Enhanced Bee Swarm Optimization Algorithm for Dynamic Economic Dispatch

Taher Niknam and Faranak Golestaneh

Abstract—**This paper proposes an enhanced bee swarm optimization method to solve the dynamic economic dispatch problem of thermal units considering the valve-point effects, ramp-rate limits, and the transmission power losses. The bee swarm optimization algorithm unlike most of the population-based algorithms employs different moving patterns to search the feasible solution space. This property makes an effective balance between exploration and exploitation. Different modifications in moving patterns of the bee swarm optimization method are proposed to search the feasible space more effectively. The efficiency of the method is validated using three test systems with 10, 30, and 60 units, including 240, 720, and 1440 design variables. The latter can be considered as a large-scale power system. The results are compared with other reported works in this area and found to be superior.**

Index Terms—**Dynamic economic dispatch (DED), enhanced bee swarm optimization (EBSO), ramp-rate limits, valve-point effects.**

Nomenclature

Indices	
i	Thermal generating units (TGU) index.
t	Time interval (hour) index.
k	Iteration index of enhanced bee swarm optimization (EBSO).
v	Experienced forager bee index.
h	Onlooker bee index.
q	Scout bee index.
j	Bees index.
Constants	
N	Number of TGUs.
NT	Number of time intervals.
a_i, b_i, c_i, d_i, e_i, h_i	Cost coefficients of the ith TGU.
$B_{i,j}$	Loss coefficients between ith and jth generators (MW^{-1}).
$p_{i\max}$, $p_{i\min}$	Maximum and minimum power output of the ith TGU, respectively (MW).
UR_i	Ramp-up rate of the ith TGU (MW/h).
DR_i	Ramp-down rate of the ith TGU (MW/h).
$p_{i\max}^t$, $p_{i\min}^t$	Maximum and minimum power output of the ith TGU at time t, respectively (MW).
$n(\zeta)$	Number of the experienced forager bees.
$Iter_{\max}$	Maximum iteration number.
$Iter$	Current iteration.
Variables	
p_i^t	Generation output of the ith TGU at time t (MW).
P_D^t	Load demand at time t (MW).
P_{Loss}^t	The system total real power losses at time t (MW).
$F(\mathbf{P})$	Total fuel cost of generation of all TGUs through dispatch periods ($).
$f_i(p_i^t)$	Total fuel cost of generation of ith TGU at time t ($).
$P(\beta, j)^t$	Position of the jth bee in the set of β at time t.
$P(\zeta, v)^t$	Position of the vth bee in the set of ζ at time t.
$P(\chi, h)^t$	Position of the hth bee in the set of χ at time t.
$P^{(}\vartheta, q)^t$	Position of the qth bee in set ϑ at time t.
w_b	Cognitive weight factor.
w_g	Social weight factor.
w_{bmin}, w_{bmax}	Minimum and maximum values of the cognitive weight factors, respectively.
w_{gmin}, w_{gmax}	Minimum and maximum values of the social weight factors, respectively.
r	Random number with uniform distribution between 0 and 1.
Sets	
ϑ	Scout bees.
χ	Onlooker bees.
ζ	Forager bees.
β	Total number of the bees.

I. Introduction

Dynamic ECONOMIC dispatch (DED) is one of the major optimization issues in power system operations. Its objective is to allocate the forecasted load demand over a certain period of time among available generators in the best economical manner, while all physical and operational constraints are satisfied. Considering different constraints for the purpose of more precise modeling, the DED shows nonconvex characteristics [1], [2].

Different methods are proposed in the literature for coping with the DED problem. Traditional methods [3] and [4] fail

Manuscript received January 5, 2012; revised February 19, 2012; accepted February 20, 2012.

The authors are with the Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz 71557-1387, Iran (e-mail: niknam@sutech.ac.ir; f.golestaneh@sutech.ac.ir).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2012.2191831

to lead to optimal solutions because of nonlinear and nonconvex characteristics of the DED problem. In addition, they are computationally complex and may trap in local optima. Over the past few years, research has been using heuristic optimization methods in the DED problem [5]. Although these methods impose no restriction on the problem formulation, they are incapable to guarantee achieving global optimal solution. The main problem of these methods is the "curse-of-dimensionality," which leads to high computational cost.

More recent methods are hybrid methods. In this regard, evolutionary programming (EP) and particle swarm optimization (PSO) were combined with sequential quadratic programming (SQP) in [6]. Combining seeker optimization algorithm and SQP was introduced in [7]. However, regulation of the control parameters of these hybrid methods is a challenging and complicated task, so modified optimization algorithms such as modified differential evolution [2], adaptive hybrid differential evolution (AHDE) [8], improved chaotic PSO (ICPSO) [9], and quantum genetic algorithm [10] were developed.

The bee swarm optimization (BSO) algorithm is a population-based optimization technique, which is inspired by foraging behavior of the honey bees. To the best of our knowledge, a few algorithms have been developed based on this idea for numerical optimization. Artificial bee colony and virtual bee algorithm are two examples [11], [12]. These types of algorithms have been proved to have better performance compared to the other population-based algorithms, such as the ant colony optimization algorithm, the PSO, and the genetic algorithm (GA), for solving numerical optimization problems [11]–[13]. In most of the optimization algorithms, all individuals in the population use a homologous pattern to search the space and update their positions. Methods that use only one moving pattern may ignore regions, which possibly contain candidate optima. In order to solve this problem, it is essential to employ algorithms, which provide different moving patterns such as the BSO algorithm. Different behaviors of the bees in the BSO set up effective balancing mechanism between exploration and exploitation.

In this paper, an EBSO algorithm is proposed to solve the DED problem considering the ramp-rate limits, the valve-point effects, and the transmission power losses. The DED as a complex, nonconvex, high dimensional and extremely constrained problem is considered as a benchmark for testing effectively and applicability of the proposed EBSO algorithm.

Despite the mentioned advantages of the BSO, the original BSO suffers from premature convergence in a high dimensional complex problem like the DED. Therefore, in this paper, several valuable enhancements to the BSO are developed to improve search capability of the BSO and enhance calculation speed. These modifications were made to design a more powerful optimization technique in comparison with the other population-based techniques, such as the GA, PSO, and the differential evolution (DE).

The EBSO algorithm uses three types of bees to find the optimal solution of the DED problem. Each type of the bees employs a different moving pattern. Accordingly, the feasible region will be searched more effectively. The EBSO algorithm uses a set of approaches, including two novel moving patterns, a reformation technique, repulsion factor, and nonlinear adaptive weights. In addition, constraint-handling schemes are offered to manage equality constraints effectively without enforcing any restrictions.

The proposed EBSO is tested on two popular test systems implemented in much research in the area, including 10-unit test system and 30-unit large-scale power system. The 10-unit test system is studied under two cases by considering and neglecting transmission power losses. In addition, to validate the applicability of the proposed method for high dimensional optimization problems, a 60-unit power system including 1440 design variables is considered. The results are compared with the most recently published works, which solved the DED problem. The results confirm the superiority and effectiveness of the EBSO algorithm over the previous ones in solving the DED problem.

The remainder of this paper is organized as follows. Section II deals with the mathematical formulation of the DED problem. The proposed EBSO algorithm for the DED problem is described in Section III. The implementation of the EBSO algorithm to solve the DED problem is presented in Section IV. The feasibility and efficiency of the proposed method are assessed on three test systems in Section V. This paper concludes in Section VI.

II. Mathematical Description

A. Objective Function

$$\text{Minimize } F(\mathbf{P}) = \sum_{t=1}^{T}\sum_{i=1}^{N} f_i(p_i^t) \tag{1}$$

where $\mathbf{P} = \begin{bmatrix} P^1 & P^2 & \ldots & P^{NT} \end{bmatrix}$ and $P^t = \begin{bmatrix} p_1^t & p_2^t & \ldots & p_N^t \end{bmatrix}^{\mathrm{T}}$.

The above fuel cost function is comprised of two terms: the smooth quadratic function and the absolute value of sinusoidal function of valve-point effects as follows:

$$\begin{aligned} f_i(p_i^t) = {} & a_i + b_i p_i^t + c_i (p_i^t)^2 \\ & + \left| e_i \times \sin(h_i \times (p_{i\min} - (p_i^t))) \right|. \end{aligned} \tag{2}$$

B. Constraint

The DED optimization problem is subject to the following constraints:

$$\sum_{i=1}^{N} p_i^t = P_D^t + P_{\text{Loss}}^t \qquad t = 1, 2, \ldots, NT \tag{3}$$

$$P_{\text{Loss}}^t = \sum_{i=1}^{N}\sum_{j=1}^{N} p_i^t B_{ij}^t p_j^t \qquad t = 1, 2, \ldots, NT \tag{4}$$

$$p_{i\min} \le p_i^t \le p_{i\max} \quad i = 1, 2, \ldots, N \quad t = 1, 2, \ldots, NT \tag{5}$$

$$p_i^t - p_i^{t-1} \le UR_i \quad i = 1, 2, \ldots, N \quad t = 1, 2, \ldots, NT \tag{6}$$

$$p_i^{t-1} - p_i^t \le DR_i \quad i = 1, 2, \ldots, N \quad t = 1, 2, \ldots, NT. \tag{7}$$

Constraint (3) represents the power balance equation in each period. Using the approximation based on the B-coefficients, transmission network losses are expressed in (4). Also, generation limits are set in (5). Finally, (6) and (7) impose up–down ramp rate limits, respectively.

III. Enhanced Bee Swarm Optimization Algorithm

In this section, the proposed EBSO algorithm is presented.

A. Overview of the Standard BSO

The EBSO algorithm includes three types of bees called experienced forager, onlooker, and scout. These bees fly in $N \times NT$-dimensional space to find optimal solution. In the EBSO, each set of the bees has a distinct flying pattern. Few numbers of the bees, which have the worst fitness values, are considered as the scout bees. Fitness values in the EBSO represent the quality of the source foods found by the bees. The remaining bees partitioned equally into onlookers and experienced foragers. The first half of these bees associated with better fitness are chosen as the experienced foragers and the rest are chosen as the onlookers. At the end of each iteration, the bees are classified to the experienced forager, onlooker, and scout based on their fitness values. The percentage of the scout, onlooker, and forager bees are defined manually and do not change dynamically. The position of each bee indicates a solution represented by

$$P(\beta, j)^t = \left[p_1(\beta, j)^t, p_2(\beta, j)^t, \ldots, p_N(\beta, j)^t\right]^{\mathrm{T}}. \tag{8}$$

B. Experienced Forager Bee

In the previous works, which are based on the behavior of the bees, a hard restriction exists on the flying pattern of bees (e.g., the forager bees fly only toward the elite bee) [11]–[13]. A different moving pattern is used in [14] which does not work satisfactorily in problems with several local optima. These methods may cause the premature convergence because of using the information achieved by the swarm imperfectly. Therefore, the EBSO algorithm employs two different moving patterns for the experienced forager bees. In this method, the experienced forager bees are sorted based on their fitness values and partitioned into two equal parts. The new food sources are achieved as follows.

1) The better part of the experienced forager bees fly considering social and cognitive information achieved by the swarm. Using this method helps to balance between exploration and exploitation. Thus, the global optimum can be found with the higher probability. Throughout the iterated procedure, each experienced forager bee remembers its own best position associated with the best personal fitness value, which is defined by

$$P(\zeta, \upsilon)^t_{\text{best}} = \left[p_1(\zeta, \upsilon)^t_{\text{best}}, p_2(\zeta, \upsilon)^t_{\text{best}}, \ldots, p_N(\zeta, \upsilon)^t_{\text{best}}\right]^{\mathrm{T}}. \tag{9}$$

The food source with the best fitness value, which is discovered by the swarm so far, is denoted by $\mathbf{G}_{\text{best}}$ and the associated bee is called the elite bee. The new food source of the υth experienced forager bee is calculated as follows:

$$\begin{aligned} P(\zeta, \upsilon)^t_{\text{new}} = {} & P(\zeta, \upsilon)^t_{\text{old}} + w_b \times r_b \times (P(\zeta, \upsilon)^t_{\text{best}} - P(\zeta, \upsilon)^t_{\text{old}}) \\ & + w_g \times r_g \times (G^{k,t}_{\text{best}} - P(\zeta, \upsilon)^t_{\text{old}}) \end{aligned} \tag{10}$$

where r_g and r_b are two random numbers between 0 and 1; also, w_g and w_b determine the importance of the social or cognitive information for each iteration, respectively.

2) In the aforesaid method for updating the experienced forager bees, each bee uses only its own information and the elite bee information, so good information obtained by the other bees may be neglected. To overcome this imperfection, in this paper, updating the position of the other half of the experienced forager bees is done by selecting two experienced forager bees randomly as $m_1 \neq m_2 \neq \upsilon$. The EBSO algorithm employs this pattern movement to increase the diversity of the solutions to some extent and help to escape from local optima. The new source food for the υth experienced forager bees is obtained as follows:

$$P(\zeta, \upsilon)^t_{\text{new}} = \begin{cases} \text{if} \quad F(\mathbf{P}(\zeta, m_1)) \leq F(\mathbf{P}(\zeta, m_2)) \\ P(\zeta, \upsilon)^t_{\text{old}} + r \times (P(\zeta, m_1)^t - P(\zeta, m_2)^t) \\ \text{else} \\ P(\zeta, \upsilon)^t_{\text{old}} + r \times (P(\zeta, m_2)^t - P(\zeta, m_1)^t) \end{cases} \tag{11}$$

where $F(\mathbf{P}(\zeta, m))$ is the fitness value of the source food, which is discovered by the mth experienced forager bee.

C. Onlooker Bees

The onlooker bees use the information obtained by the experienced forager bees to modify their flying trajectories. At each iteration of the algorithm, the experienced forager bees advertise the position and the nectar of the food sources, which are discovered by them, in the dance floor. An onlooker bee evaluates these food sources and their nectar values. Subsequently, the onlooker bee uses a probabilistic approach to select one of the source foods advertised in the dance area and follow the bee which found it. This selected experienced forager bee is called the elite bee and represented by $P(\chi, h)^t_e = \left[p_1(\chi, h)^t_e, p_2(\chi, h)^t_e, \ldots, p_N(\chi, h)^t_e\right]^{\mathrm{T}}$ with the probability of $prob_\upsilon$. The better fitness value of the source food causes the larger probability and encourages more onlooker bees to follow its explorer. The probability of the υth experienced forager bee in the set ζ can be defined as follows:

$$prob_\upsilon = \frac{1/(1 + F(\mathbf{P}(\zeta, \upsilon)))}{\sum_{C=1}^{n(\zeta)} (1/(1 + F(\mathbf{P}(\zeta, C))))} \tag{12}$$

where $F(\mathbf{P}(\zeta, \upsilon))$ is the fitness value of the source food, which is discovered by the υth experienced forager bee. After calculating the probability of the food sources, the onlooker bees employ the roulette wheel mechanism to choose their interesting elite bees. Finally, the onlookers update their position as follows:

$$P(\chi, h)^t_{\text{new}} = P(\chi, h)^t_{\text{old}} + r \times w_g \times (P(\chi, h)^t_e - P(\chi, h)^t_{\text{old}}) \quad (13)$$

where $P(\chi, h)^t_e$ represents the position of the elite bee of the hth onlooker in set χ. $P(\chi, h)^t_{\text{old}}$ and $P(\chi, h)^t_{\text{new}}$ are the current position and the new position of the hth onlooker bee, respectively.

D. Scout Bees

The BSO-RP in [14] uses a stochastic pattern to find new solutions and replaces the food sources with poor qualities. In this method, the scout bee walks randomly in a region with radius τ, finds some new solutions, and chooses the best one as its new source food. Finding just one new solution, the algorithm fails to overcome trapping in local optima. On the other hand, as the number of new source food that a scout bee finds in radius τ increases, the computational time increases. In this paper, instead of the scout bees, a reformation technique is devised to improve the diversity of solutions. Section III-E explains the proposed reformation technique. The EBSO algorithm proposes a moving pattern for scout bees which tries to improve the quality of solutions found by them. This technique uses the best position found by the swarm up to now ($\mathbf{G}_{\text{best}}$) and the mean value of the population in the previous iteration ($\mathbf{M}$) as follows:

$$P(\vartheta, q)^t_{\text{new}} = P(\vartheta, q)^t_{\text{old}} + r \times (G^t_{\text{best}} - l \times M^t) \quad (14)$$

where l is the rate of obedience and determined randomly with equal probability as $l = round[1 + rand(0, 1)]$. $P(\vartheta, q)^t_{\text{old}}$ and $P(\vartheta, q)^t_{\text{new}}$ are the current position and the new position of the qth scout bee, respectively.

E. Techniques to Alleviate Stagnation

Despite the above-mentioned strategies, premature convergence and stagnation may occur. The following techniques are employed in the EBSO algorithm to alleviate stagnation and make the algorithm more powerful in the sense that it finds the global optimum in a shorter time.

1) *Reformation:* To improve the diversity of the solutions the EBSO algorithm uses a reformation technique. In the proposed reformation technique, new positions are produced for half of the bees. These bees are selected randomly. In order to mutate each selected position, $\mathbf{P}(\beta, j)^k$, three bees m_1, m_2, and m_3 are selected randomly as $m_1 \neq m_2 \neq m_3 \neq j$. A mutant position $p(\beta, j)^{k,t}_{\text{mut}} = [p_1(\beta, j)^{k,t}_{\text{mut}}, p_2(\beta, j)^{k,t}_{\text{mut}}, \ldots, p_N(\beta, j)^{k,t}_{\text{mut}}]^{\text{T}}$ is created as follows:

$$P(\beta, j)^{k,t}_{\text{mut}} = P(\beta, m_1)^{k,t} + r \times (P(\beta, m_2)^{k,t} - P(\beta, m_3)^{k,t}). \quad (15)$$

By the following scheme, the trial vector is obtained:

$$p_\theta(\beta, j)^{k,t}_{\text{trial}} = \begin{cases} p_\theta(\beta, j)^{k,t}_{\text{mut}}, & \text{if } (r \leq 0.5) \\ p_\theta(\beta, j)^{k,t}, & \text{else} \end{cases} \quad (16)$$

where $\theta = 1, 2, \ldots, N$.

Among the randomly selected target vector, $\mathbf{P}(\beta, j)^k$, and the trial vector, the one with the better fitness value is selected as the member of the next generation.

2) *Nonlinear Adaptive Weight Factors:* In previous works associated with the BSO algorithm, linear adaptive weight factors were used [14]. However, the nonlinear weight factors are more effective for balancing between local and global search in problems such as the DED, where inputs (i.e., system load) dynamically change over time. In this paper, w_b and w_g are updated using the following nonlinear approach. It is based on the nonlinear sinusoidal function y as follows:

$$y = \sin\left(\frac{\pi}{s} \times \frac{Iter_{\max} - Iter}{Iter_{\max}}\right) \quad (17)$$

where s is a constant value within the range $0 = s = 10$. Different values of s present a wide range of performance of the weight factors from approximately linear characteristics for larger values of s to the periodic ones with several minima and maxima for smaller values of s as follows:

$$w_b = \frac{y - y_{\min}}{y_{\max} - y_{\min}}(w_{b\max} - w_{b\min}) + w_{b\min} \quad (18)$$

$$w_g = \frac{y - y_{\min}}{y_{\max} - y_{\min}}(w_{g\min} - w_{g\max}) + w_{g\max} \quad (19)$$

where $y_{\min}$ and $y_{\max}$ are the minimum and maximum values of the sinusoidal function, respectively. s is considered to be 2 in this paper. Figs. 1 and 2 show the behavior of w_b and w_g, respectively, for different values of the s. From these two figures, nonlinear weight factors (e.g., $s = 2$) more than approximately linear weight factors (e.g., $s = 8$) encourage the algorithm to concentrate on the exploration at the initial iterations and exploitation at the final iterations.

3) *Repulsion Factor:* The bees in the BSO algorithm move toward the experienced forager bees using different approaches. This procedure causes to fly over some parts of the search space, which may include profitable information and has been ignored by the population. Therefore, repulsion techniques are used in this research to improve the diversity of the solutions.

This is done by persuading some of the individuals to move in opposite directions of their elites. The modified moving patterns of the experienced forager, onlooker, and scout bees are defined as follows:

$$P(\zeta, \upsilon)^t_{\text{new}} = P(\zeta, \upsilon)^t_{\text{old}} + sign \times \begin{pmatrix} w_b \times r_b \times (P(\zeta, \upsilon)^t_{\text{best}} - P(\zeta, \upsilon)^t_{\text{old}}) \\ + w_g \times r_g \times (G^t_{\text{best}} - p(\zeta, \upsilon)^t_{\text{old}}) \end{pmatrix} \quad (20)$$

$$P(\chi, i)^t_{\text{new}} = P(\chi, i)^t_{\text{old}} + sign \times \left(w_g \times (P(\chi, i)^t_e - P(\chi, i)^t_{\text{old}})\right) \quad (21)$$

$$P(\vartheta, j)^t_{\text{new}} = P(\vartheta, j)^t_{\text{old}} + sign \times \left(r \times (G^t_{\text{best}} - l \times M^t)\right) \quad (22)$$

where $sign$ is defined as follows:

$$sign = \begin{cases} 1, & \text{if } (r \leq pr) \\ -1, & \text{else} \end{cases} \quad (23)$$

where pr has a constant value, which controls the repulsion rank in the swarm and is considered 0.8 in this paper.

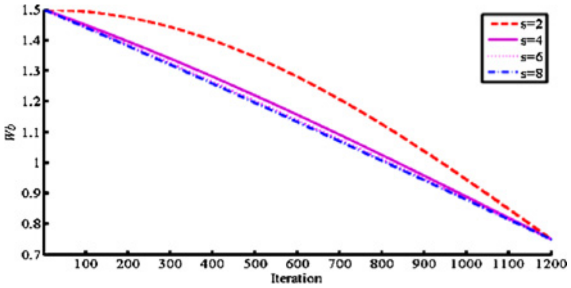

Fig. 1. Variations of w_b versus different iterations.

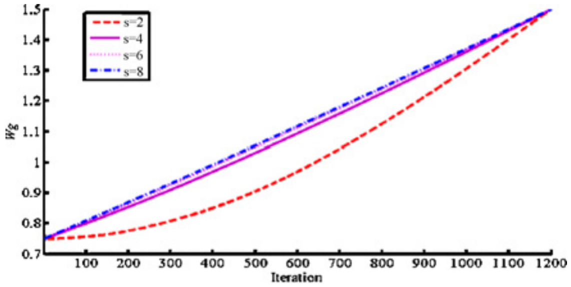

Fig. 2. Variations of w_g versus different iterations.

IV. Solution Methodology

The procedure for implementing the DED optimization algorithm can be summarized in the following steps.

Step 1) Randomly initialize the initial positions of the bees in the feasible range. The position of the individuals for each hour is restricted by the ramp-rate limits as follows:

$$\begin{array}{l} p_{i\,\min}^{t+1} = \max\left\{p_{i\,\min},\ p_i(\beta,j)^t - DR_i\right\} \\ p_{i\,\max}^{t+1} = \min\left\{p_{i\,\max},\ p_i(\beta,j)^t + UR_i\right\}. \end{array} \qquad (24)$$

Step 2) The amount of power balance violation is computed for each of the bees by

$$PD = \sum_{i=1}^{N} p_i^t - P_{\text{Loss}}^t - P_D^t. \qquad (25)$$

a) If $PD = 0$, go to Step 3.

b) If $PD \neq 0$, one of the generated units is selected randomly and PD is subtracted from it. If allocated capacity of the selected unit violates its constraint (24), then the position of the individual is fixed to the boundary values which are determined by (24). Go to Step 2. The algorithm guarantees that different units will be selected to compensate power mismatch. In each hour, if equality constraint cannot be balanced, the algorithm returns to previous hours and rearranges them.

Step 3) The individuals are sorted based on their fitness values, then $\mathbf{P}(\zeta, j)_{\text{best}}$ for each forager bee, the elite bees for each onlooker and $\mathbf{G}_{\text{best}}$ are determined.

Step 4) Modify the position of the particles by (11), (20)–(22) considering (24) and Step 2.

Step 5) Apply the reformation technique and select the final discovered food sources in the current iteration as described in Section III-E considering (24) and Step 2.

Step 6) Update weight factors using the nonlinear approach in Section III-E.

Step 7) Go to Step 3 until the current iteration number reaches the prespecified maximum iteration number.

V. Results and Discussions

In order to verify the effectiveness of the proposed algorithm for practical applications, it is applied to four test systems for solving the DED problem. In all cases, the ramp-rate limits and the valve-point effects are considered. The dispatch horizon *NT* is selected as a day with 24 h. The generator and the power balance constraints for each hour make the DED problem a high dimensional optimization problem with several local optima. Therefore, the methods used for solving the DED problem should be able to escape these local points while satisfying all constraints. All the simulations are carried out by MATLAB 7.01 on a Pentium-IV, 1.8 GHz personal computer with 2 GB RAM. Maximum iteration number is specified as 700 for all tests. The results are presented in the following subsections. The results are compared with the well-known works done in the area. It is worth noting that the other methods are not implemented in this paper and the results mentioned as their best, average and worst results and the actual simulation time are the reported values by the cited references in this paper. In addition, to evaluate the computational efficiency fairly, the equalized CPU time ($k_{\text{eq}} \times$ actual CPU time) is considered as a performance measure. The equalizing factor (k_{eq}) is the ratio of actual CPU speed (GHz) to the speed of CPU (GHz) in which the proposed algorithms are carried out. The assumed case studies are as follows:

Case 1) a 10-unit system by neglecting transmission losses [18] including 24 × 10 design variables;

Case 2) a 10-unit system considering transmission losses [6] including 24 × 10 design variables;

Case 3) a 30-unit power system including 24 × 30 design variables obtained by tripling the number of units in case system 1;

Case 4) a 60-unit power system including 24 × 60 design variables obtained by sextupling the number of units in case system 1.

A. Parameter Settings for Simulation

In order to determine the best control parameters of the EBSO algorithm, different combinations of these parameters are considered. Control parameters of the EBSO algorithm are population size and the percentage of the total bees, which are considered as the experienced forager, the onlooker, and the scout bees. Different combinations of these parameters and their test results are summarized in Table I for Case 1. Tests are carried out 40 times for each combination. In order to fairly discuss the results obtained for different parameter values, Table II provides the comparison of the equalized CPU average execution time, as well as the best, worst and average total fuel cost obtained for Case 1 using the proposed algorithm and the other recent methods reported in the literature.

TABLE I

INFLUENCE OF PARAMETERS ON EBSO PERFORMANCE FOR CASE 1

Number of Population	Percent of onlookers = 49% Percent of foragers = 49% Percent of scouts = 2%					Percent of onlookers = 40% Percent of foragers = 40% Percent of scouts = 20%					Percent of onlookers = 30% Percent of onlookers = 30% Percent of scouts = 30%				
	Maximum cost ($)	Minimum cost ($)	Average cost ($)	Standard deviation	Mean simulation time (min)	Maximum cost ($)	Minimum cost ($)	Average cost ($)	Standard deviation	Mean simulation time (min)	Maximum cost ($)	Minimum cost ($)	Average cost ($)	Standard deviation	Mean simulation time (min)
50 (5 × N)	1 018 841	1 017 880	1 018 188	168	0.200	1 018 970	1 017 448	1 017 786	161	0.201	1 018 704	1 017 501	1 018 005	168	0.200
70 (7 × N)	1 018 742	1 017 747	1 018 160	164	0.205	**1 017 891**	**1 017 147**	**1 017 526**	**147**	**0.205**	1 017 905	1 017 377	1 017 578	162	0.205
120 (12 × N)	1 018 758	1 017 719	1 018 151	161	0.46	1 018 795	1 017 297	1 017 759	153	0.46	1 017 898	1 017 378	1 017 531	157	0.46

TABLE II

COMPARISON OF TOTAL PRODUCTION COST AND SIMULATION TIME FOR CASE 1

Method	Maximum Cost ($)	Minimum Cost ($)	Average Cost ($)	Mean Equalized Simulation Time (min)	Standard Deviation
EP-SQP [6]	–	1 031 746	1 035 748	–	–
MHEP-SQP [6]*	–	1 028 924	1 031 179	–	–
AHDE [8]	–	1 020 082	1 022 474	1.466	
IPSO [15]*	–	1 023 807	1 026 863	0.066	1569.80
AIS [16]	10 24 973	1 021 980	10,23,156	33.795	–
DE [17]	1 027 634	1 023 432	1 026 475	0.399	–
CDE method 3 [18]	1 023 115	1 019 123	1 020 870	0.426	1310.70
ECE [19]	–	1 022 271	1 023 334	0.439	–
ICPSO [10]	–	1 019 072	1 020 027	0.467	493.21
Proposed EBSO	1 017 891	1 017 147	1 017 526	0.205	147.01

*Violate ramp-rate constraints in various degrees.

Tables I and II show that the EBSO with different combinations of the control parameters performs better than earlier methods in solving the DED problem. The best achieved total cost using the EBSO is $1 017 147 for the population size of 70. The percentage of the bee population that was considered as experienced foragers and onlookers was chosen to be 40%.

Table III provides details about the best solutions. According to Table I, increasing the population size beyond 70 does not improve the results significantly. Additionally, the EBSO when the percentage of both the experienced foragers and onlookers = 49% do not perform as effectively as the other combinations. Therefore, the percentage of the worst bees that fly using scout moving pattern must be more than 2% in order to allow the algorithm to find more optimal solutions. The same test is done for the other cases and consequently the following parameter setting is considered for all simulations:

1) number of population: 7 × N;
2) percentage of the experienced forager bees: 40%;
3) percentage of the onlooker bees: 40%;
4) percentage of the scout bees: 20%.

B. Effects of the Different Modifications on the EBSO Algorithm

Several approaches are suggested in this research to improve the convergence characteristics of the original BSO to avoid premature phenomena and alleviate stagnation. Tables IV and V show the obtained results using these approaches for Cases 1 and 4, respectively. These two tables demonstrate that modifications on moving patterns of the bees are unable to alleviate stagnation completely. The results show that the proposed techniques for alleviate stagnation improve the convergence performance of the EBSO significantly. In addition, Fig. 3 shows the convergence characteristics of the proposed EBSO algorithm (S_1), the EBSO algorithm without reformation (S_2), and the original BSO algorithm (S_3). For the EBSO algorithm under both scenarios, the population size and percentage of the scout bees are considered 70 and 20%, respectively. The original BSO is considered with population size equal to 70 and the percentage of the scout equal to 10%. In addition, two new solutions are found around each scout bee. As Fig. 3 demonstrates, although the number of objective function evaluations in each iteration of the EBSO algorithm is more than the other ones, with equal numbers of the objective function evaluation, the EBSO algorithm obtains more optimal solution than S_1 and S_2.

C. Comparative Study

1) *Solution Quality:* Comparison between the results obtained by the EBSO for Cases 1–3 and other methods, which have been reported in the literature, are summarized in Tables II, VI, and VII. The results of the EBSO algorithm are extracted from 40 independent runs. These tables show that the total cost values achieved using the EBSO algorithm for every three cases are the best among other cited methods. Furthermore, a popular optimization software named general algebraic modeling system (GAMS) is used to verify the effectiveness of the proposed EBSO. The same case studies are solved by GAMS uses linear programming to optimize the problems.

Although GAMS is a powerful optimization tool for convex problems, it fails to give optimal solutions for nonconvex ones. This is because of its requirement for piecewise linearization of nonlinear and nonconvex optimization functions, which

TABLE III
BEST SOLUTIONS OBTAINED BY THE PROPOSED METHOD FOR CASE 1 (MW)

Hour	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10	Total load
1	150	135	194.0932	60	122.8666	122.4498	129.5904	47	20	55	1036
2	150	135	268.0932	60	122.8666	122.4498	129.5904	47	20	55	1110
3	226.6257	215	309.3158	60	73	122.4681	129.5904	47	20	55	1258
4	303.2489	295	300.6952	60	73	122.4654	129.5904	47	20	55	1406
5	303.2486	312.1905	307.6308	60.0004	122.8709	122.4684	129.5904	47	20	55	1480
6	379.873	392.1905	298.9693	60	122.8861	122.4907	129.5904	47	20	55	1628
7	379.8731	396.8014	318.5473	60.0001	172.7367	122.4509	129.5904	47	20	55	1702
8	379.8727	396.7994	296.8239	105.8645	222.5995	122.4495	129.5904	47	20	55	1776
9	456.4968	396.7994	321.3026	122.7665	222.5997	122.4446	129.5904	77	20	55	1924
10	456.4968	396.7994	339.4651	172.7665	222.6238	160	129.5904	89.25788	50	55	2072
11	456.4968	396.7994	331.4322	222.7665	222.5997	160	129.5904	119.2579	52.0571	55	2146
12	456.4968	460	323.049	241.2132	222.5997	160	129.5904	120	52.0508	55	2220
13	456.4968	396.7994	318.2496	191.2132	222.5997	160	129.5904	120	22.0508	55	2072
14	456.4968	396.7994	289.8508	141.2132	222.5995	122.4498	129.5904	90	20	55	1924
15	379.8725	396.7994	302.0321	112.2107	172.7328	122.4499	129.5904	85.31211	20	55	1776
16	303.2484	396.7564	286.5656	62.2107	122.8665	122.4498	129.5904	55.31211	20	55	1554
17	226.6244	396.8014	299.6527	60	122.8662	122.4648	129.5904	47	20	55	1480
18	303.2484	396.7999	321.1772	60	172.7335	122.4506	129.5904	47	20	55	1628
19	379.8726	396.7994	297.2166	105.473	222.5977	122.4502	129.5904	47	20	55	1776
20	456.4968	460	340	121.313	222.5997	160	129.5904	77	50	55	2072
21	456.4969	396.7995	315.7452	120.7683	222.5997	160	129.5904	47	20	55	1924
22	379.8726	316.7995	313.7862	70.7683	172.7331	122.4498	129.5904	47	20	55	1628
23	303.2484	236.7995	235.0452	60	122.8665	122.4498	129.5904	47	20	55	1332
24	226.6242	222.2665	178.2025	60	122.8666	122.4498	129.5904	47	20	55	1184

TABLE IV
COMPARISON OF EBSO ON DIFFERENT CASES FOR CASE 1 (40 RUNS)

Methods	Maximum Cost ($)	Minimum Cost ($)	Average Cost ($)
EBSO	1 017 891	1 017 147	1 017 526
EBSO without reformation	1 022 797	1 020 361	1 021 824
EBSO without nonlinear w*	1 018 195	1 017 851	1 018 089
EBSO without repulsion factor	1 018 489	1 017 954	1 018 187
Original BSO	1 037 289	1 033 254	1 035 987

$^{*}w = \frac{(w_{\min} - w_{\max})}{Iter_{\max}} * Iter + w_{\max}$.

TABLE V
COMPARISON OF EBSO ON DIFFERENT CASES FOR CASE 4 (40 RUNS)

Methods	Maximum Cost ($)	Minimum Cost ($)	Average Cost ($)
EBSO	6 117 320	6 113 888	6 115 637
EBSO without reformation	6 146 797	6 130 361	6 139 824
EBSO without nonlinear $w*$	6 118 195	6 113 951	6 116 989
EBSO without repulsion factor	6 119 489	6 113 954	6 118 187
Original BSO	6 159 215	6 147 237	6 154 995

$^{*}w = \frac{(w_{\min} - w_{\max})}{Iter_{\max}} * Iter + w_{\max}$.

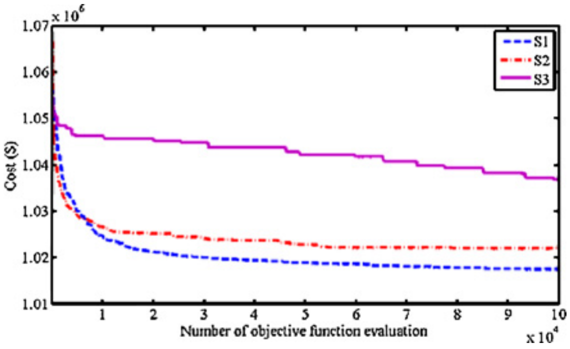

Fig. 3. Convergence characteristics based on number of objective function evaluations for Case 1.

causes trapping in local optima. To clarify the point, Table VIII is offered. Table VIII summarizes the best solutions archived by different solvers of GAMS for Cases 1 and 4. Comparing the best results of the EBSO method given in Tables II and V and the GAMS results, the shortcoming of GAMS in solving the nonconvex DED problem can be concluded.

In Case 3, the number of the units was tripled compared to Case 1 in order to have a 30-unit large-scale power system. Consequently, the degrees of nonconvexity, nonsmoothness, and nonlinearity of the problem are increased significantly. Table VII shows the ability of the suggested method to find the optimal solution for this high dimensional DED problem. Also, the worst solution of the EBSO algorithm for the DED problem of Cases 1–3 is better than the best solution of the other cited methods. Comparing the average cost obtained by the EBSO with respect to other methods emphasizes its better solution quality. In addition, as given in Table V, the best solution obtained for Case 4 is $6 113 888. There are no reported results for this case for further analysis. The convergence characteristics of the EBSO and the original BSO algorithms are shown in Fig. 4 for Case 3.

From Fig. 4, it is clear that the value of the cost function converges smoothly to the optimum solution without any sudden oscillations even for the large-scale system with several

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VI
COMPARISON OF TOTAL PRODUCTION COST AND SIMULATION TIME FOR CASE 2

Methods	Maximum Cost ($)	Minimum Cost ($)	Average Cost ($)	Mean Equalized Simulation Time (min)	Standard Deviation
EP [6]	–	1 054 685	1 057 323	–	–
EP-SQP [6]	–	1 052 668	1 053 771	–	–
MHEP-SQP [6]	–	1 050 054	1 052 349	–	–
IPSO [15]	–	1 046 275	1 048 145	–	–
AIS [16]	1 048 431	1 045 715	1 047 050	41.280	–
Proposed EBSO	1 039 272	1 038 915	1 039 188	0.22	148.02

TABLE VII
COMPARISON OF TOTAL PRODUCTION COST AND SIMULATION TIME FOR CASE 3

Method	Maximum Cost ($)	Minimum Cost ($)	Average Cost ($)	Mean Actual Simulation Time (min)
EP-SQP [6]	–	3 159 024	3 169 093	–
MHEP-SQP [6]	–	3 151 445	3 157 438	–
CDE method 3 [18]	–	3 083 930	3 090 542	–
CSAPSO [20]	–	3 066 907	3 075 023	–
Proposed EBSO	3 055 944	3 054 001	3 054 697	0.95

TABLE VIII
COMPARISON OF BEST RESULTS OF GAMS FOR CASES 1 AND 4

Optimization Technique	10-Unit System Without Losses	60-Unit System Without Losses
BARON	Infeasible	Infeasible
Conopt	1 075 718	6 280 093
LGO	1 039 846	6 226 293
MINOS	1 040 469	6 229 083
MOSEK	Infeasible	Infeasible
SNOPT	1 035 259	6 194 804
OQNLP	1 033 617	6 188 873
PATHNLP	Infeasible	Infeasible
MSNLP	1 023 879	6 159 983

local optima. This feature proves the convergence reliability of the proposed EBSO algorithm.

2) *Computational Efficiency:* Power mismatch between total power generation and demand plus transmission losses is zero (Table III).

It should be mentioned that the data in Table III are rounded up to four decimal places. Table III confirms that the ramp-rate limits are effectively satisfied.

In order to verify the proposed equality constraint handling method in this paper, the proposed EAPSO method is implemented employing two typical constraint handling methods for Case 1. "Method 1" considers a dependent unit to satisfy the equality constraint [2], while "Method 2" adds a penalty factor to fitness function where each solution is penalized in case of infeasibility with penalty proportional to the extent of constraint violation [19]. The comparative results are mentioned in Table IX.

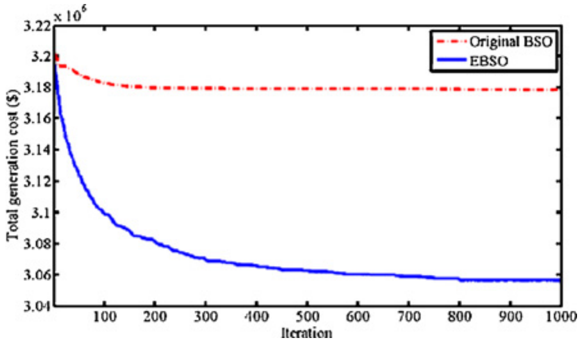

Fig. 4. Convergence characteristics of the EBSO and original BSO for 30-unit system.

TABLE IX
COMPARISONS OF TOTAL PRODUCTION COST EMPLOYING CONSTRAINT HANDLING METHODS FOR CASE 1 (40 RUNS)

Method	Maximum Cost ($)	Minimum Cost ($)	Average Cost ($)	Maximum Violation	Mean Actual Simulation Time (min)
Method 1 [2]	1 018 509	1 017 398	1 017 994	10^{-10}	0.2
Method 2 [19]	1 018 399	1 017 291	1 017 982	10^{-2}	0.16
Proposed method	1 017 891	1 017 147	1 017 526	10^{-10}	0.2

From Tables II, VI, and IX, the EBSO algorithm is efficient as far as computational time is concerned, so as a whole, it seems the EBSO algorithm is computationally more efficient than the earlier cited methods.

3) *Robustness:* Similar to other evolutionary algorithms, the EBSO uses the stochastic techniques and randomness is an intrinsic feature of it. Therefore, the performance of these algorithms was measured based on the results obtained from several independent runs.

Table I shows that the EBSO algorithm is relatively robust for different parameter combinations. Table II shows that the EBSO method provides optimal solutions with satisfactory standard deviation in comparison with the other cited methods. The calculated standard deviation for Cases 3 and 4 are \$294 and \$541, respectively, which again confirm the robustness of the EBSO algorithm.

For further analysis success rate is calculated for Case 1. The success rate is defined as $(Trail_{\mathrm{suc}}/Trail_{\mathrm{tol}}) \times 100$ in this paper where $Trail_{\mathrm{tol}}$ is the total number of the tests carried out and $Trail_{\mathrm{suc}}$ is the number of the successful tests to converge to the best solution. Results of the success rate are provided in Table X. Table X shows the BSO algorithm has satisfactory success rate and is robust.

VI. CONCLUSION

This paper presented the EBSO algorithm to solve the DED problem. The DED problem formulation, including the system, the power losses, the valve-point effects, and the ramp-rate limits, was considered. In the proposed framework, the heuristic strategies were planned to efficiently handle different operating constraints. To enhance the performance of the original BSO, several approaches were put into practice including two improved moving patterns, repulsion techniques for the diversity of the solutions, a nonlinear approach to update weight

TABLE X
SUCCESS RATE OF THE EBSO FOR CASE 1 OUT OF 40 TRIALS

Best cost ($)	Rang of the fuel cost	1 017 147–1 017 831	1 017 831–1 018 131	1 018 131–1 019 131	1 019 131–1 020 131	1 020 131–1 021 131
	Success rate	95%	5%	0	0	0

factors of the algorithm, and a powerful reformation to prevent premature convergence. Consequently, the proposed EBSO provided more optimal solutions with lower computational burden compared to other heuristic optimization methods cited in this paper. The EBSO algorithm was examined on different test systems and its performance was compared with the other recently developed techniques. The comparative study was done in terms of the solution quality, computational efficiency, and the robustness. The simulation results showed that the proposed EBSO method not only provides more optimal solutions for the DED problem in a proper computational time, but also gives solutions with satisfactory standard deviation. Moreover, the EBSO algorithm found solutions with lower cost for DED problem in comparison with different solvers of the GAMS. Furthermore, the effects of the different proposed modifications on the EBSO algorithm were examined and discussed in detail. The research work is under way in order to incorporate security issues (e.g., maximization of static and dynamic stability margins) and devise stochastic optimization approach to cope with the uncertainty in available output power of wind farms and photovoltaic units.

REFERENCES

[1] A. Pathom, K. Hiroyuki, T. Eiichi, and H. Jun, "A hybrid EP and SQP for dynamic economic dispatch with non-smooth fuel cost function," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 411–416, May 2002.

[2] X. Yuan, L. Wang, Y. Yuan, Y. Zhang, B. Cao, and B. Yang, "A modified differential evolution approach for dynamic economic dispatch with valve-point effects," *Energy Convers. Manage.*, vol. 49, no. 12, pp. 3447–3453, Dec. 2008.

[3] T. Victoire and A. Jeyakumar, "Reserve constrained dynamic dispatch of units with valve-point effects," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1272–1282, Aug. 2005.

[4] D. Ross and S. Kim, "Dynamic economic dispatch of generation," *IEEE Trans. Power Apparatus Syst.*, vol. 99, no. 6, pp. 2060–2068, Nov. 1980.

[5] P. Saravuth, N. Issarachai, and K. Waree, "Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints," *Energy Convers. Manage.*, vol. 49, no. 4, pp. 506–516, 2008.

[6] T. Victoire and V. Jeyakumar, "A modified hybrid EP–SQP approach for dynamic dispatch with valve-point effect," *Electr. Power Energy Syst.*, vol. 27, no. 8, pp. 594–601, 2005.

[7] S. Sivasubramani and K. S. Swarup, "Hybrid SOA-SQP algorithm for dynamic economic dispatch with valve-point effects," *Energy*, vol. 35, no. 12, pp. 5031–5036, Dec. 2010.

[8] Y. Lu, J. Zhou, H. Qin, Y. Li, and Y. Zhang, "An adaptive hybrid differential evolution algorithm for dynamic economic dispatch with valve-point effects," *Expert Syst. Applicat.*, vol. 37, no. 7, pp. 4842–4849, Jul. 2010.

[9] J. Lee, W. Lin, G. Liao, and T. Tsao, "Quantum genetic algorithm for dynamic economic dispatch with valve-point effects and including wind power system," *Electr. Power Energy Syst.*, vol. 33, no. 2, pp. 189–197, 2011.

[10] Y. Wang, J. Zhou, H. Qin, and Y. Lu, "Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects," *Energy Convers. Manage.*, vol. 51, no. 12, pp. 2893–2900, Dec. 2010.

[11] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Nov. 2007.

[12] X. S. Yang, "Engineering optimizations via nature-inspired virtual bee algorithms," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer, 2005, pp. 317–323.

[13] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *J. Appl. Math. Comput.*, vol. 214, pp. 108–132, Aug. 2009.

[14] R. Akbari, A. Mohammadi, and K. Ziarati, "A novel bee swarm optimization algorithm for numerical function optimization," *Commun. Nonlinear Sci. Numer. Simulat.*, vol. 15, no. 10, pp. 3142–3155, 2010.

[15] X. Yuan, A. Su, Y. Yuan, H. Nie, and L. Wang, "An improved PSO for dynamic load dispatch of generators with valve-point effects," *Energy*, vol. 34, no. 1, pp. 67–74, 2009.

[16] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial immune system for units with valve-point effect," *Electr. Power Energy Syst.*, vol. 33, pp. 868–874, Feb. 2011.

[17] D. He, G. Dong, F. Wang, and Z. Mao, "Optimization of dynamic economic dispatch with valve-point effect using chaotic sequence based differential evolution algorithms," *Energy Convers. Manage.*, vol. 52, no. 2, pp. 1026–1032, Feb. 2011.

[18] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang, "Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects," *Eng. Applicat. Artif. Intell.*, vol. 24, no. 2, pp. 378–387, 2011.

[19] A. I. Selvakumar, "Enhanced cross-entropy method for dynamic economic dispatch with valve-point effects," *Electr. Power Energy Syst.*, vol. 33, no. 3, pp. 783–790, Mar. 2011.

[20] Y. Wang, J. Zhou, Y. Lu, H. Qin, and Y. Wang, "Chaotic self-adaptive particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects," *Expert Syst. Applicat.*, vol. 38, no. 11, pp. 14231–14237, Oct. 2011.

Taher Niknam received the B.S. degree in electrical engineering from Shiraz University, Shiraz, Iran, and the M.S. and Ph.D. degrees from the Sharif University of Technology, Tehran, Iran.

He is currently with the Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz. His current research interests include optimization, power systems, distributed generation, and optimal operation of distribution networks.

Faranak Golestaneh received the B.S. degree in electrical engineering from the Shiraz University of Technology, Shiraz, Iran.

She is currently with the Department of Electronics and Electrical Engineering, Shiraz University of Technology. Her current research interests include optimization, power systems, microgirds, distributed generation, and optimal operation of distribution networks.