Improved Random Drift Particle Swarm Optimization With Self-Adaptive Mechanism for Solving the Power Economic Dispatch Problem

Wael Taha Elsayed, *Graduate Student Member, IEEE*, Yasser G. Hegazy, *Senior Member, IEEE*, Mohamed S. El-bages, and Fahmy M. Bendary

***Abstract*—This paper proposes an improved version of the random drift particle swarm optimization algorithm for solving the economic dispatch problem. The improvement is achieved through adding a crossover operation followed by a greedy selection process while replacing the mean best position of the particles with the personal best position of each particle in the velocity updating equation. The improved algorithm is also augmented with a self-adaption mechanism that eliminates the need for tuning the algorithm parameters based on characteristics of the considered optimization problem. Practical features such as valve point effects, prohibited operating zones, multiple fuel options, and ramp rate limits are considered in the mathematical formulation of the economic dispatch problem. In order to demonstrate the efficacy of the proposed algorithm, five benchmark test systems are utilized. The obtained results showed that the improved random drift particle swarm optimization algorithm is capable of providing superior results compared to the original algorithm and the state of the art techniques proposed in previous literature.**

***Index Terms*—Economic dispatch (ED) problem, metaheuristic technique, random drift particle swarm optimization (RDPSO), valve point effects.**

NOMENCLATURE

Indices

- i Index of a generating unit or an individual of the swarm.
- j Index of a fuel type, a generating unit, a prohibited operating zone, or an element of a vector.
- t Index of an iteration.

Manuscript received May 6, 2016; revised August 2, 2016, January 15, 2017, February 5, 2017, and March 16, 2017; accepted April 11, 2017. Date of publication April 18, 2017; date of current version June 1, 2017. Paper no TII-16-0431. *(Corresponding author: Wael Taha Elsayed.)*

W. T. Elsayed is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: wtahagha@uwaterloo.ca).

Y. G. Hegazy is with the Faculty of Information and Engineering Technology, German University in Cairo, Cairo 11432, Egypt (e-mail: yasser.higazi@guc.edu.eg).

M. S. El-bages and F. M. Bendary are with the Electrical Engineering Department, Shoubra Faculty of Engineering, Benha University, Cairo 11629, Egypt (e-mail: mohamed.albags@feng.bu.edu.eg; fahmybendary10@gmail.com).

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/TII.2017.2695122

Variables and Parameters

F_T	Total operational cost.
P_i	Power output of unit *i*.
$F_i(P_i)$	Cost function value of unit *i*.
n	Total number of generating units.
P_D	Total system demand.
P_{Loss}	Total transmission losses.
$P_i^{\min}$	Minimum power output of unit *i*.
$P_i^{\max}$	Maximum power output of unit *i*.
P_{i0}	Power output of unit *i* at the previous time interval.
UR_i	Upper-ramp rate limit for unit *i*.
DR_i	Down-ramp rate limit for unit *i*.
$P_{i,j}^l$	Lower limit of the *j*th prohibited zone for unit *i*.
$P_{i,j}^u$	Upper limit of the *j*th prohibited zone for unit *i*.
nj	Number of prohibited operating zones for unit *i*.
a_i, b_i, c_i, $e_{i,}$ and f_i	Fuel cost coefficients of unit *i*.
a_{ij}, b_{ij}, c_{ij}, e_{ij}, and f_{ij}	Fuel cost coefficients of unit *i* using fuel type *j*.
$P_{ij}^{\min}$	Lower bound of unit *i* using fuel type *j*.
$P_{ij}^{\max}$	Upper bound of unit *i* using fuel type *j*.
B_{ij}, B_{0i}, and B_0	Loss coefficients.
X_i^t	Position of particle *i* at iteration *t*.
Y_i^{t-1}	Personal best position of particle *i* at iteration $t-1$.
Y_i^t	Personal best position of particle *i* at iteration *t*.
n_p	Population size.
X_i^{t-1}	Position of particle *i* at iteration $t-1$.
Y_{i*}^{t-1}	Global best position at iteration $t-1$.
V_i^t	Velocity of particle *i* at iteration *t*.
V_i^{t-1}	Velocity of particle *i* at iteration $t-1$.
ω	Inertia weight.
C_1 and C_2	Acceleration coefficients.
${R_1}_i^t$ and ${R_2}_i^t$	Two random vectors generated in correspondence to particle *i* at

iteration t with elements distributed uniformly between zero and one.

$\varphi_{i,j}^t$	Parameter of the random drift particle swarm optimization algorithm determined in correspondence to the jth element of particle i at iteration t.
$Z_{i,j}^t$	jth element of the local focus position for particle i at iteration t.
d	Problem dimension.
$R_1 i, j^t$ and $R_2 i, j^t$	Two different random numbers distributed uniformly between zero and one. Each of these two numbers are generated in correspondence to the jth element of particle i at iteration t.
$X_{i,j}^{t-1}$	jth element of the position for particle i at iteration $t-1$.
C_j^t	jth element of the mean best position at iteration t.
α	Thermal coefficient.
$\delta_{i,j}^t$	Random number that obeys the standard normal distribution generated in correspondence to the jth element of particle i at iteration t.
$VT_{i,j}^t$	jth element of the velocity of the thermal motion associated with particle i at iteration t.
$Y_{i,j}^{t-1}$	jth element of the personal best position for particle i at iteration $t-1$.
Z_i^t	Local focus position for particle i at iteration t.
β	Drift coefficient.
VD_i^t	Velocity of the drift motion associated with particle i at iteration t.
$V_{i,j}^t$	jth element of particle i' velocity at iteration t.
$X_{i,j}^t$	jth element of particle i' position at iteration t.
$r_{i,j}^t$	Random number uniformly distributed between zero and one, and generated in correspondence to the jth element of particle i at iteration t.
C	Crossover probability.
$X_{i,j(\mathrm{trial})}^t$	jth element of the trial vector generated in correspondence to particle i at iteration t.

I. Introduction

The economic dispatch (ED) problem is to find the most economic distribution of system loads among system generators. Theoretically, the problem can be modeled as an optimization problem with a convex cost function. This optimization problem is relatively easy to solve; however, in real power systems, practical features such as valve point effects, prohibited operating zones, ramp rate limits, and multiple fuel options are usually encountered. Neglecting these features, improper or approximated handling of these features while solving the ED problem may lead to significant monetary losses through presenting inaccurate solutions of the problem [1], [2]. On the other hand, considering these practical features converts the ED problem to a complex and hard to solve optimization problem in which the cost function is nonconvex and nonsmooth. In the previous literature, many techniques have been proposed for solving the ED problem. Classical methods such as the lambda iteration method require a convex cost function. The dynamic programing does not impose any constraints on the shape of the cost function; however, as the problem dimension increases, the computational effort required by the dynamic programming exponentially increases, and large-scale ED problems cannot be solved within a satisfactory computational time. Similar to the dynamic programming, the metaheuristic techniques do not impose specific characteristics of the cost function. There are numerous metaheuristic techniques proposed in the previous literature, and several techniques succeed to find the global optimal solution of some benchmark systems. Some metaheuristic techniques have been applied in their basic form for solving the ED problem such as particle swarm optimization (PSO) [3] and differential evolution (DE) [4]. Modified and improved versions of the metaheuristic techniques have also been applied for solving the ED problem such as modified particle swarm optimization (MPSO) [5] and improved differential evolution [6]. Hybrid methods in which two or more metaheuristic techniques are combined to solve the ED problem have been proposed such as in [7]. One major limitation that exists in many of the previously proposed metaheuristic techniques is the need for tuning the parameters of these techniques before applying them for solving a specific ED problem. A certain setting of parameters for a certain metaheuristic technique may provide satisfactory results with specific benchmark systems. However, this specific setting may fail to provide satisfactory results with another set of benchmark systems. It even may fail if the benchmark system characteristics have been changed such as changing the total system load or the number of generating units. To solve this problem, the metaheuristic techniques have to be augmented with a self-tuning capability. Nonetheless, augmenting any metaheuristic technique with a self-tuning capability may reduce the efficacy of the results obtained by this technique due to the fact that some additional computational efforts have to be consumed by the algorithm for tuning its own parameters while solving the main optimization problem. As a result, to augment any metaheuristic technique with a self-tuning capability, this technique has to be a highly efficient technique. A novel efficient variant of the PSO algorithm, random drift particle swarm optimization (RDPSO), has been proposed in [8]. Sun *et al.* [8] showed that the RDPSO can compete and outperform many of the state of the art metaheuristic techniques. This paper proposes two modifications to the RDPSO algorithm in order to increase the efficacy of the RDPSO algorithm significantly. The proposed modifications are to add a crossover operation followed by a greedy selection process and to replace the mean best position of the particles with the personal best position of each particle in the velocity updating equation. An improved random drift particle swarm optimization (IRDPSO) algorithm is then developed. In addition, a self-adaption mechanism is suggested to produce a self-tuning IRDPSO (ST-IRDPSO) algorithm. The proposed IRDPSO and ST-IRDPSO algorithms have the following advantages:

1) Remarkable money savings can be achieved compared to the savings attained by the previously proposed state of the art techniques in the field of solving the nonconvex economic dispatch problem. These savings are achieved through providing lower minimum cost values, average cost values, and standard deviation values compared to those obtained by the state of the art techniques.
2) The execution time required for a single run of the proposed algorithms is much less than the 5 min interval, which is required for updating the load forecasting [9], [10].
3) The self-tuning capability of the ST-IRDPSO algorithm minimizes the human interference, which helps to minimize the operation cost, the associated human errors, and the required time for tuning the algorithm parameters.

The paper is organized as follows. The ED problem formulation is presented in Section II. Section III reviews the PSO and RDPSO algorithms. The proposed IRDPSO algorithm with the self-tuning capability is introduced in Section IV. Simulation results are shown in Section V. Discussion of the results and expected future studies are presented in Section VI, followed by the conclusion in Section VII.

II. Mathematical Formulation of the ED Problem

A comprehensive mathematical formulation of the ED problem is as follows:

$$\text{Minimize}: F_T = \sum_{i=1}^{n} F_i(P_i). \tag{1}$$

$$\text{Subject to}: \sum_{i=1}^{n} P_i = P_D + P_{\text{Loss}} \tag{2}$$

$$\max(P_i^{\min}, P_{i0} - DR_i) \le P_i \le \min(P_i^{\max}, P_{i0} + UR_i) \tag{3}$$

$$\begin{aligned} &P_i^{\min} \le P_i \le P_{i,1}^{l} && (i = 1, 2, ..., n)\\ &P_{i,j-1}^{u} \le P_i \le P_{i,j}^{l} && (j = 2, 3, ..., nj)\ (i = 1, 2, ..., n)\\ &P_{i,nj}^{u} \le P_i \le P_i^{\max} && (i = 1, 2, ..., n). \end{aligned} \tag{4}$$

If the valve point effects are not considered, the cost function is convex and can be modeled as follows:

$$F_i(P_i) = a_i + b_i\, P_i + c_i\, P_i^2. \tag{5}$$

On the other hand, when the valve point effects are considered, the cost function becomes nonconvex and can be written as follows:

$$F_i(P_i) = a_i + b_i\, P_i + c_i\, P_i^2 + \left| e_i \sin(f_i \times (P_i^{\min} - P_i))\right|. \tag{6}$$

If there are units with multiple fuel options in the system, the cost function of these units while considering the valve point effects can be modeled as follows [11]:

$$F_i(P_i) = a_{ij} + b_{ij} P_i + c_{ij} P_i^2 + \left|e_{ij} \times \sin(f_{ij} \times (P_{ij}^{\min} - P_i))\right| \quad \text{if} \quad P_{ij}^{\min} \le P_i \le P_{ij}^{\max}. \tag{7}$$

Equation (2) represents the power balance constraint in which P_{Loss} is computed as follows:

$$P_{\text{Loss}} = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00}. \tag{8}$$

Equation (3) represents the upper and lower bounds imposed to the generated power of unit *i* in the case of considering the ramp rate limits. If the ramp rate limits are not considered, constraint (3) is replaced by the following constraint:

$$P_i^{\min} \le P_i \le P_i^{\max}. \tag{9}$$

Finally, (4) is used to model the prohibited operating zones.

III. PSO and RDPSO Algorithms

The following sections review the PSO and RDPSO algorithms.

A. PSO Algorithm

The PSO is an algorithm used for finding the global optimal solution of nonsmooth and nonconvex optimization problems. The concept of the PSO algorithm was inspired while simulating a simplified social model of bird flocks or fish schools [12]. The PSO achieved a remarkable success through presenting high-quality solutions for nonsmooth and nonconvex optimization problems in many research fields and applications. The success of the PSO algorithm oriented a great portion of the research toward improving the algorithm. Many hybrid methods that incorporate the PSO algorithm have also been tested in different disciplines. The PSO algorithm starts by initializing a population of particles with size n_p. Each particle has a position X_i in a *d*-dimensional space. The particle moves from one position to another in the search space with a velocity V_i. The position of each particle *i* at iteration *t* represents a possible solution to the optimization problem under consideration. The best position found so far by a particle *i* is known as the personal best Y_i. The personal best is updated from one iteration to the next in a minimization problem as follows:

$$Y_i^t = \begin{cases} X_i^t \text{ if} & f(X_i^t) < f(Y_i^{t-1}) \\ Y_i^{t-1} \text{ if} & f(X_i^t) \ge f(Y_i^{t-1}) \end{cases} \tag{10}$$

where $i \,\epsilon\, N$ and $N = 1, 2, \ldots, n_p$. The best solution found so far in the whole population is known as the global best Y_{i*}^t where

$$i^* = \arg \min_{i \in N}\ f(Y_i^t). \tag{11}$$

The velocity equation, by which the particles update their positions, is given as follows:

$$\begin{aligned} V_i^t = \omega\, V_i^{t-1} &+ C_1\, R_{1i}^{\ t}(Y_i^{t-1} - X_i^{t-1}) \\ &+ C_2\, R_{2i}^{\ t}(Y_{i*}^{t-1} - X_i^{t-1}). \end{aligned} \tag{12}$$

The inertia weight ω is used to balance the local and global search of the particles [8]. After calculating the velocity of particle *i*, the position of particle *i* is updated in each iteration as follows:

$$X_i^t = X_i^{t-1} + V_i^t. \tag{13}$$

B. RDPSO Algorithm

According to [8] and the trajectory analysis provided in [13], moving each particle toward its local focus improves the convergence characteristics of the PSO algorithm. Based on this, Sun *et al.* [8] modified the velocity equation of the PSO algorithm such that the particles move toward their local focus, which is calculated using [8]

$$Z_{i,j}^{t} = \varphi_{i,j}^{t}\, Y_{i,j}^{t-1} + (1 - \varphi_{i,j}^{t})\, Y_{i*,j}^{t-1} \tag{14}$$

where $j \epsilon D$ and $D = 1, 2, \ldots, d$. In (14), $\varphi_{i,j}^{t}$ is calculated as follows:

$$\varphi_{i,j}^{t} = \frac{C_1\, R_{1i,j}^{t}}{C_1\, R_{1i,j}^{t} + C_2\, R_{2i,j}^{t}}. \tag{15}$$

The RDPSO is inspired by the model of a free electron moving in a metal conductor exposed to an electric field. According to the free electron model [8], [14], in addition to the random movement caused by the electron thermal motion, the electron is also in a drift motion due to the electric field to which it is exposed. Therefore, the movement of the electron is the superposition of the thermal motion and the drift motion. Thereafter, based on the free electron model, the velocity of the particles in the RDPSO algorithm has two components. The first one represents the velocity of the thermal motion and is computed as follows [8]:

$$VT_{i,j}^{t} = \alpha \left| C_j^{t} - X_{i,j}^{t-1} \right| \delta_{i,j}^{t}. \tag{16}$$

In (16), the jth element of the mean best position at iteration t (C_j^t) is computed as follows:

$$C_j^{t} = \frac{\sum_{i=1}^{n_p} Y_{i,j}^{t-1}}{n_p}. \tag{17}$$

The second component represents the velocity of the drift motion and is computed as follows:

$$VD_i^{t} = \beta\, (Z_i^{t} - X_i^{t-1}). \tag{18}$$

The effect of the drift velocity is to pull the solution toward the local focus. Combining the two types of motion, the velocity equation of the RDPSO algorithm can be expressed with

$$V_{i,j}^{t} = \alpha \left| C_j^{t} - X_{i,j}^{t-1} \right| \delta_{i,j}^{t} + \beta\, (Z_i^{t} - X_i^{t-1}). \tag{19}$$

After calculating the velocity of each particle using (19), each particle updates its position using (13).

IV. Proposed Algorithms

The proposed improvement of the RDPSO and suggested self-adaptive mechanism are presented in this section. The first section discusses the modifications proposed to improve the RDPSO algorithm, and the second section presents the proposed self-adaption mechanism.

A. Improved RDPSO Algorithm

Two modifications are proposed to improve the RDPSO performance. These two modifications significantly improve the RDPSO algorithm performance as demonstrated in Section V of this paper. The first modification is to add a crossover operation to the RDPSO algorithm. This modification enhances the diversity of the population, and hence it improves the performance of the algorithm. In the crossover operation, a mating process is done between the new solution obtained by (13) and the local best position to create a new trial vector, as described by the following equation:

$$X_{i,j\,(\text{trial})}^{t} = \begin{cases} X_{i,j}^{t} & \text{if } r_{i,j}^{t} < C \\ Y_{i,j}^{t-1} & \text{otherwise} \end{cases}. \tag{20}$$

After the crossover operation, a greedy selection process [15], [16] is applied in which the local best position is replaced by the trial vector if the latter has a lower fitness value.

Based on extensive experimentation with the RDPSO algorithm, the authors observed that if the personal best is used instead of the mean best position in (16), the minimum total cost obtained over a certain number of runs will be improved. Therefore, the second proposed modification is to replace the mean best position in (16) with the local best position. Consequently, the velocity of the thermal motion is computed in the IRDPSO algorithm using

$$VT_{i,j}^{t} = \alpha\, (Y_{i,j}^{t-1} - X_{i,j}^{t-1})\, \delta_{i,j}^{t}. \tag{21}$$

The velocity equation of the IRDPSO is now described as follows:

$$V_{i,j}^{t} = \alpha\, (Y_{i,j}^{t-1} - X_{i,j}^{t-1})\, \delta_{i,j}^{t} + \beta\, (Z_i^{t} - X_i^{t-1}). \tag{22}$$

Similar to the PSO and RDPSO algorithms, the IRDPSO algorithm starts by initializing a population of particles. In this population, each particle has a position X_i in a d-dimensional space, and moves from one position to another with a velocity V_i. The position of each particle i at iteration t represents a possible solution to the optimization problem under consideration. In the first iteration, the local best Y_i is set equal to the initial population. In the subsequent iterations, the local best position is updated in a minimization problem using (10). During evaluating the fitness function for each solution, the solution that violates any of the constraints is penalized by assigning it a remarkable high cost value. This strategy is used to guide the algorithm toward the feasible solution region. The IRDPSO algorithm can be summarized as follows:

The IRDPSO algorithm	
1:	Initialize the population of particles randomly
2:	Set the local best equal to the current population
3:	Evaluate the fitness function for the population and determine the global best position
4:	**While** stopping criteria is not met **do**
5:	Calculate the local focus using (14) and (15)
6:	Calculate the velocity of the particles using (22)
7:	Calculate the new position of the particles using (13)
8:	Perform the crossover operation using (20)
9:	Update the local best position using a greedy selection
10:	Update the global best position
11:	**End While**

B. Sensitivity Analysis and Parameters Selection for the Self-Adaptive Mechanism

Before applying the IRDPSO for solving the ED problem, the parameters of the IRDPSO algorithm have to be tuned. As a solution for tuning the parameters of a metaheuristic technique, chaotic sequences have been used in some previous publications, such as in [17], to provide values for the algorithm parameters during the search. Although the chaotic sequences slightly mitigate the dependence of the algorithm performance on the nature of the problem considered and eliminate the need for manual tuning of the algorithm, there is no learning process involved in using chaotic sequences in order to improve or adapt the values of the algorithm parameters during the search. Therefore, there are a lot of chances that the chaotic sequences fail to find the suitable parameters setting during the search. Consequently, chaotic sequences are not used in this paper to adapt the algorithm parameters, and a different methodology that involves a learning process is used for assigning values to the algorithm parameters.

The IRDPSO algorithm has five parameters; the acceleration coefficients C_1 and C_2, the thermal coefficient, the drift coefficient, and the crossover probability. The algorithm performance varies with the chosen parameters setting and the problem considered. A certain parameters setting can be used to provide satisfactory results for a specific optimization problem. However, if the problem has been changed, another parameters setting may be required to provide satisfactory results. Allowing the IRDPSO algorithm for tuning all its own five parameters increases the problem dimension by five and reduces the IRDPSO capability of finding the global optimal solution for the original problem. On the other hand, if one or two parameters only of the IRDPSO have been chosen to be self-adaptive, the self-adaption capability of the algorithm will be weak, since there are at least three other parameters requiring tuning based on the optimization problem considered. Therefore, a compromise solution is required. It is observed using extensive experimentation that, among the five parameters of the IRDPSO algorithm, the crossover probability and the drift velocity coefficient become less sensitive to the problem characteristics when assigned specific values, and hence fixed values can be chosen for them. A value of the drift coefficient equal to 1 has been found to provide satisfactory results for all the considered case studies. In addition to this, a crossover probability value of 0.6 provided satisfactory results in all the case studies except in the case of large-scale systems, such as 140-units system, in which a value of 0.9 provided superior results. On the other hand, the thermal coefficient and the acceleration coefficients have to be varied from one problem to another in order to provide satisfactory results in all the case studies. Based on this, the IRDPSO algorithm has been augmented with a self-adaptive mechanism, in which the thermal coefficient and the acceleration coefficients have been tuned by the IRDPSO itself. This is done by increasing the problem dimension by three. Before starting the IRDPSO, the three parameters are generated randomly and are used by the algorithm in the first iteration. If the solution obtained by the IRDPSO algorithm in a certain iteration is better than that in the previous iteration, the parameters setting used to generate this better solution will remain the same in the next iteration, otherwise, the parameters obtained by the IRDPSO algorithm in the current iteration will replace the existing parameters in the next iteration.

TABLE I
LIST OF ACRONYMS FOR ALGORITHMS REPORTED IN THE PREVIOUS LITERATURE

Method	Acronyms
Simulated annealing	SA ([18], [20])
Tabu search	TS [18]
Particle swarm optimization	PSO ([18], [20], [23])
Tabu search algorithm	TSA [19]
Genetic algorithms	GA [3]
Chaotic teaching-learning-based optimization with Lévy flight	CTLBO [20]
Random drift particle swarm optimization	RDPSO [8]
Chaotic bat algorithm	CBA [21]
Conventional genetic algorithm with multiplier updating	CGA-MU [22]
Improved genetic algorithm with multiplier updating	IGA_MU [22]
Differential evolution	DE [23]
Particle swarm optimization with local random search	PSO-LRS [23]
Fuzzy adaptive particle swarm optimization	FAPSO [24]
Improved differential evolution	IDE [6]
Modified particle swarm optimization	MPSO [5]
Self-tuning hybrid differential evolution	ST-HDE [26]
Bat algorithm	BA [27]
Particle swarm optimization technique with the sequential quadratic programming	PSO–SQP [28]
Evolutionary programming with the sequential quadratic programming (SQP)	EP–SQP [28]
Hybrid differential evolution and gravitational search algorithm	DEGSA [29]
Root tree optimization	RTO [30]
Teaching learning based optimization	TLBO [31]
Group search optimizer	GSO [32]
Immune algorithm for economic dispatch problem	IA_EDP [33]
Chaotic particle swarm optimization	CPSO [34]
Fuzzy adaptive chaotic ant swarm optimization algorithm and the sequential quadratic programming	FCASO-SQP [35]
Tournament-based harmony search with tournament size (t) equal to eight	THS ($t = 8$) [36]
Continuous greedy randomized adaptive search procedure with a self-adaptive differential evolution approach	C-GRASP–SaDE [37]
Grey wolf optimization	GWO [38]
PSO augmented with chaotic sequences and crossover operation	CCPSO [11]
PSO with a certain constraint treatment strategy	CTPSO [11]
Continuous quick group search optimizer	CQGSO [32]
Differential Evolution based on truncated Levy-type flights	DEL [39]
Cuckoo search algorithm	CSA [40]
Kinetic gas molecule optimization	KGMO [41]
Synergic predator-prey optimization	SPPO [42]

V. SIMULATION RESULTS

Five case studies, with one benchmark test system in each case study, are presented for validating the performance of the proposed algorithms. The benchmark systems are 6-units system, 10-units system, 13-units system, 40-units system, and 140-units system. The performance of the IRDPSO and the ST-IRDPSO algorithms have been compared with the performance of several state of the art algorithms. Table I presents a list of these algorithms and their respective acronyms. There are 36

TABLE II
SIX GENERATORS TEST SYSTEM: STATISTICAL RESULTS OF PROPOSED ALGORITHMS AND PREVIOUS LITERATURE

Method	Minimum cost ($/h)	Average cost ($/h)	Average time per run (s)	Standard deviation
SA [18]	15461.10	15488.98	50.36	28.3678
TS [18]	15454.89	15472.56	20.55	13.7195
PSO [18]	15450.14	15465.83	6.82	10.1502
TSA [19]	15451.631	15462.263	5.98	18.09
GA [3]	15459	15469	41.58	0.057
CTLBO [20]	15,441.697*	15441.763	NA	0.0194
RDPSO [8]	15442.757*	15445.024	NA	2.28
CBA [21]	15,450.238	15,454.76	0.704	2.965
RDPSO	15449.89	15458.01	0.707	13.647
IRDPSO	**15449.89**	**15456.55**	**0.676**	**10.9865**
ST-IRDPSO	**15449.89**	**15450.70**	**0.727**	**1.416**

*The computed total losses corresponding to the optimal generation values reported in [20] and [8] do not obey (8).
NA: - Data are Not Available.

algorithms in Table I. The references for these algorithms are also indicated in the same table.

The values of the parameters used for the original RDPSO are the same as those reported in [8]. These values are $C_1 = C_2 = 2$, drift coefficient $\beta = 1.5$, and the thermal coefficient α decreases from 0.9 to 0.3 as the search progresses. For the IRDPSO, the values of the parameters used with the RDPSO have also been used except for $\beta = 1$. The crossover probability in the IRDPSO is selected to be 0.6. For the ST-IRDPSO algorithm, all the parameters are either self-tuned or fixed at specific values, as explained in Section. IV-A, population size of 10 is used in the RDPSO, IRDPSO, and ST-IRDPSO algorithms for all the case studies. The statistical results are obtained using 100 runs in each case study. The MATLAB platform on a personal computer with core i7 (2.4 GHz) processor and 8 GB of RAM has been used for simulating all the case studies.

A. Case Study 1

Prohibited operating zones, ramp rate limits, and transmission losses are considered in this case study. The used benchmark system has six thermal units. The total system load is 1263 MW. The cost coefficients, loss coefficients, and generator limits data of this benchmark system are provided in [3]. The prohibited operating zones and the ramp rate limits are also the same as those defined in [3]. Total number of iterations is fixed to 1000. The average time required by a single run of the ST-IRDPSO algorithm is 727.5×10^{-3} s. Table II shows a comparison between the results obtained by the proposed algorithms and the results obtained by other algorithms from previous literature. Fig. 1 shows the convergence characteristics of the RDPSO, IRDPSO, and ST-IRDPSO algorithms. The convergence characteristics in Fig. 1 are recorded for the run where the minimum cost value is obtained. From Fig. 1, it can be observed that the three algorithms converged to the minimum cost in less than 200 iterations.

As noted in Table II, the IRDPSO and the ST-IRDPSO algorithms have provided satisfactory results compared to the results in the previous literature. However, it should be noted that

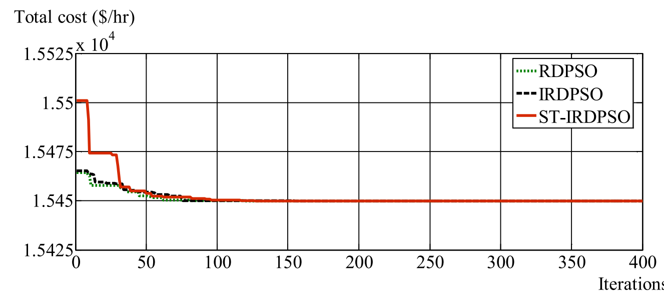


Fig. 1. Convergence characteristics of the RDPSO, IRDPSO, and St-IRDPSO algorithms (6-units system).

TABLE III
OUTPUT POWER OF SIX-GENERATOR TEST SYSTEM USING ST-IRDPSO

P_1	P_2	P_3	P_4	P_5	P_6
447.5131	173.2975	263.4668	139.0360	165.4843	87.16047
Total power (MW)		1275.958	Total losses (MW)		12.958
Power Mismatch (MW)		-99.13×10^{-12}	Total Cost ($/h)		15449.8945

ST-IRDPSO is the only algorithm that includes the self-tuning capability. Excluding the average cost values of the CTLBO [20] and RDPSO [8], the ST-IRDPSO has the lowest average cost value with a low standard deviation value. The GA [3] has a lower standard deviation value compared to that of the ST-IRDPSO; however, this is not an advantage in the GA [3], since the average cost value of the GA [3] is much higher than the corresponding value of the ST-IRDPSO, which implies that the GA has been trapped in a local minima during most of the runs. The standard deviation value of the CTLBO [20] is lower than that of the ST-IRDPSO, yet the optimal solution reported in [20] does not satisfy (8). Based on the comparison in Table II, it can be concluded that the ST-IRDPSO algorithm presented the best performance followed by the IRDPSO and the CBA [21] algorithms. The optimal solution obtained by the proposed algorithms is shown in Table III. This solution achieves a power mismatch equal to -99.13×10^{-12} MW.

B. Case Study 2

In this case study, the multiple fuel options and the valve point effects are considered. The benchmark system used in this case study has ten units. All the relevant data of this system are available in [22]. The total system load is 2700 MW. The total number of iterations is 5000. Table IV presents the minimum cost value, average cost value, standard deviation, and average computational time per one run for the RDPSO, IRDPSO, ST-IRDPSO, and other algorithms reported in the previous literature. The average time of a single run of the ST-IRDPSO is 0.845 s.

According to [2], the total computed cost values for the optimal solutions reported in [5] and [6] are examples of class I inaccuracy. Excluding the results of MPSO [5] and IDE [6], it can be observed from Table IV that the IRDPSO and the ST-IRDPSO succeed to provide minimum and average cost values better than the RDPSO and many other algorithms proposed

TABLE IV
TEN GENERATORS TEST SYSTEM: STATISTICAL RESULTS OF PROPOSED ALGORITHMS AND PREVIOUS LITERATURE

Method	Minimum cost ($/h)	Average cost ($/h)	Average time per run (s)	Standard deviation
CGA-MU [22]	624.7193	627.6087	26.64	NA
IGA_MU [22]	624.5178	625.8692	7.32	NA
DE [23]	624.5146	624.5246	2.8236	NA
PSO [23]	624.5074	624.5074	3.3852	NA
PSO-LRS [23]	624.2297	625.7887	NA	NA
FAPSO [24]	624.2189	624.2782	5.9	NA
IDE [6]*	608.1227	608.2533	2.34	NA
MPSO [5]*	607.98	607.99	2.93	NA
RDPSO	623.915	623.989	0.842	0.0276
IRDPSO	**623.83**	**623.838**	**0.846**	**0.00519**
ST-IRDPSO	**623.83**	**623.836**	**0.845**	**0.00529**

*The total computed cost values by the authors for the optimal solutions reported in [5] and [6] are much higher than those reported in these references.
NA: - Data are Not Available.

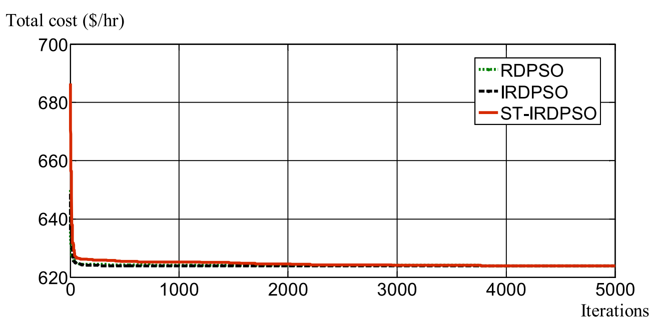


Fig. 2. Convergence characteristics of the RDPSO, IRDPSO, and ST-IRDPSO algorithms (10-units system).

in the previous literature. Table IV shows that the difference between the average cost value and the minimum cost value is very small for the proposed algorithms. The standard deviation values of the IRDPSO and ST-IRDPSO are lower than the standard deviation value of the RDPSO. Fig. 2 depicts the convergence characteristics of the proposed algorithms in addition to the RDPSO algorithm. In this case study, the convergence characteristics of the IRDPSO algorithm is better than that of the ST-IRDPSO algorithm, whereas the IRDPSO algorithm converged in less than 300 iterations, as shown in Fig. 2.

C. Case Study 3

In the third case study, the valve point effects are considered. The benchmark system considered in this case study has 13 units. The cost functions of all the units are nonconvex due to the valve point effects. The cost function coefficients and the generator limits are provided in [25]. The total system load is 1800 MW. The total number of iterations in this case study is 13 500. Table V provides a comparison between the results obtained by the following algorithms: RDPSO, IRDPSO, ST-IRDPSO, ST-HDE [26], BA [27] PSO–SQP [28], EP–SQP [28], DEGSA [29], SA [20], PSO [20], CTLBO [20], and RTO [30].

The average computational time required by one run of the ST-IRDPSO is 2.28 s. The best minimum cost and the best average cost are obtained by the ST-IRDPSO algorithm followed by the IRDPSO algorithm. The best standard deviation value is obtained by the IRDPSO followed by the ST-IRDPSO. Both the DEGSA [29] and the ST-IRDPSO provided the same minimum cost value; however, the average cost and the standard deviation values of the ST-IRDPSO are lower than those of the DEGSA [29]. Table V shows also that the RDPSO and the IRDPSO algorithms provide the lowest values with respect to the average computational time per run. Fig. 3 displays the convergence curves of the RDPSO, IRDPSO, and ST-IRDPSO algorithms.

TABLE V
THIRTEEN GENERATORS TEST SYSTEM: STATISTICAL RESULTS OF PROPOSED ALGORITHMS AND PREVIOUS LITERATURE

Method	Minimum cost ($/h)	Average cost ($/h)	Average time per run (s)	Standard deviation
ST-HDE [26]	17,963.89	18,046.38	NA	NA
BA [27]	17,963.83	18,085.06	NA	NA
PSO–SQP [28]	17,969.93	18,029.99	33.97	NA
EP–SQP [28]	17,991.03	18,106.93	121.93	NA
DEGSA [29]	17963.83	17994.04	9.61	27.75
SA [20]	18048.17	18173.73	NA	81.2
PSO [20]	17975.65	18253.82	NA	179
CTLBO [20]	17972.81	18013.38	NA	43.2
RTO [30]	17969.802	18056.936	NA	NA
RDPSO	17972.83	18039.24	2.25	52.34
IRDPSO	**17965.848**	**17972.8090**	**2.26**	**0.8326**
ST-IRDPSO	**17963.83**	**17966.57**	**2.28**	**3.307**

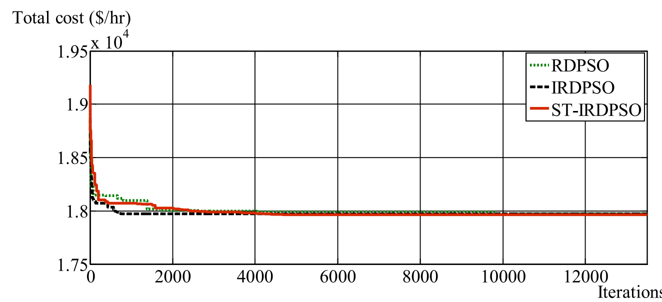


Fig. 3. Convergence characteristics of the RDPSO, IRDPSO, and ST-IRDPSO algorithms (13-units system).

D. Case Study 4

In this case study, a 40-units benchmark system is utilized. The valve point effects are considered in all the units. The cost function coefficients and the generator limits of this system can be found in [25]. This problem is difficult to solve due to the existence of 40 nonconvex cost functions. Solving the ED problem of this system may be more difficult than that of real systems in which some generating units have valve point effects instead of all the system units. Finding the global optimal solution of this system consumed several years of research in which numerous number of algorithms have been studied. Table VI provides a comparison between the performance of the RDPSO, IRDPSO, ST-IRDPSO, and other algorithms in the previous literature.

Among all the algorithms in Table VI, the results of the ST-IRDPSO have the lowest average cost value and the lowest standard deviation. The lowest minimum cost value has also been provided by the ST-IRDPSO algorithm, as indicated

TABLE VI
FORTY GENERATORS TEST SYSTEM: STATISTICAL RESULTS OF PROPOSED ALGORITHMS AND PREVIOUS LITERATURE

Method	Minimum cost ($/h)	Average cost ($/h)	Average time per run (s)	Standard deviation
TLBO [31]	129960	NA	NA	NA
GSO [32]	124265.4	124609.18	14.636	NA
IA_EDP [33]	121436.972	122492.701	1.092	182.527
CPSO [34]	121,865.23	122,100.87	114.65	NA
FCASO-SQP [35]	121,456.98	122,026.21	133.54	NA
BA [27]	121414.91	122094.67	NA	NA
CTLBO [20]	121553.83	121790.23	NA	150
DEGSA [29]	121412.545	121625.74	40.095	155.93
THS ($t = 8$) [36]	121425.15	121528.65	NA	NA
C-GRASP–SaDE [37]	121414.621	121736.025	NA	166.894
RDPSO	121722.03	121972.90	3.2898	243.798
IRDPSO	**121506.040**	**121623.369**	**3.39**	**103.96**
ST-IRDPSO	**121412.535**	**121443.792**	**3.54**	**33.44**

NA: - Data are Not Available.

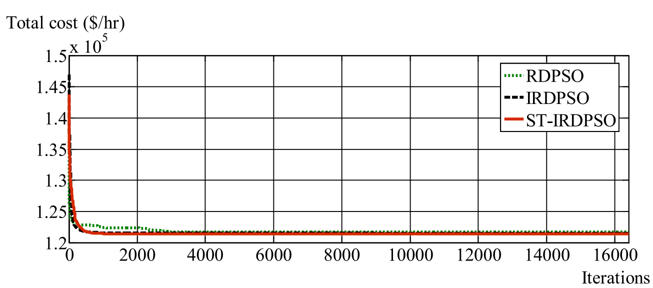


Fig. 4. Convergence characteristics of the RDPSO, IRDPSO, and ST-IRDPSO algorithms (40-units system).

in Table VI. The IA_EDP [33] provided the lowest average computational time; however, the ST-IRDPSO algorithm provided lower minimum cost, average cost, and standard deviation values compared to the ones obtained by the IA_EDP [33]. Fig. 4 presents the convergence curves of the RDPSO, IRDPSO, and ST-IRDPSO. From Fig. 4, it can be observed that the IRDPSO and ST-IRDPSO algorithms converged faster than the RDPSO algorithm to the corresponding minimum cost values.

E. Case Study 5

A real large-scale system is utilized in this case study. This system is the power system of Korea [11]. The system has 140 units. The total system load is 49 342 MW. In this system, 12 units have cost functions with valve point effects, and four units have prohibited operating zones. The complete data of this system are provided in [11]. Table VII shows a comparison between the statistical results of the RDPSO, IRDPSO, ST-IRDPSO, and 12 other algorithms presented in the previous literature.

The best solution observed in the previous literature of this system has a total generation cost equal to 1,559,953.18 $/h [38]. The optimal solution found by the RDPSO, IRDPSO, and ST-IRDPSO has a total generation cost equal to 1,559,708.679 $/h. Both the IRDPSO and the ST-IRDPSO have provided average cost values lower than the ones of the other algorithms. The

TABLE VII
ONE FORTY KOREAN POWER SYSTEM: STATISTICAL RESULTS OF PROPOSED ALGORITHMS AND PREVIOUS LITERATURE

Method	Minimum cost ($/h)	Average cost ($/h)	Average time per run (s)	Standard deviation
GWO [38]	1,559,953.18	1,560,132.93	8.93	1.024
CCPSO [11]	1,657,962.73	1,657,962.73	150	0
CTPSO [11]	1,657,962.73	1,657,964.06	100	7.315
IDE [6]	1,564,648.66	1,564,663.54	27.88	NA
MPSO [5]	1,560,436	1,560,445	18.43	NA
GSO [32]	1,728,151.17	1,745,515.00	53.80	NA
CQGSO [32]	1,657,962.72	1,657,962.74	31.67	NA
DEL [39]	1,657,962.72	1,658,001.70	57.98	57.9836
CSA [40]	1,655,746.14	1,655,904.66	38.90	592.70
KGMO [41]	1,583,944.6	1,583,952.14	28.14	NA
C-GRASP–SaDE [37]	1,657,962.72	1,658,006.27	NA	NA
SPPO [42]	1,655,680.0	1,657,265.04	NA	2872.80
RDPSO	1,559,708.679	1,559,775.46	5.986	105.294
IRDPSO	**1,559,708.679**	**1,559,729.70**	**6.134**	**41.59**
ST-IRDPSO	**1,559,708.679**	**1,559,751.215**	**6.135**	**56.969**

NA: - Data are Not Available.

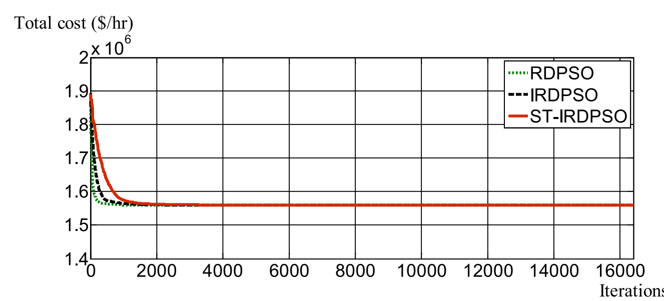


Fig. 5. Convergence characteristics of the RDPSO, IRDPSO, and ST-IRDPSO algorithms (140-units system).

lowest standard deviation value is obtained by GWO [38], but the minimum and average cost values of the ST-IRDPSO and the IRDPSO are lower than those of the GWO [38]. The total number of iterations in one single run is 16 400. The average computational time for a single run of the ST-IRDPSO algorithm is 6.135 s. This time is much lower than the 5 min interval between executing the economic dispatch in real systems [9], [10]. Fig. 5 displays the convergence characteristics of the proposed algorithms and the RDPSO algorithm. From Fig. 5, it can be observed that the RDPSO and the IRDPSO converged slightly faster than the ST-IRDPSO. Generally, Figs. 1–5 confirm that the number of iterations required by the IRDPSO and ST-IRDPSO for convergence to the minimum cost value is comparable to that required by the original RDPSO algorithm, while Tables II and IV–VII show that the IRDPSO and the ST-IRDPSO have provided lower minimum cost, average cost, and standard deviation values than those obtained by the RDPSO algorithm and many state-of-the-art techniques presented in the previous literature.

VI. DISCUSSION OF THE RESULTS AND FUTURE STUDIES

Studying the application of the RDPSO and the IRDPSO algorithms for solving the ED problem of five benchmark systems showed that the average cost and standard deviation values obtained from the IRDPSO are lower than those obtained from the RDPSO algorithm in all the cases. In addition, the minimum cost value obtained from the IRDPSO are lower than the corresponding value obtained from the RDPSO algorithm in all the cases except for the 6-units and 140-units system in which both the algorithms provided the same minimum cost value. These statistical results confirm the superiority obtained from the proposed modifications to the RDPSO algorithm. From Table VI, it can be observed that unsatisfactory results have been obtained by the RDSPO algorithm and the IRDPSO algorithm for the 40-units system due to the fact that the parameters setting used by these algorithms is not optimized for this case study. The same parameters setting have been used in all the case studies. This parameters setting provided satisfactory results in four case studies but failed to provide satisfactory results for the 40-units system. This explains the importance of having a self-tuning capability. The results obtained by the ST-IRDPSO algorithm showed that the algorithm is capable of hitting the global optimal or the best known solution of five benchmark systems while enjoying the advantage of being a self-tunable algorithm. Future research can be oriented to improve the average cost value and to reduce the simulation time further per each run of the ST-IRDPSO algorithm.

VII. Conclusion

An improved version of the RDPSO algorithm is proposed in this paper. The improvements proposed are to add a crossover operation and to replace the mean best position with the personal best position in the thermal velocity component. A self-tuning mechanism is added to the IRDPSO algorithm to eliminate the need for manually tuning the algorithm. To investigate the performance of the proposed algorithms, i.e., IRDPSO and ST-IRDPSO, five benchmark test systems have been utilized. Each benchmark system has been chosen such that it has different features, or different size compared to the other systems. The last system is an example of a real large-scale system for which the IRDPSO and the ST-IRDPSO provided the best minimum and average cost values compared to the corresponding values of previously proposed state-of-the-art techniques. The results showed that the proposed modifications to the RDPSO algorithm have significantly improved the algorithm performance in terms of the minimum cost, average cost, and standard deviation values. In addition, the self-tuning version of the algorithm was capable of hitting the best known solution for each benchmark considered.

References

[1] W. T. Elsayed and E. F. El-Saadany, "A fully decentralized approach for solving the economic dispatch problem," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2179–2189, Jul. 2015.

[2] W. T. Elsayed, Y. G. Hegazy, F. M. Bendary, and M. S. El-Bages, "A review on accuracy issues related to solving the non-convex economic dispatch problem," *Elect. Power Syst. Res.*, vol. 141, pp. 325–332, Dec. 2016.

[3] Z. L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187–1195, Aug. 2003.

[4] N. Noman and H. Iba, "Differential evolution for economic load dispatch problems," *Elect. Power Syst. Res.*, vol. 78, no. 8, pp. 1322–1331, Aug. 2008.

[5] M. Basu, "Modified particle swarm optimization for nonconvex economic dispatch problems," *Int. J. Elect. Power Energy Syst.*, vol. 69, pp. 304–312, 2015.

[6] M. Basu, "Improved differential evolution for economic dispatch," *Int. J. Elect. Power Energy Syst.*, vol. 63, pp. 855–861, 2014.

[7] T. Sen and H. D. Mathur, "A new approach to solve economic dispatch problem using a hybrid ACO–ABC–HS optimization algorithm," *Int. J. Elect. Power Energy Syst.*, vol. 78, pp. 735–744, 2016.

[8] J. Sun, V. Palade, X.-J. Wu, W. Fang, and Z. Wang, "Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 222–232, Feb. 2014.

[9] E. A. Bakirtzis, P. N. Biskas, D. P. Labridis, and A. G. Bakirtzis, "Multiple time resolution unit commitment for short-term operations scheduling under high renewable penetration," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 149–159, Jan. 2014.

[10] L. Xie, Y. Gu, X. Zhu, and M. G. Genton, "Short-term spatio-temporal wind power forecast in robust look-ahead power system dispatch," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 511–520, Jan. 2014.

[11] J.-B. Park, Y.-W. Jeong, J.-R. Shin, and K. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 156–166, Feb. 2010.

[12] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromach. Human Sci.*, Nagoya, Japan, 1995, pp. 39–43.

[13] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 58–73, Feb. 2002.

[14] M. A. Omar, *Elementary Solid State Physics: Principles and Applications*. Reading, MA, USA: Addison-Wesley, 1993.

[15] M. F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Evolutionary algorithms for dynamic economic dispatch problems," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1486–1495, Mar. 2016.

[16] H. Singh and L. Srivastava, "Recurrent multi-objective differential evolution approach for reactive power management," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 1, pp. 192–204, 2016.

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

[18] S. Pothiya, I. Ngamroo, and W. Kongprawechnon, "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.

[19] S. Khamsawang and S. Jiriwibhakorn, "DSPSO–TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions," *Energy Convers. Manage.*, vol. 51, no. 2, pp. 365–375, 2010.

[20] X. He, Y. Rao, and J. Huang, "A novel algorithm for economic load dispatch of power systems," *Neurocomputing*, vol. 171, pp. 1454–1461, 2016.

[21] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X.-S. Yang, "Economic dispatch using chaotic bat algorithm," *Energy*, vol. 96, pp. 666–675, 2016.

[22] C.-L. Chiang, "Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1690–1699, Nov. 2005.

[23] N. Amjady and H. Nasiri-Rad, "Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm," *Expert Syst. With Appl.*, vol. 37, no. 7, pp. 5239–5245, 2010.

[24] T. Niknam, H. D. Mojarrad, and H. Z. Meymand, "Non-smooth economic dispatch computation by fuzzy and self adaptive particle swarm optimization," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2805–2817, 2011.

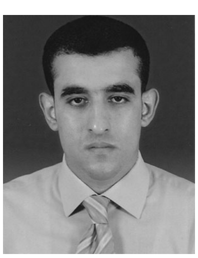
[25] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Trans. Evol. Comput.*, vol. 7, no. 1, pp. 83–94, Feb. 2003.

[26] S.-K. Wang, J.-P. Chiou, and C.-W. Liu, "Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm," *IET Gener., Transmiss. Distrib.*, vol. 1, no. 5, pp. 793–803, 2007.

[27] S. S. S. Hosseini, X. S. Yang, A. H. Gandomi, and A. Nemati, "Solutions of non-smooth economic dispatch problems by swarm intelligence," *Adapt. Hybridization Comput. Intell.*, vol. 18, pp. 129–146, 2015.

[28] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO–SQP for economic dispatch with valve-point effect," *Elect. Power Syst. Res.*, vol. 71, no. 1, pp. 51–59, 2004.

[29] L. D. Le, L. D. Ho, D. N. Vo, and P. Vasant, "Hybrid differential evolution and gravitational search algorithm for nonconvex economic dispatch," in *Proc. 18th Asia Pacific Symp. Intell. Evol. Syst.*, 2015, vol. 2, pp. 89–103.
[30] Y. Labbi, D. Ben Attous, H. A. Gabbar, B. Mahdad, and A. Zidan, "A new rooted tree optimization algorithm for economic dispatch with valve-point effect," *Elect. Power Energy Syst.*, vol. 79, pp. 298–311, 2016.
[31] S. Banerjee, D. Maity, and C. K. Chanda, "Teaching learning based optimization for economic load dispatch problem considering valve point loading effect," *Int. J. Elect. Power Energy Syst.*, vol. 73, pp. 456–464, 2015.
[32] M. Moradi-Dalvand, B. Mohammadi-Ivatloo, A. Najafi, and A. Rabiee, "Continuous quick group search optimizer for solving non-convex economic dispatch problems," *Elect. Power Syst. Res.,* vol. 93, pp. 93–105, Dec. 2012.
[33] V. Aragón, S. Esquivel, and C. C. Coello, "An immune algorithm with power redistribution for solving economic dispatch problems," *Inf. Sci.*, vol. 295, pp. 609–632, 2015.
[34] J. Cai, Q. Li, L. Li, H. Peng, and Y. Yang, "A hybrid CPSO–SQP method for economic dispatch considering the valve-point effects," *Energy Convers. Manage.*, vol. 53, no. 1, pp. 175–181, Jan. 2012.
[35] J. Cai, Q. Li, L. Li, H. Peng, and Y. Yang, "A hybrid FCASO–SQP method for solving the economic dispatch problems with valve-point effects," *Energy*, vol. 38, no. 1, pp. 346–353, Feb. 2012.
[36] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and A. L. Bolaji, "Tournament-based harmony search algorithm for non-convex economic load dispatch problem," *Appl. Soft Comput.*, vol. 47, pp. 449–459, 2016.
[37] J. X. V. Neto, G. Reynoso-Meza, T. H. Ruppel, V. C. Mariani, and L. dos Santos Coelho, "Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution," *Elect. Power Energy Syst.*, vol. 84, pp. 13–24, 2017.
[38] M. Pradhan, P. K. Roy, and T. Pal, "Grey wolf optimization applied to economic load dispatch problems," *Elect. Power Energy Syst.*, vol. 83, pp. 325–334, 2016.
[39] L. dos Santos Coelho, T. C. Bora, and V. Cocco Mariani, "Differential evolution based on truncated Levy-type flights and population diversity measure to solve economic load dispatch problems," *Int. J. Elect. Power Energy Syst.*, vol. 57, pp. 178–188, 2014.
[40] D. N. Vo, P. Schegner, and W. Ongsakul, "Cuckoo search algorithm for nonconvex economic dispatch," *IET Gener., Transmiss. Distrib.*, vol. 7, no. 6, pp. 645–654, 2013.
[41] M. Basu, "Kinetic gas molecule optimization for nonconvex economic dispatch problem," *Elect. Power Energy Syst.*, vol. 80, pp. 325–332, 2016.
[42] N. J. Singha, J. S. Dhillonb, and D. P. Kothari, "Synergic predator-prey optimization for economic thermal power dispatch problem," *Appl. Soft Comput.*, vol. 43, pp. 298–311, 2016.



Wael Taha Elsayed (GSM'13) was born in Qaluobia, Egypt, in 1985. He received the B.Sc. (with honors) and M.Sc. degrees in electrical engineering from the Shoubra Faculty of Engineering, Benha University, Banha, Egypt, in 2007 and 2010, respectively, and the M.Sc. degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in August 2014. He is currently working toward the Ph.D. degree in electrical engineering at the Shoubra Faculty of Engineering, Benha University, Egypt.

In 2007, using a point-based evaluation criterion, he was ranked first in the Faculty of Engineering at Shoubra among 1600 students who graduated in that year and was honored by the President of Egypt at a party held by the Supreme Council of the Egyptian Universities. His current research interests include the economic dispatch problem and applications of decentralized and distributed algorithms in power systems.

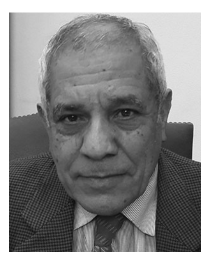
Yasser G. Hegazy (SM'03) received the B.Sc. and M.Sc. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 1986 and 1990, respectively, and the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1996.

Currently, he is a Professor of electrical engineering and the Vice President of the German University in Cairo, Cairo, Egypt. His research interests include power system reliability, distributed generation, renewable energy systems, and power quality.

Dr. Hegazy is the chair of the International Conference of Engineering and Technology. He received the IEEE 2016 outstanding Engineering Award and the 2004 State encouragement Award in Engineering Sciences from the Egyptian Academy of Scientific Research & Technology.

Mohamed S. El-bages received the B.Sc. and M.Sc. degrees from Benha University, Cairo, Egypt, in 1982 and 1991, respectively, and the Ph.D. degree from Cairo University, Giza, Egypt, in 1997, all in electrical power engineering.

Currently, he is an Assistant Professor with the Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha University. His research interests include economic dispatch, load forecasting, power system planning, and operations.



Fahmy M. Bendary received the B.Sc. degree in electrical engineering from Ain Shams University, Cairo, Egypt and the M.Sc. degree in electrical engineering from Cairo University, Giza, Egypt, in 1966 and 1979, respectively, and the Ph.D. degree in electrical engineering from Paris-Sud Université, Orsay, France, in 1983.

From 1984 to 1990, he was a Lecturer at the military technical college, Cairo, Egypt. During the period from 1991 to 1995, he was an Associate Professor at Zagazig University. He became a Full Professor of automatic control and its applications to power system in 1995. He is currently a Professor of power system and automatic control at the faculty of engineering (Shoubra), Benha University, Cairo, Egypt. He supervised around 50 Master's and Ph.D. theses in different universities. He is a coauthor of more than one hundred papers in the field of power system optimization, operation, planning, and control. He participated and attended more than 30 international conferences in Egypt, Tunisia, Saudia Arabia, Europe, and USA. His current research focuses on smart grids and the integration of renewable energy sources into the grid.

Dr. Fahmy is also a member in IEC, CIGRE, and CIRED committees at the ministry of electricity and renewable energy in Egypt. He is also a member of promotion committee for academic staff in the field of electric power systems at the Supreme Council of the Egyptian Universities from 2015 till now.