A SOLUTION TO AN ECONOMIC DISPATCH PROBLEM BY A FUZZY ADAPTIVE GENETIC ALGORITHM

H. NEZAMABADI-POUR, S. YAZDANI, M. M. FARSANGI AND M. NEYESTANI

Abstract. In practice, obtaining the global optimum for the economic dispatch **(ED)** problem with ramp rate limits and prohibited operating zones is presents difficulties. This paper presents a new and efficient method for solving the economic dispatch problem with non-smooth cost functions using a Fuzzy Adaptive Genetic Algorithm (FAGA). The proposed algorithm deals with the issue of controlling the exploration and exploitation capabilities of a heuristic search algorithm in which the real version of Genetic Algorithm (RGA) is equipped with a Fuzzy Logic Controller (FLC) which can efficiently explore and exploit optimum solutions. To validate the results obtained by the proposed FAGA, it is compared with a Real Genetic Algorithm (RGA). Moreover, the results obtained by FAGA and RGA are also compared with those obtained by other approaches reported in the literature. It was observed that the FAGA outperforms the other methods in solving the power system economic load dispatch problem in terms of quality, as well as convergence and success rates.

1. Introduction

An economic dispatch (ED) problem has complex, nonlinear characteristics as well as equality and inequality constraints. The objective of an economic dispatch problem of a power system is to determine the optimal combination of power outputs for all generators, which minimizes the total fuel cost while satisfying constraints. In the traditional ED problem, the cost function for each generator is approximately represented by a single quadratic function and the problem is solved using mathematical programming based on optimization techniques such as the lambda-iteration, gradient and dynamic programming methods. However, many mathematical assumptions such as convexity, quadratic, differentiable or linear objectives are required to simplify the problem.

The practical ED problem with ramp rate limits, prohibited operating zones, valve-point effects and multi-fuel options is represented as a non-smooth or nonconvex optimization problem with equality and inequality constraints and this makes the problem of finding the global optimum difficult to solve by traditional methods.

Received: August 2009; Revised: March 2010 and April 2010; Accepted: August 2010

Key words and phrases: Economic dispatch, Genetic algorithm, Fuzzy adaptive genetic algorithm, Non-smooth cost functions.

A considerable amount of research has been conducted for solving a practical ED problem with non-convex cost functions using various heuristic approaches such as Genetic Algorithm (GA) [3, 4, 19, 27], Simulated Annealing (SA) [26], Artificial Neural Network (ANN) [12], Tabu Search (TS) [13], Evolutionary Programming (EP) [23, 28] , Particle Swarm Optimization (PSO) [2, 6, 14, 20, 22], Ant Colony Optimization (ACO) [24] and Differential Evolutionary (DE) [25].

Among these, only [2] and [22] considered the exploration and exploitation capabilities of the algorithm. However, the issue of controlling exploration and exploitation capabilities of a heuristic search algorithm is a key factor in finding optimum solutions. Exploration is the ability of expanding the search space, whereas exploitation is the ability of finding the optima around a good solution. When exploration increases, the algorithm tends to search for new points in the search space. Therefore, for a high performance search, it is essential to have a tradeoff between exploration and exploitation. In view of this, the authors in [18] proposed an approach to the non-smooth ED problem using the Memetic Algorithm (MA) with three different local searches. MA is a hybrid GA that uses a genetic search to explore the search space and a local search to exploit information in the search region. To validate the results obtained by the proposed MAs, the problem is solved by by an RGA as well as an MA adapted from CHC in the literature.[7].

In this paper, in continuation of our previous work [18], in order to make a suitable tradeoff between exploration and exploitation, an alternative approach is proposed to the non-smooth ED problem using a Fuzzy Adaptive Genetic Algorithm (FAGA) in which the genetic algorithm is equipped with fuzzy logic that can efficiently search and exploit the optimum solutions. One type of non-smooth ED problems, i.e. ED with ramp rate limits and prohibited operating zones, is considered.

To validate the results obtained by the proposed FAGA, the problem is also solved by RGA and the results are compared with results previously obtained using MA and CHC in [18]. The results obtained by FAGA are also compared with those obtained by previous approaches reported in the literature. The basic concepts of the GA and MA are briefly explained in the next section followed by a description of FAGA. The basic formulation of ED problem is given in section 3. Section 4 introduces cases that are used in experiments. Implementation and results are presented in section 5 and section 6 provides a conclusion.

2. Overview of Genetic, Memetic and Fuzzy Adaptive Genetic Algorithms

2.1. Genetic Algorithm.

GA has desirable characteristics as an optimization tool and offers significant advantages over traditional methods. GA efficiently searches the large solution space containing discrete or discontinuous variables and nonlinear constraints. Thus, it is able to give a good solution of a certain problem in a reasonable computation time. The optimal solution is sought from a population of solutions using random process. Applying to the current population, the following three operators create a new generation: selection, crossover and mutation.

The reproduction is a process dependent on an objective function to maximize or minimize, which depends on the problem.

GA can be implemented through Binary GA (BGA) and Real GA (RGA). The first step in the solution of an optimization problem using BGA is the encoding of the variables. The most usual approach is to represent these variables as strings of 0s and 1s. A collection of such strings is called population. Then, selection, crossover and mutation are applied on the encoded variables. On the other hand, RGA uses the real codes and the selection, crossover and mutation are applied to the variable directly.

2.2. **Memetic Algorithm.** MA is a population-based meta-heuristic search method inspired by the conjecture of natural selection and Dawkins' notion of 'meme' [11, 16] . GAs that have been hybridized with local search techniques are often called Memetic Algorithms [17]. Therefore, MA is a hybrid GA that uses a genetic search to explore new solutions and a local search to exploit information in the search region. In other words, MA tries to achieve a balance or a suitable tradeoff between exploration and exploitation by genetic search and a local search. The term 'memetic algorithm' was used for the first time by Moscato [17] and then in Norman and Moscato [15] , where MA was a hybrid of traditional GA and simulated annealing.

The unit of information in the memetic approach is referred to as a meme rather than a gene. The main difference between MA and GA is that memes can be improved upon by their owner. This improvement is obtained by incorporating local search into the genetic algorithm. Different versions of MA are reported in the literature [1, 7, 15] and may be classified into two different groups. The first group uses efficient hill- climbers on continuous domain. Hill-Climbing (HC) is a local search algorithm that starts from a single solution point. Local search is applied to each member of population. If improvement is achieved, the population members are replaced and used to generate the next population by selection and recombination.

The second group of MAs are known as MAs with crossover-based local search (XLS) algorithms. The crossover in GA is an operator that produces offspring around the parents. Therefore, it may be considered to be a move operator for a local search strategy. In XLS, a crossover creates offspring distributed densely around the parents, favoring local tuning. Crossover Hill-Climbing (XHC) [7] is a kind of XLS approach that allows the self-adaptive capacity of real-parameter crossover operators to be exploited inside the proper XLS, i.e., it is a self- adaptive crossover local search method. The mission of XHC is to obtain the best possible accuracy levels for leading the population toward the most promising search areas, producing their refinement [7]. In addition, the MA employs an adaptive mechanism that determines the probability with which every solution receives the application of XHC. Thus it attempts to adjust the exploration/exploitation balance. The crossover introduced in [7] is an extended version of the BLX-α, called parent-centric BLX-α (PLX-α). Suppose that $X_1 = (x_1^1, x_1^2, \ldots, x_1^n)$ and $X_2 = (x_2^1, x_2^2, \ldots, x_2^n)$, $x^d \in [x^{d,min}, x^{d,max}] \subset \Re$, $d = 1, 2, \ldots, n$ are two real-coded chromosomes that have been selected as parents to produce two offspring using the crossover operator. Here x_i^d indicates the variable d (dth dimension) of chromosome i and $x^{d,min}$ and $x^{d,max}$ are the lower and upper bounds of the variable d , respectively. According to [8] , PLX-α randomly generates one of two possible offspring: $O_1 = (o_1^1, o_1^2, \ldots, o_1^n)$ and $O_2 = (o_2^1, o_2^2, \ldots, o_2^n)$, where o_1^d is a randomly chosen number from a uniform distribution on the interval $[\max\{x^{d,min}, x_1^d - |x_1^d - x_2^d|.\alpha\}, \min\{x^{d,max}, x_1^d - |x_1^d - x_2^d|.\alpha\}]$ and o_2^d is randomly chosen from the interval $[\max\{x^{d,min}, x_2^d - |x_1^d - x_2^d|.\alpha\}, \min\{x^{d,max}, x_2^d - |x_1^d - x_2^d|.\alpha\}]$.

In our previous work in [18], by proposing three different local searches, we arrived at three different MAs which we called FVMA, SVMA and TVMA. It was shown that SVMA performs better than FVMA and TVMA. In SVMA, the HC local search is applied only on the first K individuals with the best fitness function. In this way, the algorithm searches the neighborhood of the K selected individuals to find a solution with greater improvement in the value of the fitness function. If such a solution exists, the new individual replaces the old. This local search may be repeated m times for the K selected individuals.

To evaluate the proposed FAGA, the results are compared with those obtained by the SVMA and XHC. The flow charts for SVMA and XHC are shown in Figure 1 and 2, respectively.

2.3. **Fuzzy Adaptive Genetic Algorithm.** Achieving a good balance between exploration and exploitation is a difficult issue when applying a heuristic algorithm to solve different problems. When exploration increases, the algorithm tends to search for new points in the search space. While this capability is vital for a heuristic search, the algorithm also needs to exploit the neighborhood of good solutions for optima. Furthermore, the exploration and exploitation cannot occur simultaneously. In other words, once a heuristic algorithm has reached the stage of high exploration, the probability of it converging to a near-optimum solution is small. On the other hand, a excessive exploitation may lead the algorithm to stagnation and premature convergence to a local optimum.

One of the major steps in applying GA or any other heuristic algorithm to a particular problem is to choose its components e.g. different operators (in GA, selection, mutation, crossover, and replacement mechanism) and their parameters (in GA , mutation and crossover rates, selection pressure, and the population size). The value of these parameters significantly affects exploration and exploitation abilities and determine whether the algorithm will find a near-optimum solution efficiently [5].

Many researchers have tried to address this issue in. For example, MA makes an effort to carry out the steps of exploration and exploitation independently of each other. As mentioned before, MA uses GA as a global search to do the exploration and applies a local search such as hill-climbing or simulated annealing to exploit around good solutions to find the optimum.

Eiben et al. (1999) considers two major ways for setting parameter values: parameter tuning, and parameter control [5]. In parameter tuning, one tries to find a suitable value for the parameters before running the algorithm. The algorithm

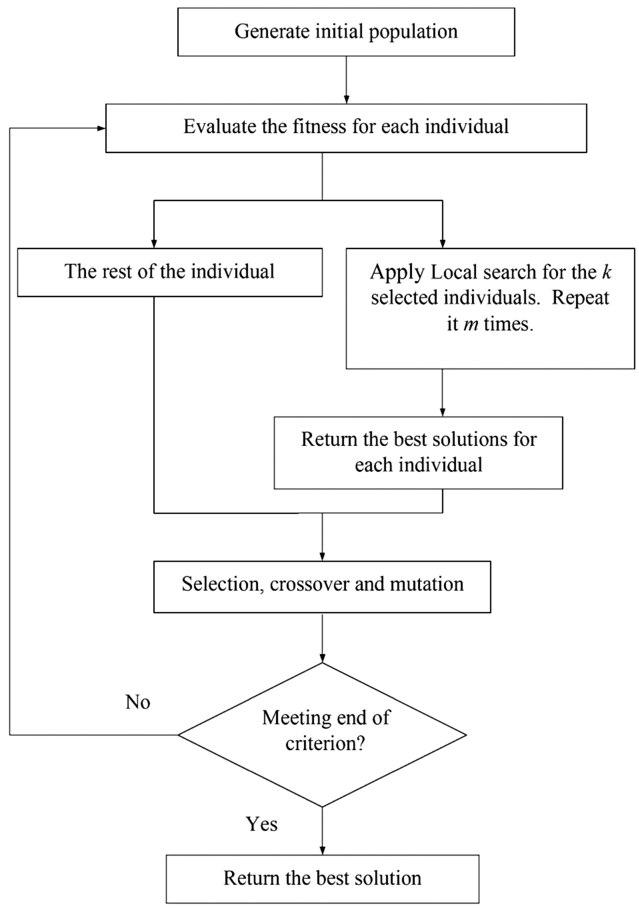

FIGURE 1. The Principle of SVMA

is then run using these values which remain fixed during the run. This approach suffers from the following technical drawbacks [5]:

a) Parameters are dependent on each other and trying all combinations is not practical.
b) It is time consuming, even if the parameters are optimized regardless of their relationships.
c) During a run, an algorithm needs different values of exploration and exploitation. Even if the parameters were tuned well, it may not lead to the desired exploration and exploitation.

These drawbacks motivated researchers to use parameter control techniques for parameter setting. Parameter control techniques may be classified into three categories: deterministic, self-adaptive, and adaptive. Deterministic parameter control occurs when the value of a parameter is changed by a set of deterministic rules.

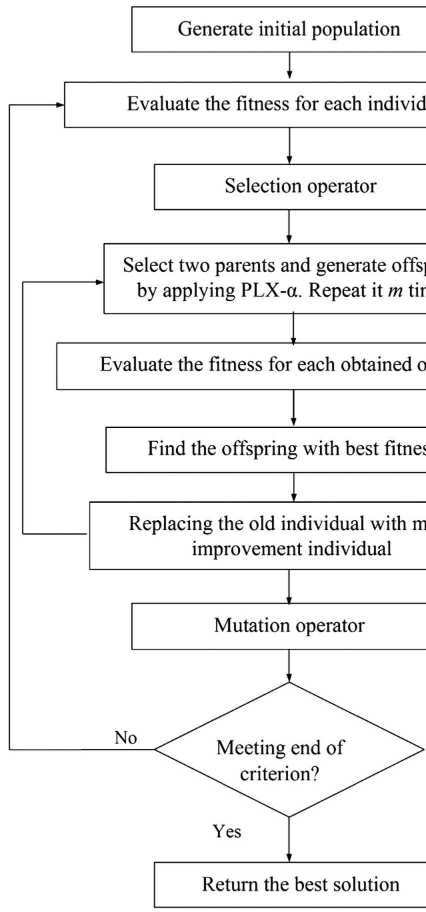

FIGURE 2. The Principle of XHC

The rules determine the parameter value, usually using a time-varying function but without any feedback from search and performance indices.

When one designing a GA based on fixed parameters or controlling parameters using deterministic rules, to avoid getting trapped in a local optimum in the first few iterations, a heuristic search algorithm should explore the search space to find new solutions. After sufficient iterations, exploration fades out, exploitation begins and the algorithm tunes itself in semi-optimal points [21].

The self-adaptive parameter control terminology uses the idea of "evolution of evolution". In this policy, the parameters to be adapted are encoded as added genes into the chromosomes to adjust during the reproduction.

The adaptive terminology of control uses feedback from the search (e.g. search direction, performance indices, convergence) to change the parameters . The feedback signals are selected from genotype or phenotype and monitor the search conditions before changing the parameters so as to lead the search to a desired state. In this approach, based on existing conditions and using a set of heuristic rules, one can control the parameters dynamically to achieve desired abilities of exploration and exploitation. Generally, adaptive terminology of control uses the current value of the parameters together with performance feedback signals to produce new value of the parameters.

Another important approach for adaptive control of parameters is the adaptive parameter control technique using Fuzzy Logic Controllers (FLC). The GA that uses a FLC to dynamically adjust its parameters is known as the Fuzzy Adaptive Genetic Algorithm (FAGA). The FLC provide facilities for handling the complex problem of adaptation of GA parameters by extracting expert knowledge in terms of fuzzy rules. The FLC uses a combination of performance indices and current GA control parameters as inputs and computes new control parameter values to be used by GA. Figure 3 depicts the structure of the FAGA.

Many researchers have tried to address the issue of balancing between exploration and exploitation using FAGA [8, 9, 10, 29]. Herrera and Lozano (2003) have Carried out some basic research in this field and shown how one can design an FAGA [10]. We note that FLC can also be used to adaptively control parameters of other heuristic algorithms.

From another point of view, parameter control techniques are categorized into three groups based on levels of adaptation [5]:

Population level: in which the parameters are set to be the same for all individuals in the population.

Individual level: where the parameters are set to be different for each individual of the population but similar for all components of a particular individual.

Component level: where the parameters are different for different components (gene/allele) in an individual and also differ from individual to individual.

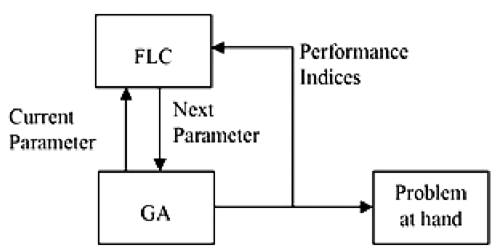

Figure 3. FAGA Structure

The authors believe that adaptive parameter control (especially fuzzy adaptive control) is the best way for balancing exploration and exploitation in a particular heuristic algorithm. Therefore, a FAGA is proposed which is applied at the individual level. Since, to understand the mechanism of this FAGA, it is necessary to have some knowledge of FLC, in the rest of this section we first describe fuzzy inference systems and then proceed to discuss our FAGA. Fuzzy systems provide the means of representing expert knowledge about the process in terms of fuzzy (IF-THEN) rules. A fuzzy rule is the basic unit for capturing knowledge in fuzzy systems. Just like a conventional rule in artificial intelligence, a fuzzy rule has two components: an if part and a then part, also referred to as antecedent and consequent, respectively. The main structure of the fuzzy rule is given by equation (1).

$$IF < antecedent > THEN < consequent > \tag{1}$$

The antecedent of a fuzzy rule has a condition that can be satisfied to a degree. As in conventional rules, the antecedent of a fuzzy rule may combine multiple simple conditions into a complex one using AND, OR and NOT logic operators. The consequent of a fuzzy rule can be classified into two main categories:

a) Fuzzy consequent (equation (2)), where C is a fuzzy set.

b) Functional consequent (equation (3)), where p,q and r are constant.

Basically, fuzzy logic controllers incorporate an expert's experience into the system design and are composed of four blocks (Figure 4). A FLC comprises a fuzzifier that transforms the 'crisp' inputs into fuzzy inputs using membership functions that represent fuzzy sets of input vectors, a knowledge-base that includes the information given by the expert in the form of linguistic fuzzy rules, an inference-system (Engine) and a defuzzifier that transforms the fuzzy results of the inference into a crisp output using a defuzzification method [10].

The knowledge-base has two components: a data-base, which defines the membership functions of the fuzzy sets used in the fuzzy rules, and a rule-base comprising a collection of linguistic rules that are joined by a specific operator. The generic structure of a FLC is shown in Figure 4. Based on the consequent type of fuzzy rules, there are two common types of FIS which vary according to differences between the specifications of the consequent part (equation (2) and (3)). The first fuzzy system uses the inference method proposed by Mamdani in which the rule consequence is defined by fuzzy sets and has the following structure:

$$IF\ x\ is\ A\ and\ y\ is\ B\ THEN\ f\ is\ C \tag{2}$$

The second fuzzy system proposed by Takagi, Sugeno and Kang (TSK) contains an inference engine in which the conclusion of a fuzzy rule comprises a weighted linear combination of the crisp inputs rather than a fuzzy set. The TSK system has the following structure:

$$IF\ x\ is\ A\ and\ y\ is\ B\ THEN\ f = px + qy + r \tag{3}$$

where p, q and r are constant parameters. The TSK models are suitable for approximating a large class of non-linear systems.

The knowledge-base containing the database and rule-base of a FLC can be constructed from an expert's knowledge, where the expert selects the membership functions and rules. Fuzzy systems can also be constructed from data, which alleviates the problem of knowledge acquisition. In practice, we may use soft computing tools, hybridizing artificial neural networks (ANN), evolutionary algorithms (EA) and clustering techniques with fuzzy logic. In the proposed FAGA, the FLC is constructed by knowledge of an expert.

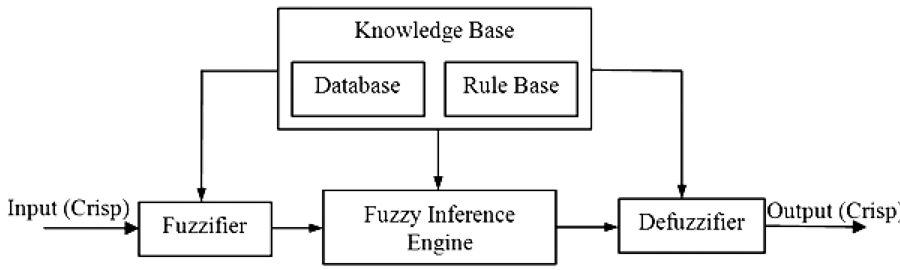

FIGURE 4. FLC Structure

2.4. **The Proposed FAGA.** The proposed FAGA is an algorithm that integrates a FLC to adapt the standard deviation of Gaussian mutation, σ , during a run. The Gaussian mutation in RGA is as follows:

$$x_i^d(new) = x_i^d(old) + N(0, \sigma_i) \tag{4}$$

where $x_i^d(old)$ and $x_i^d(new)$ represent dimension d (variable d) of individual i before and after mutation, respectively and $N(0, \sigma_i)$ is a random number generated by Gaussian density function with zero mean and standard deviation σ_i . In this paper, the standard deviation is selected as a control parameter because of its ability to make the population diverse and hence be an important factor in barring premature convergence. When σ is increased, the search algorithm based on a mutation operator produces offspring far from the parent. Thus the population diversity as well as exploration ability of the algorithm increases. On the other hand, when σ is decreased, offspring are produced close to their parents and the algorithm can exploit around the existing solution. Also, an algorithm with a small value of σ may potentially converge to a near-optimal solution. It is better to rewrite σ_i in more detail as $\sigma_i^d = \sigma_i^0(x^{d,max} - x^{d,min})$, where $x^{d,max}$ and $x^{d,min}$ represent the upper and lower bounds of variable x^d (dimension d), and σ_i^0 is a scale factor that is similar for all dimensions of individual i, specifies the range σ_i^d.

As mentioned, the proposed FAGA is applied at the individual level. In other words, the FLC is run for each individual in the mutation stage and $\sigma_i^0(t+1)$ (the control parameter) is computed. Two types of inputs are considered: the current value of parameter control ($\sigma_i^0(t)$) and performance indices of GA such as diversity measures, average fitness and variance of fitness. In this paper, the Expected Value of each individual (EV_i) and fitness Variance of population (Var) are used along with $\sigma_i^0(t)$ as inputs and control parameter $\sigma_i^0(t+1)$ is the output. EV_i and Var are computed respectively as follows:

$$EV_i = N.\frac{fit_i}{\sum_{j=1}^{N} fit_j} = \frac{fit_i}{fit^{ave}} \tag{5}$$

$$Var = \frac{fit^{max} - fit^{ave}}{fit^{max} - fit^{min}} \tag{6}$$

where $fit^{ave} is the average fitness of the current population, fit_i$,fit^{max} ,fit^{min} , are respectively the fitness values of individual i , the best individual and the worst individual, and N is the population size. The input EV_i shows the quality of individual i with respect to other individuals of population. A large value of EV_i indicates that the corresponding individual is of good quality and may lead us to a near optimum solution (in an ideal case to the optimum). The input Var is a phenotypic measure that represents the amount of diversity in the population. For a converged population, Var tends to zero while a large value for Var indicates a diverse population.

Each input and output is defined by an associated set of linguistic variables. A linguistic variable is quantified by a fuzzy set and qualified by a linguistic term. A knowledge base (containing data base and rule base) depends on the problem and the knowledge base for ED problem is explained in section 5.

3. Formulation of Economic Dispatch Problem

3.1. **Traditional ED Problem with Smooth Cost Functions.** In the traditional ED problem, the cost function for each generator has been approximately represented by a single quadratic function. The primary objective of the ED problem is to determine the optimal combination of power outputs of all generating units so that the required load demand at minimum operating cost is met while satisfying system equality and inequality constraints. Therefore, the ED problem can be described as a minimization process with the following objective:

$$\min F = \sum_{i=1}^{N_G} F_i(P_{Gi}) = \sum_{i=1}^{N_G} (a_i {P_{Gi}}^2 + b_i P_{Gi} + C_i) \tag{7}$$

subject to

$$\sum_{i=1}^{N_G} P_{Gi} = P_{load} + P_{loss} \tag{8}$$

$$P_{Gi\ min} \le P_{Gi} \le P_{Gi\ max} \qquad for\ i = 1, 2, \ldots, N_G \tag{9}$$

where F is the total generation cost ($/hr), F_i is the fuel-cost function of generator ($/hr), N_G is the number of generators, P_{Gi} is the real power output of generator i (MW), and a_i, b_i and c_i are the fuel-cost coefficients of generator i , P_{load} is the total load in the system (MW), P_{loss} is the network loss (MW) that can be calculated by B matrix loss formula,$P_{Gi\ min}$ and $P_{Gi\ max}$ are the minimum and maximum power generation limits of generator i .

3.2. **Practical ED Problem with Non-smooth Cost Functions.** As mentioned, for a complete formulation of the ED, it is necessary to take ramp rate limits, prohibited operating zones, valve- point effects, and multifuel options into consideration. The resulting ED will be a nonconvex optimization problem that has multiple minima, which makes the problem of finding the global optimum difficult. We shall here consider a special type of non- smooth ED problems, i.e. an ED with ramp rate limits and prohibited operating zones as follows:

a) Generator Ramp Rate Limits. By considering generator ramp rate limits, the effective real power operating limits are modified as follows:

$$\max(P_{Gi\ min}, P_{Gi}^0 - DR_i) \le P_{Gi} \le \min(P_{Gi\ max}, P_{Gi}^0 + UR_i) \qquad for\ i = 1, 2, \ldots, N_G \tag{10}$$

where P_{Gi}^0 is the previous operating point of generator i , DR_i and UR_i are the down and up ramp limits of the generator i .

b) Prohibited Operating Zones. A generator with prohibited regions (zones) has discontinuous fuel-cost characteristics. The discontinuous fuel-cost characteristics of the generators by considering prohibited zones are shown in Figure 5.

Taking into account these prohibited operating zones, we have the following constraint :

$$P_{Gi} \in \begin{cases} P_{Gi\ min} \le P_{Gi} \le P_{Gi}^{LB_1} & \\ P_{Gi}^{UB_{k-1}} \le P_{Gi} \le P_{Gi}^{LB_k} & k = 2, 3, \ldots, N_{PZi} \\ P_{Gi}^{UB_k} \le P_{Gi} \le P_{Gi\ max} & i = 1, 2, \ldots, N_{GPZ} \end{cases} \tag{11}$$

where $P_{Gi}^{LB_K}$ and $P_{Gi}^{UB_K}$ are the lower and upper boundaries of prohibited operating zone k of generator i in (MW), respectively, N_{PZi} is the number of prohibited operating zones of generator i and N_{GPZ} is the number of generators with prohibited operating zones.

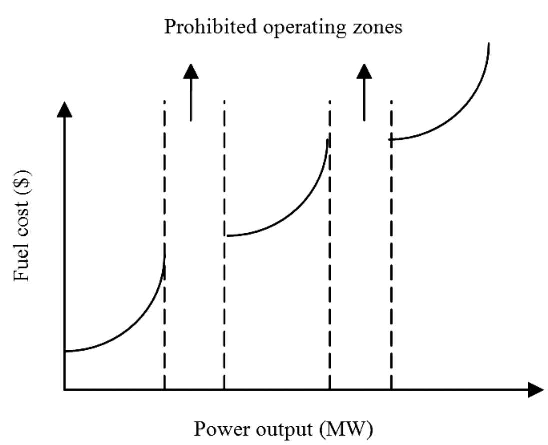

FIGURE 5. Input-output Curve with Prohibited Operating Zones

4. Case Studies

The superiority of FAGA has been demonstrated on two test systems for solving an ED problem with non-convex solution spaces. The study systems are as follows:

1) The First Study System. This study system consists of six generators with ramp rate limit and prohibited operating zones. The input data for a 6-generator system are given in [6] and the total demand is set as 1263 MW. All the generators have ramp rate limits. The network losses are calculated by B matrix loss formula. That the best generation cost reported until now is 15446.02 \$/h [2].

2) The Second Study System. This study system consists of 15 generators with ramp rate limits and prohibited operating zones. The input data of this system are given in [6] and has a total load of 2630 MW. Again, network losses are calculated by the B matrix loss formula. The main difference of the study systems 1 and 2 is that system 2 has more local minima. Thus, the ability of the proposed algorithms is investigated on a larger system. The best generation cost reported until now is 32751.39 \$/h [2].

For convenience, the data for the two systems are given in the Appendix.

5. Implementation of FAGA and RGA

The implementation of FAGA and RGA for the ED problem for the systems under study is presented below. It should be noted that in this paper the penalty method is used to handle constraints in the ED problem. Also, the objective function is converted into an appropriate fitness function, so that the algorithms may be able to solve the problem.

5.1. Implementation of FAGA and RGA for the First Study System.

1) The Use of RGA. For study system1 (with six generators), the goal of the optimization is to find the best generation for six generators. Therefore, a configuration is considered for each individual in RGA as a vector $[P_{G1}, P_{G2}, P_{G3}, P_{G4}, P_{G5}, P_{G6}]$.

The size of population was taken to be 20. In RGA, the chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated with some measure of fitness, which is calculated from the objective function (equation (7) subject to 8 - 11).

Moving to a new generation is carried out according to the results obtained for the old generation. A roulette wheel is created from the obtained values of the objective function of the current population. To create the next generation, new chromosomes, called offspring, are formed using a crossover operator and a mutation operator. In RGA, linear crossover and Gaussian mutation are used with the crossover probability $P_c = 0.7$ and the mutation probability $P_m = 0.1$.

For the study of system1, the number of iterations was 50, the stopping criterion.

2) The Use of FAGA. RGA is applied to the ED problem and FLC is used to improve its performance. FAGA was implemented using Stochastic Universal

Sampling (SUS) selection, linear crossover and Gaussian mutation with crossover probability $P_c = 0.7$ and mutation probability $P_m = 0.1$. Before applying the selection operator, linear fitness scaling was performed to control selection pressure. The parameter of linear scaling was set up such that the ratio of the selection probability of the best individual of current population over the individual with average fitness was 3.

As mentioned in section 2-4, each input and output is defined by an associated set of linguistic variables. A linguistic variable is quantified by a fuzzy set and qualified by a linguistic term. Figure 6 presents the membership functions of the inputs and output. Based on our experience the standard deviation parameter, σ_i^0 , should be in the interval $[0, 0.3]$. This interval guarantees that GA is able to carry out the necessary exploration and exploitation. The set of linguistic terms used for σ_i^0 as well as Var and EV_i was {Low, Medium, High}. As a result, EV_i and Var have values in [0, 3] and [0, 0.6], respectively. A 'Mamdani' type FLC was implemented using 'min' intersection, 'max' union, 'min' implication method (clipping method), 'max' aggregation method and 'centroid' defuzzification. The pseudo-code of the proposed FAGA is shown in Figure 7.

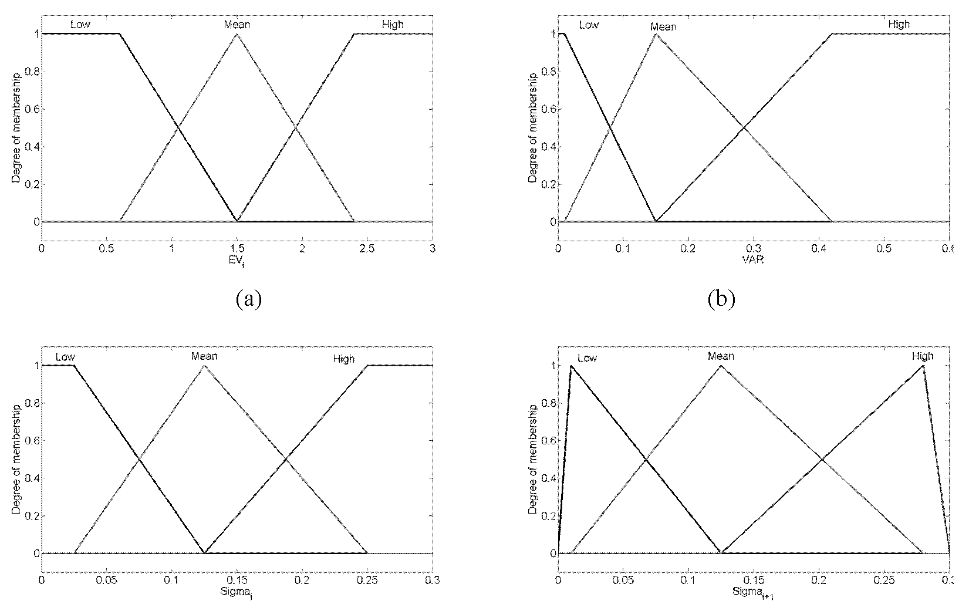

FIGURE 6. Membership Functions of Inputs and Output
a) EV_i b) Var c) $\sigma_i^0(t)$ d) $\sigma_i^0(t+1)$

Fuzzy rules describe the relation between inputs and output of the FLC. Fuzzy rules (Rule-Base) used in the proposed FLC are shown in Table 1. The Rule-Base was produced considering three following heuristics:

- The fuzzy rules 19-27 are justified by the heuristic: "decrease σ_i^0 to exploit around good solutions for finding optimum solution".
- The fuzzy rules 1-6 and 10-12 are justified by the heuristic: "increase σ_i^0 when diversity is not sufficient".
- The fuzzy rules 1-9 and 13-18 are justified by the heuristic: "otherwise do not change σ_i^0. It may provide a better solution or may allow convergence".

```
Begin
    t=0;
    initialize population P(t);                   % randomly generate the initial
                                                  population
    evaluate population P(t);                     % compute fitness values
    t = t + 1;
    while (not termination condition) do
        Scaling the fitness;                      % linear fitness scaling
        Select (parent-selection) P(t) from P(t − 1);  % SUS operator
        CalculateVar and EV_i for population      % EV_i is calculated for each in-
                                                  dividual
        Crossover P(t) to yield C(t);             % linear crossover operator
        Calculate σ_i^0 for population            % σ_i^0 is calculated for each indi-
                                                  vidual
        mutate C(t)                               % Gaussian mutation operator
        evaluate C(t) ;
    end
end
```

Figure 7. Pseudo-code of the Proposed FAGA

In order to show the effectiveness of the proposed FAGA, the results obtained by FAGA were compared with results obtained by MAs and RGA and other algorithms available in the literature. To make the results comparable, the size of population and the number iterations were the same as those found in the literature.

To find the minimum cost, the algorithms were run for 50 independent runs with different random seeds. The results obtained by the FAGA, RGA and MAs are shown in Table 2. The last four columns of the table show the results obtained by a binary version of GA, PSO, a modified (new) version of PSO having local random search (NPSO-LRS) reported in [22] and a self-organizing hierarchical PSO (SOH_PSO) reported in [2]. This table shows that the FAGA performs better than other algorithms in terms of the best generation schedule with minimum network loss in addition to minimum generation cost.

The optimal solutions obtained by all heuristic algorithms (the generation level of the units) are within the limits. The entries in Table 3 show the prohibited zones, generation limits and the results obtained by the algorithms. As is illustrated, these results are within the allowed regions.

The average best-so-far of each run were recorded and averaged over 50 independent runs. For clarity, the convergence characteristics for minimum cost are

Rule#	EV_i	Var	$\sigma_i^0(t)$	$\sigma_i^0(t+1)$
1	Low	Low	Low	High
2	Low	Low	Medium	High
3	Low	Low	High	High
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	High
6	Low	Medium	High	High
7	Low	High	Low	Low
8	Low	High	Medium	Medium
9	Low	High	High	High
10	Medium	Low	Low	Medium
11	Medium	Low	Medium	High
12	Medium	Low	High	High
13	Medium	Medium	Low	Low
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	High
16	Medium	High	Low	Low
17	Medium	High	Medium	Medium
18	Medium	High	High	High
19	High	Low	Low	Low
20	High	Low	Medium	Low
21	High	Low	High	Medium
22	High	Medium	Low	Low
23	High	Medium	Medium	Low
24	High	Medium	High	Low
25	High	High	Low	Low
26	High	High	Medium	Low
27	High	High	High	Low

Table 1. Proposed Rule-base for the Proposed FLC

presented in Figure 8. This figure shows that the FAGA performs better than other algorithms in terms of convergence rate.

5.2. **Implementation of FAGA and RGA for the Second Study System.** To investigate the ability of the FAGA to obtain the solution and the convergence characteristics of the algorithms, the same study was carried out on the second system which is larger . The size of population is considered to be 100. In RGA, linear crossover and Gaussian mutation were used with crossover probability $P_c = 0.7$ and the mutation probability $P_m = 0.1$. The number of iterations was 200, which is the stopping criterion.

The results obtained are given in Table 4. The last three columns of the table show the obtained results by binary version of GA and PSO reported in [6] and SOH_PSO reported in [2] . The results obtained by all algorithms (listed in Table 4) reveals that the optimal solution found by FAGA is better than those obtained by

unit	RGA	FAGA	SVMA [18]	XHC [18]	GA [22]	PSO[22]	NPSO-LRS[22]	SOH_PSO[2]
P1	437.40479	445.6843	441.10864	432.18417	474.8066	447.4970	446.9600	438.21
P2	167.83279	172.1456	173.2927	171.41996	178.6363	173.3221	173.3944	172.58
P3	261.12049	265	260.01194	259.26337	262.2089	263.4745	262.3436	257.42
P4	139.78646	135.8666	141.37487	146.26538	134.2826	139.0594	139.512	141.09
P5	174.76442	169.5886	167.07667	166.24179	151.9039	165.4761	164.7089	179.37
P6	94.55358	87.2219	92.47441	99.88581	74.1812	87.128	89.0162	86.88
Total Gen.	1275.4625	1275.507	1275.3392	1275.2605	1276.03	1276.01	1275.94	1275.55
Loss	12.47183	12.494	12.34867	12.2594	13.0217	12.9584	12.9361	12.55
Total load	1262.9907	1263.013	1262.9905	1263	1263	1263.05	1263	-
Cost	15444.77	**15442.89**	15443.02	15446.37	15459	15450	15450	15446.02

Table 2. Comparison of the Obtained Results of the Algorithms (6- generator System)

	generation limit		prohibited zones		obtained optimal values			
Unit	Pmax	Pmin	Zone 1 (MW)	Zone 2 (MW)	RGA	FAGA	SVMA [18]	XHC [18]
1	500	100	[210-240]	[350-380]	437.40479	445.6843	441.10864	432.18417
2	200	50	[90-110]	[140-160]	167.83279	172.1456	173.2927	171.41996
3	300	80	[150-170]	[210-240]	261.12049	265	260.01194	259.26337
4	150	50	[80-90]	[110-120]	139.78646	135.8666	141.374874	146.26538
5	200	50	[90-110]	[140-150]	174.76442	169.5886	167.07667	166.24179
6	120	50	[75-85]	[100-105]	94.55358	87.2219	92.47441	99.88581

Table 3. The Obtained Optimal Values are Within the Generation Limits and Prohibited Operating

other algorithms. The convergence characteristics in finding the minimum cost are given in Figure 9. This figure shows that the FAGA outperforms other algorithms.

Table 5 gives the mean values and the standard deviations of the results obtained on the 50 independent runs for the study systems 1 and 2. This table illustrates that FAGA not only provides better solutions but it is also more robust. This is because the variance of the solutions obtained by the FAGA is smaller than the variance for the other algorithms.

6. **Conclusions**

This paper presents an alternative approach to the non-smooth ED problem using FAGA. The proposed algorithm deals with the issue of controlling of exploration and exploitation capabilities of GA using a fuzzy logic controller which can efficiently explore and exploit the optimum solutions. One type of non-smooth ED problem, i.e. ED with ramp rate limits and prohibited operating zones, is considered.

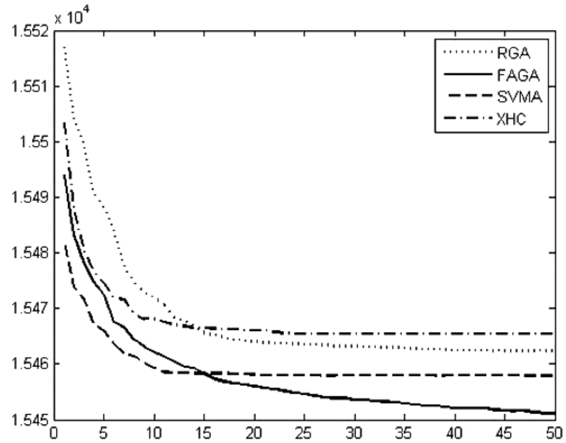

Figure 8. Convergence Characteristics of FAGA, SVMA, RGA and XHC on the Average Best-so-far in Finding the Solution in Study System 1

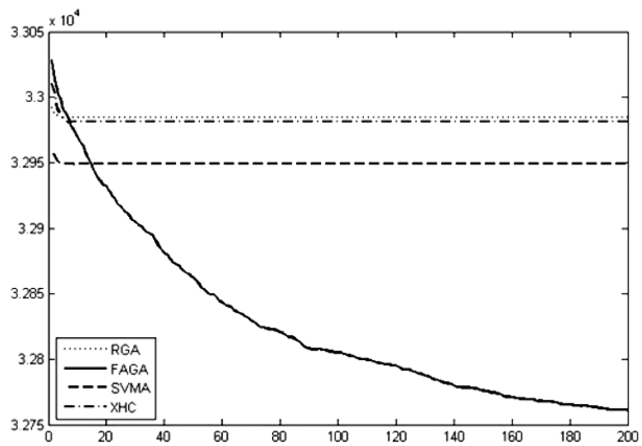

Figure 9. Convergence Characteristics of FAGA, SVMA, RGA and XHC on the Average Best-so-far in Finding the Solution in Study System 2

Comparisons with the results of earlier methods in the literature, show that the proposed FAGA is better in terms of solution quality, dynamic convergence, robustness and stability.

unit	RGA	FAGA	SVMA[18]	XHC [18]	GA [22]	PSO [22]	SOH _PSO[2]
P1	453.4254	455	451.7666	447.58422	415.3108	439.1162	455
P2	366.5059	380	348.4377	377.86384	359.7206	407.9727	380
P3	119.4675	129.9098	128.4636	126.64927	104.425	119.6324	130
P4	118.9688	130	128.9972	121.47775	74.9853	129.9925	130
P5	166.2064	170	161.5901	159.62502	380.2844	151.0681	170
P6	458.4367	457.5862	457.4644	458.08802	426.7902	459.9978	459.96
P7	425.6331	430	425.7521	423.2246	341.3164	425.5601	430
P8	116.279	60.666	137.0207	88.1951	124.7867	98.5699	117.53
P9	89.7486	76.0249	75.0004	61.61113	133.1445	113.4936	77.9
P10	114.9537	149.7171	130.8868	158.96636	89.2567	101.1142	119.54
P11	77.693	80	70.5982	73.9875	60.0572	33.9116	54.5
P12	71.345	80	77.6038	67.18293	49.9998	79.9583	80
P13	28.1113	25	30.2831	36.93463	38.7713	25.0042	25
P14	28.7535	20.9559	20.7348	37.73344	41.9425	41.414	17.86
P15	26.2959	15.6749	19.5886	22.84158	22.6445	35.614	15
Total generation	2661.8238	2660.5348	2664.1881	2661.9654	2668.4	2262.4	2662.29
Loss	31.9192	30.489	33.5005	31.9322	38.2782	32.4306	32.28
Total load	2629.9046	2630.0458	2630.6876	2630.0332	2630.1218	2230.03	-
cost	32839.26	**32714.56**	32830.19	32835.39	33113	32858	32751.39

Table 4. Comparison of the Obtained Results of the Algorithms (15- generator System)

	study system 1		study system 2	
algorithm	ST	mean	ST	mean
RGA	12.9	15462.33	48.67	32984.03
FAGA	**4.38**	**15451.1**	**23.03**	**32761.16**
SVMA[18]	10.52	15457.88	45.63	32948.79
XHC[18]	11.92	15465.39	46.13	32981

Table 5. The Mean Value and Standard Deviation (ST) of Fitness Among the Independent Runs

7. Appendix

Data for the study system 1:

unit									Prohibited zones	
	a	b	c	P_{min}	P_{max}	UR	DR	P^0	Zone 1 (MW)	Zone 2 (MW)
1	240	7	0.007	100	500	80	120	440	[210-240]	[350-380]
2	200	10	0.0095	50	200	50	90	170	[90-110]	[140-160]
3	220	8.5	0.009	80	300	65	100	200	[150-170]	[210-240]
4	200	11	0.009	50	150	50	90	150	[80-90]	[110-120]
5	220	10.5	0.008	50	200	50	90	190	[90-110]	[140-150]
6	190	12	0.0075	50	120	50	90	110	[75-85]	[100-105]

$$B_{ij} = \begin{bmatrix} -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.015 \\ -0.0005 & -0.0006 & -0.001 & -0.0006 & 0.0129 & -0.0002 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.001 & -0.0006 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \end{bmatrix}$$

$$B_{0i} = [\ -0.0003908 \quad -0.0001297 \quad 0.0007047 \quad 0.0000591 \quad 0.0002161 \quad -0.0006635\ ]$$

$$B_{00} = 0.0056$$

Data for the study system 2:

unit									Prohibited zones		
	a	b	c	P_{min}	P_{max}	UR	DR	P^0	Zone 1 (MW)	Zone 2 (MW)	Zone 3 (MW)
1	671	10.1	0.0003	150	455	80	120	400	-	-	-
2	574	10.2	0.0002	150	455	80	120	300	[420,450]	[305,335]	[185,225]
3	374	8.8	0.0011	20	130	130	130	105	-	-	-
4	374	8.8	0.0011	20	130	130	130	100	-	-	-
5	461	10.4	0.0002	150	470	80	130	90	[390,420]	[260,335]	[180,200]
6	630	10.1	0.0003	135	460	80	120	400	[430,455]	[365,395]	[230,255]
7	548	9.8	0.0004	135	465	80	120	350	-	-	-
8	227	11.2	0.0003	60	300	65	100	95	-	-	-
9	173	11.2	0.0008	25	162	60	100	105	-	-	-
10	175	10.7	0.0012	20	160	60	100	110	-	-	-
11	186	10.2	0.0036	20	80	80	80	60	-	-	-
12	230	9.9	0.0055	20	80	80	80	40	[30,55]	[65,75]	-
13	225	13.1	0.0004	25	85	80	80	30	-	-	-
14	309	12.1	0.0019	15	55	555	55	20	-	-	-
15	323	12.4	0.0044	15	55	55	55	20	-	-	-

$$B_{ij} = 0.0001 \times \begin{bmatrix} 14 & 12 & 7 & -1 & -3 & -1 & -1 & -1 & -3 & 5 & -3 & -2 & 4 & 3 & -1 \\ 12 & 15 & 13 & 0 & -5 & -2 & 0 & 1 & -2 & -4 & -4 & 0 & 4 & 10 & -2 \\ 7 & 13 & 76 & -1 & -13 & -9 & -1 & 0 & -8 & -12 & -17 & 0 & -26 & 111 & -28 \\ -1 & 0 & -1 & 34 & -7 & -4 & 11 & 50 & 29 & 32 & -11 & 0 & 1 & 1 & -26 \\ -3 & -5 & -13 & -7 & 90 & 14 & -3 & -12 & -10 & -13 & 7 & -2 & -2 & -24 & -3 \\ -1 & -2 & -9 & -4 & 14 & 16 & 0 & -6 & -5 & -8 & 11 & -1 & -2 & -17 & 3 \\ -1 & 0 & -1 & 11 & -3 & 0 & 15 & 17 & 15 & 9 & -5 & 7 & 0 & -2 & -8 \\ -1 & 1 & 0 & 50 & -12 & -6 & 17 & 168 & 82 & 79 & -23 & -36 & 1 & 5 & -78 \\ -3 & -2 & -8 & 29 & -10 & -5 & 15 & 82 & 129 & 116 & -21 & -25 & 7 & -12 & -72 \\ -5 & -4 & -12 & 32 & -13 & -8 & 9 & 79 & 116 & 200 & -27 & -34 & 9 & -11 & -88 \\ -3 & -4 & -17 & -11 & 7 & 11 & -5 & -23 & -21 & -27 & 140 & 1 & 4 & -38 & 168 \\ -2 & 0 & 0 & 0 & -2 & -1 & 7 & -36 & -25 & -34 & 1 & 54 & -1 & -4 & 28 \\ 4 & 4 & -26 & 1 & -2 & -2 & 0 & 1 & 7 & 9 & 4 & -1 & 103 & -101 & 28 \\ 3 & 10 & 111 & 1 & -24 & -17 & -2 & 5 & -12 & -11 & -38 & -4 & -101 & 578 & -94 \\ -1 & -2 & -28 & -26 & -3 & 3 & -8 & -78 & -72 & -88 & 168 & 28 & 28 & -94 & 1283 \end{bmatrix}$$

$$B_{0i} = \begin{matrix} [-0.0001 & -0.0002 & 0.0028 & -0.0001 & 0.0001 & -0.0003 & -0.0002 & \ldots \\ -0.0002 & 0.0006 & 0.0039 & -0.0017 & -0.0000 & -0.0032 & 0.0067 & -0.0064] \end{matrix}$$

$$B_{00} = 0.0055$$

Acknowledgements. This research is supported in part by the Fuzzy Systems and Applications Center of Excellence, Shahid Bahonar University of Kerman, Kerman, Iran.

References

[1] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore and M. Sumner, *A fast adaptive memetic algorithm for off-line and on-line control design of PMSM drives*, IEEE Trans. on Systems, Man and Cybernetics-Part B, Special Issue on Memetic Algorithms, **37(1)** (2007), 28-41.

[2] K. T. Chaturvedi, M. Pandit and L. Srivastava, *Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch*, IEEE Trans. Power Syst., **23(3)** (2008), 1079-1087.

[3] P. H. Chen and H. C. Chang, *Large-scale economic dispatch by genetic algorithm*, IEEE Trans. Power Syst., **10(4)** (1995), 1919-1926.

[4] C. L. Chiang, *Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels*, IEEE Trans. Power Syst., **20(4)** (2005), 1690-1699.

[5] A. E. Eiben, R. Hinterding and Z. Michlewicz, *Parameter control in evolutionary algorithms*, IEEE Transaction on Evolutionary Computation, **3(2)** (1999), 124-141.

[6] Z. L. Gaing, *Particle swarm optimization to solving the economic dispatch considering the generator constraints*, IEEE Trans. Power Syst., **18(3)** (2003), 1187-1195.

[7] F. Herrera, N. Krasnogor and D. Molina, *Real-coded memetic algorithms with crossover hill-climbing*, Evolutionary Computation, **12(3)** (2004), 273-302.

[8] F. Herrera and M. Lozano, *Adaptive control of mutation probability by fuzzy logic controllers*, Parallel Processing Solving from Nature VI, Berlin: Springer, (2000), 335-344.

[9] F. Herrera and M. Lozano, *Adaptive fuzzy operators based on coevolution with fuzzy behaviours*, IEEE Transaction on Evolutionary Computation, **5(2)** (2001), 1-18.

[10] F. Herrera and M. Lozano, *Fuzzy adaptive genetic algorithm: design, taxonomy, and future directions*, Soft Computing, **7** (2003), 545-562.

[11] N. Krasnogor and J. Smith, *A tutorial for competent memetic algorithms: model, taxonomy and design issues*, IEEE Trans. Evol. Comput., **9(5)** (2005), 474-488.

[12] K. Y. Lee, A. Sode-Yome and J. H. Park, *Adaptive Hopfield neural network for economic load dispatch*, IEEE Trans. Power Syst., **13(2)** (1998), 519-526.

[13] W. M. Lin, F. S. Cheng and M. T. Tsay, *An improved tabu search for economic dispatch with multiple minima*, IEEE Trans. Power Syst., **17(1)** (2002), 108-112.

[14] E. Mehdizadeh, S. Sadi-nezhad and R. Tavakoli-moghaddam, *Optimaization of fuzzy clastering criteria by a hybrid PSO and fuzzy c–means clustering algorithm*, Iranian Journal of Fuzzy Systems, **5(3)** (2008), 1-14

[15] Memetic Algorithms Home Page, http://www.densis.fee.unicamp.br/∼moscato /memetic_ home. html.

[16] P. Merz and B. Freisleben, *Fitness landscape analysis and memetic algorithms for the quadratic assignment problem*, IEEE Trans. Evol. Comput., **4(4)** (2000), 337-352.

[17] P. Moscato, *On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms*, Memetic Algorithms Home Page, http://www. densis. fee. unicamp. br /∼moscato /memetic_ home. html.

[18] M. Neyestani, M. M. Farsangi and H. Nezamabadi-pour, *Memetic algorithm for economic dispatch with nonsmooth cost functions*, Accepted by Iranian Journal of Electrical and Computer Engineering (IJECE) (in Farsi).

[19] S. O. Orero and M. R. Irving, *Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach*, Proc. Inst. Elect. Eng. Gen. Transm. Distrib., **143(6)** (1996), 529-534.

[20] J. B. Park, K. S. Lee, J. R. Shin and K. Y. Lee, *A particle swarm optimization for economic dispatch with nonsmooth cost functions*, IEEE Trans. Power Syst., **20(1)** (2005), 34 -42.

[21] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, *GSA: a gravitational search algorithm*, Information Sciences, **179(13)** (2009), 2232-2248.

[22] A. I. Selvakumar and K. Thanushkodi, *A new particle swarm optimization solution to non-convex economic dispatch problems*, IEEE Trans. Power Syst., **22(1)** (2007), 42-51.

[23] N. Sinha, R. Chakrabarti and P. K. Chattopadhyay, *Evolutionary programming techniques for economic load dispatch*, IEEE Trans. Evol. Comput., **7(1)** (2003), 83-94.

[24] T. Sum-im, *Economic dispatch by ant colony search algorithm*, Proc. IEEE Conference on Cybernetics and Intelligent Systems, (2004), 416-421.

[25] S. K. Wang, J. P. Chiou and C. W. Liu, *Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm*, IET Generation Transmission & Distribution, **1(5)** (2007), 793-803.

[26] K. P. Wong and C. C. Fung, *Simulated annealing based economic dispatch algorithm*, Proc. Inst. Elect. Eng. C, **140(6)** (1993), 509-515.

[27] K. P. Wong and Y. W. Wong, *Genetic and genetic simulated-annealing approaches to economic dispatch*, Proc. Inst. Elect. Eng. Gen. Transm. Distrib., **141(5)** (1994), 507-513.

[28] H. T. Yang, P. C. Yang and C. L. Huang, *Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions*, IEEE Trans. Power Syst., **11(1)** (1996), 112-118.

[29] X. Zeng and B. Rabenasolo, *A fuzzy logic based design for adaptive genetic algorithms*, European Congress on Intelligent Techniques and Soft Computing, (1997), 660-664.

H. Nezamabadi-pour*, Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran
E-mail address: nezam@mail.uk.ac.ir

S. Yazdani, Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran
E-mail address: sajjad.yazdani@gmail.com

M. M. Farsangi, Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran
E-mail address: mmaghfoori@mail.uk.ac.ir

M. Neyestani, Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran
E-mail address: mehdi2594@yahoo.com

*Corresponding author