldots\ldots,N \tag{4}$$

With the following constraints

$$\sum_{i=1}^{N} P_i = P_D + P_L \tag{5}$$

$$P_L = \sum_{i=1}^{N} \sum_{i=1}^{N} P_i B_{ij} P_j \tag{6}$$

$$P_{i\min} \le P_i \le P_{i\max} \tag{7}$$

The following steps are followed to solve the economic load dispatch problem with the constraints:

Step-1:

Allocate lower limit of each plant as generation, evaluate the transmission loss and incremental loss coefficients and update the demand.

$$P_i = P_{i\min}\,,\ X_i = 1 - \sum_{j=1}^{N} P_i B_{ij}\,,\ P_D^{new} = P_D + P_L^{old} \tag{8}$$

Step-2:

Apply quadratic programming to determine the allocation P_i^{new} of each plant.

If the generation hits the limit, it should be fixed to that limit and the remaining plants only should be considered for next iteration.

Step-3:

Check for the convergence

$$\left| \sum_{i=1}^{N} P_i - P_D^{new} - P_L \right| \le \varepsilon \tag{9}$$

Where ε is the tolerance. Repeat until the convergence criteria is meet.

A brief description about the quadratic programming method is presented in the next section.

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

III. QUADRATIC PROGRAMMING METHOD

A linearly constrained optimization problem with a quadratic objective function is called a Quadratic Program (QP). Due to its numerous applications; quadratic programming is often viewed as a discipline in and of itself. Quadratic programming is an efficient optimization technique to trace the global minimum if the objective function is quadratic and the constraints are linear. Quadratic programming is used recursively from the lowest incremental cost regions to highest incremental cost region to find the optimum allocation. Once the limits are obtained and the data are rearranged in such a manner that the incremental cost limits of all the plants are in ascending order.

The general quadratic program can be written as:

$$\text{Minimize} \quad f(x) = cx + \frac{1}{2}x^T Qx \tag{10}$$

$$\text{Subject to} \quad Ax \leq b \text{ and } x \geq 0 \tag{11}$$

Where **c** is an *n*-dimensional row vector describing the coefficients of the linear terms in the objective function, and **Q** is an $(n \times n)$ symmetric matrix describing the coefficients of the quadratic terms. If a constant term exists it is dropped from the model. As in linear programming, the decision variables are denoted by the *n*-dimensional column vector **x**, and the constraints are defined by an $(m \times n)$ **A** matrix and an *m*-dimensional column vector **b** of right-hand-side coefficients. We assume that a feasible solution exists and that the constraint region is bounded. When the objective function *f*(**x**) is strictly convex for all feasible points the problem has a unique local minimum which is also the global minimum. A sufficient condition to guarantee strictly convexity is for **Q** to be positive definite.

If there are only equality constraints, then the QP can be solved by a linear system. Otherwise, a variety of methods for solving the QP are commonly used, namely; interior point, active set, conjugate gradient, extensions of the simplex algorithm etc. The direction search algorithm is minor variation of quadratic programming for discontinuous search space. For every demand the following search mechanism is followed between lower and upper limits of those particular plants. For meeting any demand the algorithm is explained in the following steps:

1) Assume all the plants are operating at lowest incremental cost limits.
2) Substitute $P_i = L_i + (U_i - L_i)X_i$, where $0 < X_i < 1$ and make the objective function quadratic and make the constraints linear by omitting the higher order terms.
3) Solve the ELD using quadratic programming recursively to find the allocation and incremental cost for each plant within limits of that plant.
4) If there is no limit violation for any plant for that particular piece, then it is a local solution.
5) If for any allocation for a plant, it is violating the limit, it should be fixed to that limit and the remaining plants only should be considered for next iteration.
6) Repeat steps 2, 3, and 4 till a solution is achieved within a specified tolerance.

IV. GENETIC ALGORITHM APPROACH

A. *Overview of Real Coded Genetic Algorithm*

Genetic Algorithm (GA) can be viewed as a general-purpose search method, an optimization method, or a learning mechanism, based loosely on Darwinian principles of biological evolution, reproduction and “the survival of the fittest.” GA maintains a set of candidate solutions called population and repeatedly modifies them. At each step, the GA selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Candidate solutions are usually represented as strings of fixed length, called chromosomes.

Given a random initial population GA operates in cycles called generations, as follows [10]:

- Each member of the population is evaluated using an objective function or fitness function.
- The population undergoes reproduction in a number of iterations. One or more parents are chosen stochastically, but strings with higher fitness values have higher probability of contributing an offspring.
- Genetic operators, such as crossover and mutation, are applied to parents to produce offspring.
- The offspring are inserted into the population and the process is repeated.

Over successive generations, the population “evolves” toward an optimal solution. GA can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear. GA has been used to solve difficult engineering problems that are complex and difficult to solve by conventional optimization methods.

B. *Implementation of RCGA*

Implementation of GA requires the determination of six fundamental issues: chromosome representation, selection function, the genetic operators, initialization, termination and evaluation function. Brief descriptions about these issues are provided in the following sections [11-14].

i. Chromosome Representation

Chromosome representation scheme determines how the problem is structured in the GA and also determines the genetic operators that are used. Each individual or chromosome is made up of a sequence of genes. Various types of representations of an individual or chromosome are: binary digits, floating point numbers, integers, real values, matrices, etc. Generally natural representations are more efficient and

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

produce better solutions. Real-coded representation is more efficient in terms of CPU time and offers higher precision with more consistent results.

ii. Selection Function

To produce successive generations, selection of individuals plays a very significant role in a genetic algorithm. The selection function determines which of the individuals will survive and move on to the next generation. A probabilistic selection is performed based upon the individual's fitness such that the superior individuals have more chance of being selected. There are several schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, normal geometric, elitist models and ranking methods.

The selection approach assigns a probability of selection P_j to each individuals based on its fitness value. In the present study, normalized geometric selection function has been used. In normalized geometric ranking, the probability of selecting an individual P_i is defined as:

$$Pi = q^{'}\left(1-q\right)^{r-1} \tag{12}$$

$$q^{'} = \frac{q}{1-(1-q)^{P}} \tag{13}$$

where,

q = probability of selecting the best individual

r = rank of the individual (with best equals 1)

P = population size

iii. Genetic Operators

The basic search mechanism of the GA is provided by the genetic operators. There are two basic types of operators: crossover and mutation. These operators are used to produce new solutions based on existing solutions in the population. Crossover takes two individuals to be parents and produces two new individuals while mutation alters one individual to produce a single new solution. The following genetic operators are usually employed: simple crossover, arithmetic crossover and heuristic crossover as crossover operator and uniform mutation, non-uniform mutation, multi-non-uniform mutation, boundary mutation as mutation operator. Arithmetic crossover and non-uniform mutation are employed in the present study as genetic operators. Crossover generates a random number r from a uniform distribution from 1 to m and creates two new individuals by using equations:

$$x_i^{'} = \begin{cases} x_i, \text{ if } i < r \\ y_i \text{ otherwise} \end{cases} \tag{14}$$

$$y_i^{'} = \begin{cases} y_i, \text{ if } i < r \\ x_i \text{ otherwise} \end{cases} \tag{15}$$

Arithmetic crossover produces two complimentary linear combinations of the parents, where r = U (0, 1):

$$\bar{X}' = r\,\bar{X} + (1-r)\,\bar{Y} \tag{16}$$

$$\bar{Y}' = r\,\bar{Y} + (1-r)\,\bar{X} \tag{17}$$

Non-uniform mutation randomly selects one variable j and sets it equal to a non-uniform random number.

$$x_i' = \begin{cases} x_i + (b_i - x_i)\, f(G) & \text{if } r_1 < 0.5, \\ x_i + (x_i + a_i)\, f(G) & \text{if } r_1 \geq 0.5, \\ x_i, & \text{otherwise} \end{cases} \tag{18}$$

where,

$$f(G) = (r_2(1 - \frac{G}{G_{\max}}))^{b} \tag{19}$$

r_1, r_2 = uniform random nos. between 0 to 1.

G = current generation.

$G_{\max}$ = maximum no. of generations.

b = shape parameter.

iv. Initialization, Termination and Evaluation Function

An initial population is needed to start the genetic algorithm procedure. The initial population can be randomly generated or can be taken from other methods.

GA moves from generation to generation until a stopping criterion is met. The stopping criterion could be maximum number of generations, population convergence criteria, lack of improvement in the best solution over a specified number of generations or target value for the objective function.

Evaluation functions or objective functions of many forms can be used in a GA so that the function can map the population into a partially ordered set.

v. Parameter Selection for RCGA

For different problems, it is possible that the same parameters for GA do not give the best solution and so these can be changed according to the situation. The parameters employed for the implementations of RCGA in the present study are given in Table I. Optimization were performed with the total number of generations set to 100. The optimization processes is run 20 times and best among the 20 runs are taken as the final solutions.

TABLE I: PARAMETERS USED IN RCGA

Parameter	Value/Type
Maximum generations	100
Population size	50
Type of selection	Normal geometric [0 0.08]
Type of crossover	Arithmetic [2]
Type of mutation	Nonuniform [2 100 3]
Termination method	Maximum generation

C. Proposed RCGA Approach

The following steps are followed for the implementation of GA for economic load dispatch problems.

1) Select the plant having maximum capacity and range as a reference plant.
2) Fix the reference plant allocation by equation (5) and (6).
3) Convert the constrained optimization problem as an unconstrained problem by penalty function method as:

$$\min \quad F = \sum_{i=1}^{N} F_i(P_i) + 1000 * \left| \left(\sum_{i=1}^{N} P_i - P_D - \sum_{i=1}^{N}\sum_{j=1}^{N} B_{ij} P_i P_j \right) \right| \quad (20)$$

4) Apply RCGA to minimize *F*.

V. Particle Swarm Optimization Approach

A. Overview of Particle Swarm Optimization

The PSO method is a member of wide category of swarm intelligence methods for solving the optimization problems. It is a population based search algorithm where each individual is referred to as particle and represents a candidate solution. Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also to the flying experience of the other particles. In PSO each particles strive to improve themselves by imitating traits from their successful peers. Further, each particle has a memory and hence it is capable of remembering the best position in the search space ever visited by it. The position corresponding to the best fitness is known as *pbest* and the overall best out of all the particles in the population is called *gbest* [15-16].

The modified velocity and position of each particle can be calculated using the current velocity and the distances from the $pbest_{j,g}$ to $gbest_g$ as shown in the following formulas [11, 17-20]:

$$v_{j,g}^{(t+1)} = w * v_{j,g}^{(t)} + c_1 * r_1(\ ) * (pbest_{j,g} - x_{j,g}^{(t)}) + c_2 * r_2(\ ) * (gbest_g - x_{j,g}^{(t)}) \quad (21)$$

$$x_{j,g}^{(t+1)} = x_{j,g}^{(t)} + v_{j,g}^{(t+1)} \quad (22)$$

With $j = 1,2,...,n$ and $g = 1,2,...,m$

where,

n = number of particles in the swarm

m = number of components for the vectors v_j and x_j

t = number of iterations (generations)

$v_{j,g}^{(t)}$ = the *g*-th component of the velocity of particle *j* at iteration t, $v_g^{\min} \le v_{j,g}^{(t)} \le v_g^{\max}$;

w = inertia weight factor

c_1, c_2 = cognitive and social acceleration factors respectively

r_1, r_2 = random numbers uniformly distributed in the range (0, 1)

$x_{j,g}^{(t)}$ = the *g*-th component of the position of particle *j* at iteration *t*

$pbest_j$ = *pbest* of particle *j*

$gbest$ = *gbest* of the group

The *j*-th particle in the swarm is represented by a *d*-dimensional vector $x_j = (x_{j,1}, x_{j,2}, \ldots\ldots, x_{j,d})$ and its rate of position change (velocity) is denoted by another *d*-dimensional vector $v_j = (v_{j,1}, v_{j,2}, \ldots\ldots, v_{j,d})$. The best previous position of the *j*-th particle is represented as $pbest_j = (pbest_{j,1}, pbest_{j,2}, \ldots\ldots, pbest_{j,d})$. The index of best particle among all of the particles in the swarm is represented by the $gbest_g$. In PSO, each particle moves in the search space with a velocity according to its own previous best solution and its group's previous best solution. The velocity update in a PSO consists of three parts; namely momentum, cognitive and social parts. The balance among these parts determines the performance of a PSO algorithm. The parameters c_1 and c_2 determine the relative pull of *pbest* and *gbest* and the parameters r_1 and r_2 help in stochastically varying these pulls. In the above equations, superscripts denote the iteration number.

B. Parameter Selection for PSO

For the implementation of PSO, several parameters are required to be specified, such as c_1 and c_2 (cognitive and social acceleration factors, respectively), initial inertia weights, swarm size, and stopping criteria. These parameters should be selected carefully for efficient performance of PSO. 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 before being tugged back. On the other hand, 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, w, provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run [11, 15-20]. The parameters employed for the implementations of PSO in the present study are given in Table II.

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

TABLE II: PARAMETERS USED IN PSO

Parameter	Value/Type
Maximum generations	100
Swarm size	20
Cognitive factors (c_1) & social acceleration factors (c_2)	c_1 =2.0 c_2=2.0
Inertia weights	w_{start}, =0.9 w_{end} = 0.4

VI. RESULTS AND DISCUSSIONS

A. Numerical Example 1

First, continuous quadratic cost curve for the plants is considered. The system consists of 26 bus, 6 units, and the demand of the system was divided into 12 small intervals as shown in Fig. 1. Generating units' data are given in Table 3.1. The cost function coefficients along with minimum and maximum generation capacity for each fuel option are given in Table III. Table IV, shows the optimal generators' power outputs for each hour including their corresponding fuel costs using quadratic programming method. Total production cost of 12 intervals is $156065.8. Table V, shows the same using RCGA method. Total production cost of 12 intervals is $151008. It is clear from Table IV and V that RCGA gives better solutions.

B. Numerical Example 2

A test system-2 having ten plants each with three fuel options distributed in three areas is considered. The cost function coefficients along with minimum and maximum generation capacity for each fuel option are given in Table VI.

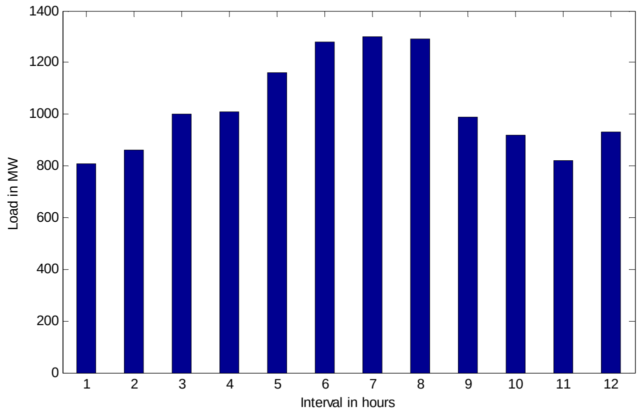

Fig. 1. Load pattern of numerical example 1

The system is found to have minimum and maximum generation capacity of 1353 MW and 3695 MW, respectively. The load demand is assumed to vary between 2400 MW and 2700 MW in steps of 50 MW. The results obtained by the proposed PSO method are given in Table VII. The results are compared in Table VIII with the results obtained by other methods to show its superiority.

TABLE III: DATA FOR EXAMPLE - 1: 26-BUS 6-UNIT TEST SYSTEM

Unit/Cost	a ($/MW^{2}h)	b ($/MWh)	c ($/h)	P_{min} (MW)	P_{max} (MW)
Unit-1	0.007	7	240	100	500
Unit-2	0.0095	10	200	50	200
Unit-3	0.009	8.5	220	80	300
Unit-4	0.009	11	200	50	150
Unit-5	0.008	10.5	220	50	200
Unit-6	0.0075	12	120	50	120

TABLE IV: RESULTS OF QUADRATIC PROGRAMMING FOR EXAMPLE - 1: 26-BUS, 6-UNIT TEST SYSTEM

U/T	1	2	3	4	5	6	7	8	9	10	11	12
P_{g1}	350.315	363.153	399.336	401.934	436.308	462.628	467.039	464.833	396.740	378.617	352.879	381.200
P_{g2}	102.124	111.506	137.931	139.827	164.950	184.186	187.408	185.796	136.036	122.803	103.998	124.691
P_{g3}	183.725	193.286	220.174	222.101	247.633	267.157	270.426	268.791	218.247	204.788	185.636	206.708
P_{g4}	51.3537	60.8204	87.4125	89.3168	114.364	133.424	136.611	135.017	85.5089	72.2016	53.2457	74.1007
P_{g5}	84.4816	94.9211	124.221	126.317	153.650	174.333	177.787	176.06	122.124	107.465	86.5684	109.558
P_{g6}	50.00	50.00	50.00	50.00	69.6387	91.3661	94.9925	93.1791	50.00	50.00	50	50.00
Total cost in $	9902.8	10561.0	12479.0	12621.0	14812.0	16657.0	16972.0	16814.0	12339.0	11369.0	10033.0	11506.0

TABLE V: RESULTS OF RCGA FOR EXAMPLE - 1: 26-BUS, 6-UNIT TEST SYSTEM

U/T	1	2	3	4	5	6	7	8	9	10	11	12
P_{g1}	358.703	372.283	411.002	414.002	449.170	477.707	482.007	480.650	408.034	387.693	362.035	392.762
P_{g2}	101.352	111.093	135.375	140.499	164.256	184.537	188.681	185.869	136.231	124.439	103.507	123.946
P_{g3}	184.183	195.593	221.740	221.29	248.533	269.031	270.369	267.497	217.577	203.106	185.359	106.069
P_{g4}	50.6959	59.1026	87.9846	87.5271	112.636	132.6	135.418	135.748	84.4349	72.1771	52.1834	72.3675
P_{g5}	79.7693	88.6800	117.111	119.788	150.242	168.121	171.386	170.557	116.305	101.551	81.8496	104.635
P_{g6}	50.0326	50.0185	50.0152	50.6013	67.3418	87.8866	93.4453	90.2401	50.1375	50.4198	50.1862	50.0615
Total cost $	9682.0	10306.0	12110.0	12243.0	14277.0	15969.0	16257.0	16113.0	11979.0	11069.0	9806.0	11197.0

TABLE VI: DATA FOR EXAMPLE - 2: 10-UNIT NEW ENGLAND TEST SYSTEM

Unit	Fuel option	a	b	c	P_{min}	P_{max}	Priority
1	1	0.002176	-0.3975	26.76	100	196	1
1	2	0.001861	-0.3059	21.13	196	250	2
2	1	0.00162	-0.198	13.65	50	114	1
2	2	0.001138	-0.03998	1.865	114	157	2
2	3	0.004194	-1.269	118.4	157	230	3
3	1	0.001457	-0.3116	39.79	200	332	1
3	2	0.00080351	0.03389	-2.876	332	388	2
3	3	0.00001176	0.4864	-59.14	388	500	3
4	1	0.001049	-0.03114	1.983	99	138	3
4	2	0.002758	-0.6348	52.85	138	200	2
4	3	0.005935	-2.338	266.8	200	265	1
5	1	0.001066	-0.08733	13.92	190	338	1
5	2	0.001597	-0.5206	99.76	338	407	2
5	3	0.0001498	0.4462	-53.99	407	490	3
6	1	0.001049	0.03114	1.983	85	138	3
6	2	0.002757	0.6348	52.85	138	200	2
6	3	0.005935	-2.338	266.3	200	265	1
7	1	0.001107	-0.1325	18.95	200	331	1
7	2	0.001165	-0.2267	43.77	331	391	2
7	3	0.0002454	0.3555	-43.55	391	500	3
8	1	0.001049	-0.03114	1.983	99	138	1
8	2	0.002758	-0.6348	52.85	138	200	2
8	3	0.005935	-2.338	266.8	200	265	3
9	1	0.007038	-0.04514	15.3	130	213	1
9	2	0.001554	-0.5675	88.53	213	370	3
9	3	0.0006121	-0.01817	14.23	370	440	2
10	1	0.001102	-0.09938	13.97	200	362	1
10	2	0.000042	0.5084	-61.13	362	407	2
10	3	0.001137	-0.2024	46.71	407	490	3

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

TABLE VII: RESULTS OF PSO FOR EXAMPLE - 2: 10-UNIT NEW ENGLAND TEST SYSTEM

Unit/Load	2400	2450	2500	2550	2600	2650	2700
1	189.7405	194.0906	206.5190	211.5316	216.5442	214.0190	218.2499
2	202.3427	204.5997	206.4573	208.6815	210.9058	209.7852	211.6626
3	253.8953	260.3920	265.7391	272.1416	278.5441	275.3187	280.7228
4	233.0456	234.6405	235.9531	237.5249	239.0967	238.3049	239.6315
5	241.8297	250.7094	258.0177	266.7686	275.5194	271.1110	278.4973
6	233.0456	234.6405	235.9531	237.5249	239.0967	238.3049	239.6315
7	253.2750	261.8258	268.8635	277.2903	285.7170	281.4718	288.5845
8	233.0456	234.6405	235.9531	237.5249	239.0967	238.3049	239.6315
9	320.3832	326.4744	331.4877	337.4906	343.4934	415.6581	428.5216
10	239.3969	247.9866	255.0562	263.5212	271.9861	267.7217	274.8667
Total cost	481.0326	502.9185	525.7588	549.3634	573.9008	598.4015	623.3292

TABLE VIII: COMPARISION OF RESULTS FOR EXAMPLE - 2: 10-UNIT NEW ENGLAND TEST SYSTEM

Load	Results of ref 21	Results of ref 22	Results of ref 23	Results of ref 24	Results of ref 25	Result of proposed method PSO
2400	488.46	487.91	481.72	481.73	481.72	**481.0326**
2500	526.16	**525.69**	526.24	526.23	526.238	525.7588
2600	**573.52**	574.28	574.38	574.39	574.38	573.9008
2700	625.22	623.81	626.25	623.8	623.809	**623.3292**

VII. Conclusion

This paper presents the applications of computational intelligence techniques to economic load dispatch problems considering both continuous and discontinuous fuel cost functions. First, a continuous fuel cost function is considered for a 26 bus, 6 unit test system and both conventional (quadratic programming method) and computational intelligence (real coded genetic algorithm) methods are applied to find the optimum generator allocation. It is seen that the results obtained by the computational intelligence method is better compared to the quadratic programming method. Further, a discontinuous fuel cost function is considered for a 10 unit New England test system and another computational intelligence technique (particle swarm optimization) is applied to find the optimum generator allocations. The results are compared with other published methods to show its superiority.

Reference

[1.] J. Wood and B. F. Wollenberg, “Power Generation Operation and Control,” 2nd edition, New York: Willey, 1996.

[2.] B. H. Chowdhury and S. Rahman, “A review of recent advances in economic dispatch,” *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1248-1259, November 1990.

[3.] A. Jiang and S. Ertem, “Economic dispatch with non-monotonically increasing incremental cost units and transmission system losses”,*IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 891-897, May 1995.

[4.] H.W. Dommel, “Optimal power dispatch”, *IEEE Transactions on Power Apparatus and Systems*, PAS93 No. 3, pp. 820–830, 1974.

[5.] C.O. Alsac, J. Bright, M. Paris, and Stott, “Developments in LP-based optimal power flow, *IEEE Transaction of Power Systems*”, Vol. 5 No. 3, pp. 697–711, 1990.

[6.] J. Nanda, D.P. Kothari, S.C. Srivastava, “New optimal power-dispatch algorithm using fletcher’s quadratic programming method”, *IEE Proceedings*, Vol. 136 No. 3, pp. 153–161, 1989.

[7.] 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, pp. 112-118, 1996.

Open Science Index, Electrical and Computer Engineering Vol:4, No:3, 2010 publications.waset.org/2222.pdf

[8.] T. Jayabarathi, G. Sadasivam and V. Ramachandran, "Evolutionary programming based economic dispatch of generators with prohibited operating zones," Electric Power Systems Research, Vol. 52, No. 3, pp. 261-266, 1999.
[9.] Sidhartha Panda and N. P. Padhy, "Comparison of Particle Swarm Optimization and Genetic Algorithm for FACTS-based Controller Design", *Applied Soft Computing*. vol. 8, issue 4, pp. 1418-1427, 2008.
[10.] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley, 1989.
[11.] Sidhartha Panda, S.C. Swain, P.K. Rautray, R. Mallik, G. Panda, "Design and analysis of SSSC-based supplementary damping controller", *Simulation Modelling Practice and Theory*, doi: 10.1016/j.simpat.2010.04.007.
[12.] Sidhartha Panda, C. Ardil, "Real-coded genetic algorithm for robust power system stabilizer design", *International Journal of Electrical, Computer and System Engineering*, vol. 2, no. 1, pp. 6-14, 2008.
[13.] Sidhartha Panda, S.C. Swain, A.K. Baliarsingh, "Real-coded genetic algorithm for robust coordinated design of excitation and SSSC-based controller", *Journal of Electrical Engineering*, vol. 8, issue 4, pp. 31-38, 2008.
[14.] S. Panda, S.C. Swain, A.K. Baliarsingh, C. Ardil, "Optimal supplementary damping controller design for TCSC employing RCGA", *International Journal of Computational Intelligence*, vol. 5, no. 1, pp. 36-45, 2009.
[15.] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", *Proceedings of the IEEE International. Conference on Neural Networks*, vol. 4, Piscataway, NJ, IEEE Service Center, 1995, pp. 1942–1948.
[16.] J. Kennedy, R. Eberhart, "Swarm Intelligence", Academic press, San Diego, CA, 2001.
[17.] Sidhartha Panda and N.P.Padhy, "Optimal location and controller design of STATCOM using particle swarm optimization", Journal of the Franklin Institute, vol.345, pp. 166-181, 2008.
[18.] Sidhartha Panda, N.P.Padhy, R.N.Patel, "Power system stability improvement by PSO optimized SSSC-based damping controller", Electric Power Components & Systems, vol. 36, no. 5, pp. 468-490, 2008.
[19.] S. Panda, N.P.Padhy and R.N.Patel, "Robust coordinated design of PSS and TCSC using PSO technique for power system stability enhancement", *Journal of Electrical Systems*, vol. 3, no. 2, pp. 109-123, 2007.
[20.] S. Panda, S. K. Tomar, R. Prasad, C. Ardil, "Reduction of linear time-invariant systems using routh-approximation and PSO", *International Journal of Applied Mathematics and Computer Sciences*, vol. 5, no. 2, pp. 82-89, 2009.
[21.] N. Ramaraj and K. Nagappan, "Analytical method to optimize generation schedule", Journal *of The Institution of Engineer*s (India), vol 66, p 240, 1987.
[22.] C.E. Lin and G.L. Vivianib, "Hierarchical Economic Dispatch of Piecewise Quadratic Cost Functions", *IEEE Transactions on PAS*, vol. PAS-103, no 6, June, 1984.
[23.] N. Ramaraj and R Rajaram, "Analytical approach to optimize generation schedule of plant with multiple fuel options", *Journal of The Institution of Engineers* (India), vol. 68, p 106, 1987.
[24.] R.B. Kelkar and R.C. Desai, "easy approach to generation scheduling with multiple fuel options", *Journal of The Institution of Engineers* (India), vol. 80, p 96, 1999.
[25.] R.M.S. Danraj and F. Gajendran, "An efficient algorithm to find optimal economic load dispatch for plants having discontinuous fuel cost functions", *Journal of The Institution of Engineers* (India), vol. 85, pp. 64-67, 2004.

Sarat Chandra Swain received his M.E. degree from UCE Burla in 2001. Presently he is working as an Assistant Professor in the Department of Electrical Engineering, School of technology, KIIT,University, Bhubaneswar, Orissa, India. He is working towards his PhD in KIIT University in the area of Application of Computational Intelligent Techniques to Power System.

Dr. Sidhartha Panda received Ph.D. degree from Indian Institute of Technology (IIT), Roorkee, India in 2008, M.E. degree from VSS University of Technology, (erstwhile UCE, Burla) in 2001 and. B.E. degree from Bangalore University in 1991 all in Electrical Engineering. Presently he is working as a Professor in the EEE department at National Institute of Science and Technology (NIST), Berhampur, Orissa. Earlier he worked as an Associate Professor in KIIT Deemed University and also in various other engineering colleges for about 15 years. He has published about 60 papers in various International Journals. Presently, he is acting as a reviewer of some International Journals namely; IEEE Transactions on Industrial Electronics, Applied Soft Computing (Elsevier), International Journal Electric Power and Energy Systems (Elsevier), International Journal Simulation Modelling Practice and Theory (Elsevier), International Journal of Control and Intelligent Systems (ACTA Press). The biography of Dr Panda has been included "Who's Who in the World": 2010 edition, and "Who's Who in Science and Engineering": 2011-2012, by in Marquis', USA, "2000 Outstanding Intellectuals of The 21st Century", and "2000 Outstanding Scientists: 2010", and nominated for "Top 100 Engineers: 2010", by International Biographical Centre, Cambridge, England.. His areas of research include MATLAB/SIMULINK, Flexible AC Transmission Systems (FACTS), Modeling of Machines, Power Systems and FACTS, Controller Design, Power System Stability, Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Multi-objective Optimization, Economic Operation of Power System, Fuzzy Logic, Model Order Reduction, Distributed Generation and Wind Energy.

Dr. A. K. Mohanty is an Ex-Professor in N.I.T.Rourkela.Now he is working as a Professor Emeritus in Electrical Engg. Deptt. at KIIT UNIVERSITY Bhubaneswar.

Cemal Ardil is with National Academy of Aviation, AZ1045, Baku, Azerbaijan, Bina, 25th km, NAA