Evolutionary Algorithms for Dynamic Economic Dispatch Problems

M. F. Zaman, *Student Member, IEEE*, Saber M. Elsayed, *Member, IEEE*, Tapabrata Ray, *Member, IEEE*, and Ruhul A. Sarker, *Member, IEEE*

***Abstract*—The dynamic economic dispatch problem is a high-dimensional complex constrained optimization problem that determines the optimal generation from a number of generating units by minimizing the fuel cost. Over the last few decades, a number of solution approaches, including evolutionary algorithms, have been developed to solve this problem. However, the performance of evolutionary algorithms is highly dependent on a number of factors, such as the control parameters, diversity of the population, and constraint-handling procedure used. In this paper, a self-adaptive differential evolution and a real-coded genetic algorithm are proposed to solve the dynamic dispatch problem. In the algorithm design, a new heuristic technique is introduced to guide infeasible solutions towards the feasible space. Moreover, a constraint-handling mechanism, a dynamic relaxation for equality constraints, and a diversity mechanism are applied to improve the performance of the algorithms. The effectiveness of the proposed approaches is demonstrated on a number of dynamic economic dispatch problems for a cycle of 24 h. Their simulation results are compared with each other and state-of-the-art algorithms, which reveals that the proposed method has merit in terms of solution quality and reliability.**

***Index Terms*—Constrained optimization, constraint handling, differential evolution, dynamic economic dispatch, genetic algorithm, non-uniform mutation.**

I. Introduction

OVER the last few decades, the rapid increase in the use of fossil fuel has led to a consequential worldwide reduction in this resource. Therefore, its optimal utilization in power generation has become an important research topic [1]. Since the operating costs of different generating units vary significantly, it is a challenging problem to schedule the right mix of generation from a number of units to serve a particular load demand at minimum cost, which is known as power economic dispatch problem. It consists of allocating the total generation required among the available thermal generating units, assuming that the commitment of thermal units has been previously determined. The majority of reported work deals with static economic dispatch; that the system was scheduled to serve a particular load level for an hour [2], [3]. However, although this scheduling may be beneficial for a certain hour, it may not work for the next hour (or the next few hours), depending on demand, because the generation from the units may not be changed significantly from one to the next operating hour due to ramp limits. This problem can be minimized by adopting dynamic scheduling for a load cycle of 24 h and considering ramp limits [4]. This scheduling problem is known as the dynamic economic dispatch (DED) problem that aims to minimize the overall operating cost by satisfying hourly load demands, ramp limits, and other constraints. The problem involves ramp limits, generation capacity, and load balance constraints.

Traditionally, the cost function of the thermal generators is a quadratic function, but in real life, large steam generators have a multi-fuel option and some ripple appears on the cost function while the steam is admitted through the valve, which is known as the valve-point effect (VPE) [5]. Therefore, the cost function becomes quadratic, non-smooth, non-convex, and multi-modal characteristics [2], [7]. In addition, a real-life power system encounters some unexpected events, such as unit faults and demand changes. To counter this, a spinning reserve (SR), usually the largest unit's capacity, is maintained in scheduling to increase system reliability. However, it derives a solution from its optimal point because a cheaper unit cannot run at its full capacity [5]. Therefore, finding the optimal scheduling for a DED is not an easy task. To solve DED problems, all the solution approaches appearing in the literature can be categorized as 1) conventional optimization methods and 2) meta-heuristic-based optimization techniques [8].

Conventional optimization methods, such as linear programming-based method [9], lambda iteration method [10], and the interior point method [11], have been widely used to solve the dispatch problem. These techniques are usually computationally efficient, but they deal mainly with convex cost functions [12]. As the cost function with VPE is non-smooth and non-convex, these approaches are unable to generate good quality solutions. Recently, a few researchers developed a mixed integer quadratic programming (MIQP) [8] approach for solving DED with the VPE. In their model formulation, an approximation is considered by linearizing the piecewise convex cost function, whereby the excessive number of linear segments in a large generator introduces many integer variables and additional constraints.

Meta-heuristic-based optimization techniques do not require certain mathematical properties of the objective function to be satisfied and have been successfully applied to many complex practical optimization problems. Over the last few decades, several meta-heuristic methods have been effectively used to

Manuscript received October 25, 2014; revised March 02, 2015; accepted April 29, 2015. Paper no. TPWRS-01480-2014.

The authors are with the School of Engineering and Information Technology, University of New South Wales, ADFA campus, Canberra 2600, Australia (e-mail: md.zaman@student.adfa.edu.au; s.elsayed@adfa.edu.au; t.ray@adfa.edu.au; r.sarker@adfa.edu.au).

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/TPWRS.2015.2428714

solve DED problems, for example, genetic algorithm (GA) [13], simulated annealing (SA) [14], particle swarm optimization (PSO) [15], adaptive PSO (APSO) [15], artificial bee colony (ABC) [13], artificial immune system (AIS) [16], evolutionary programming (EP) [17], improved PSO (IPSO) [18], time-varying acceleration coefficients IPSO (TVACIPSO) [19], improved chaotic PSO (ICPSO) [20], chaotic self-adaptive PSO (CSAPSO) [21], deterministically guided PSO (DGPSO) [22], enhanced bee swarm optimization (EBSO) [23], differential evolution (DE) [24], modified DE (MDE) [25], chaotic DE (CDE) [26], chaotic sequence-based DE (CSDE) [27], self-adaptive modified firefly algorithm (SAMFA) [28], modified teaching-learning algorithm (MTLA) [29], improved enhanced cross-entropy (ECE) [30], harmony search (HS) [31], hybrid swarm intelligence-based HS (HHS) [31], and imperialist competitive algorithm (ICA) [32].

Also, hybrid methods that combine two or more approaches, such as the bee colony optimization and sequential quadratic programming (BCO-SQP) [33], EP-SQP [17], PSO-SQP [7], and modified hybrid EP-SQP (MHEP-SQP) [34], have been applied to solve this problem. In most of these approaches, the equality constraints are usually handled using the penalty-function technique. However, there are too many equality constraints in DED problems that are mutually coupled, which makes it difficult to generate feasible solutions and maintain feasibility after crossover and mutation operations in a meta-heuristic such as GA. To ensure feasibility, Elaiw *et al.* [35] proposed to satisfy the equality constraints sequentially for every hour starting from the first hour of the operational period. However, there is a possibility that this approach will not be able to meet demand during peak periods due to ramp limits.

In this paper, we propose two evolutionary algorithms for solving DED problems, one based on GA and the other on DE. We have carefully considered different components in designing these algorithms. In them, the individuals in the initial population are generated using a new heuristic which ensures that the equality constraints are satisfied in each hour. To ensure feasibility for an entire operational cycle, we consider a look-ahead demand strategy along with the load demand constraints. To maintain diversity in the population, we use a non-uniform mutation in GA. To avoid the difficulty of selecting the best control parameters, a self-adaptive mutation and crossover mechanism is used in DE. We apply the ε-constrained method to handle the equality constraints, with a diversity strategy used to skip from an early convergence. The performances of the proposed approaches in solving five known DED problems taken from [5], [8], and [36] are compared with those of other EAs and recently published state-of-the-art algorithms. The analysis of results ensures that our proposed approaches outperform all the other algorithms compared in this paper. We have analyzed the effect of different components on the performance of these algorithms and demonstrated that these algorithms can be implemented for dynamic scheduling on a rolling horizon basis.

The rest of this paper is organized as follows: Section II presents the problem formulation, Section III the proposed methodology, Section IV the experimental results, Section V analyzing the effect of different components in algorithm design, Section VI dynamic scheduling with rolling horizon framework, and Section VII conclusions and future research.

II. Mathematical Formulation

In this section, a mathematical model of the DED problem is presented [8]:

$$\text{Minimize } F_C = \sum_{t=1}^{T}\sum_{i=1}^{N_T} FC_i(P_{GT_{i,t}}) \quad (1)$$

$$\text{where } FC_i(P_{GT_{i,t}}) = a_i + b_i P_{GT_{i,t}} + c_i P_{GT_{i,t}}^2 + \left| d_i \sin\{e_i(P_{GT_i}^{\min} - P_{GT_{i,t}})\}\right| \; t \in T \quad (2)$$

$$\text{subject to } \sum_{i=1}^{N_T} P_{GT_{i,t}} = P_{D_t} + P_{loss_t} \quad t \in T \quad (3)$$

$$P_{loss_t} = \sum_{i=1}^{N_T}\sum_{j=1}^{N_T} P_{GT_{i,t}}, B_{i,j} P_{GT_{j,t}} \; t \in T \quad (4)$$

$$P_{GT_i}^{\min} \le P_{GT_{i,t}} \le P_{GT_i}^{\max} \quad (5)$$

$$P_{GT_{i,t}} - P_{GT_{i,t-1}} \le UR_i \quad i \in N_T, t \in T \quad (6)$$

$$P_{GT_{i,t-1}} - P_{GT_{i,t}} \le DR_i \quad i \in N_T, t \in T \quad (7)$$

$$\sum_{i=1}^{N_T} P_{GT_i}^{\max} - (P_{D_t} + P_{loss_t} + SR_t) \ge 0 \; t \in T \quad (8)$$

$$\sum_{i=1}^{N_T} \min(P_{GT_i}^{\max} - P_{GT_{i,t}}, UR_i) - SR_t \ge 0 t \in T \quad (9)$$

$$\sum_{i=1}^{N_T} \min\left(P_{GT_i}^{\max} - P_{GT_{i,t}}, \frac{UR_i}{6}\right) - SR_t^m \ge 0 \; t \in T. \quad (10)$$

The objective function (1) is to minimize the sum of all fuel costs for the thermal power plants under consideration (N_T) during the operational cycle (T). The fuel consumption characteristic curve of conventional energy is usually expressed by a quadratic function. However, the sudden opening of the intake valve of steam turbine may cause VPE, which can be reflected by integrating a rectifying sinusoidal wave in the main function. The fuel cost function, including the VPE of each thermal unit, can be expressed in (2), where $a_i(\$/h)$, $b_i(\$/MWh)$, and $c_i(\$/MW^2h)$ are the cost coefficients, $d_i(\$/h)$ and $e_i(rad/MW)$ the valve-point coefficients of the ith unit power plant, and $P_{GT_{i,t}}$ the ith plant's output at the tth hour.

Constraints (3) referred the power balance equation in each cycle. Using the transmission loss (TL) coefficients B, TL of each period is expressed in (4). Constraint (5) is the capacity constraints, where $P_{GT_i}^{\min}$ and $P_{GT_i}^{\max}$ are the minimum and maximum output powers of the ith unit, respectively. Constraints (6) and (7) represent up and down ramp rate limits, respectively, where *UR* and *DR* are the upward and downward transition limits, respectively. Constraints (8) and (9) are used to satisfy the 1-h reserve requirements and constraint (10), 10-min ones for which the ramp is rationally considered as $UR/6$. Here, *SR* and SR^m are the spinning reserves for 1 h and 10 min, respectively.

III. Proposed Methodology

In this paper, we have considered GA [37] and DE [38] based algorithms for solving DED problems. They are population-based algorithms and well-known for their success in solving complex optimization problems. However, sometimes they suffer from premature convergence and become trapped in local solutions specifically when solving multimodal functions [4]. As the performances of these algorithms in solving constrained optimization problems are highly dependent on their control parameters, population diversity, and constraint handling mechanism, we have taken special care in designing our algorithms in this paper. For both GA and DE, we briefly discuss the initial population generation, search operators, control parameters, and constraint-handing techniques in the following few subsections.

A. Representation and Initial Population

The chromosome or representation of decision variables for both GA and DE can be expressed as follows:

$$x_j = [P_1^1, P_2^1 \ldots . P_{NT}^1, P_1^2, P_2^2 \ldots . P_{NT}^2, \ldots . . P_1^T, P_2^T \ldots . P_{NT}^T] \quad (11)$$

where $P_i^t = P_{GT_{i,t}}, \forall i, t$ and $j \in N_D$. N_D is the number of the population, and the number of decision variables for a DED problem is $N_x = T \times N_T$.

In general, an EA starts with a randomly generated population. As the DED problem has a bounded feasible region with many equality constraints, the individuals in the initial population are generated as per the following equation:

$$x_{i,j} = x_i^{\min} + (x_i^{\max} - x_i^{\min}) lhsdesign(1, N_x)$$
$$i \in N_x \; and \; j = 1 \ldots N_D \quad (12)$$

where $x^{\min}$ and $x^{\max}$ are the upper and lower bounds of each variable that can be found from each power plant's limits, and *lhsdesign* a MATLAB function used for Latin hypercube sampling [39].

B. Genetic Algorithm

GA has two main search operators known as crossover and mutation. Crossover is the process of exchanging chromosome material to create a new offspring, and mutation helps to diversify the population. In this research, we use simulated binary crossover and non-uniform mutation as both have shown better performance in comparison with other GA variants [40].

1) Simulated Binary Crossover (SBX): Among different crossover operators available, SBX has been widely used [37]. In it, child populations (y_i^1, y_i^2) are generated step by step such as

$$y_i^1 = 0.5[(1+\beta_{qi})x_i^1 + (1-\beta_{qi})x_i^2] \quad (13)$$
$$y_i^2 = 0.5[(1-\beta_{qi})x_i^1 + (1+\beta_{qi})x_i^2] \quad (14)$$

where β_{qi} is calculated as

$$\beta_{qi} = \begin{cases} (2u_i)^{1/\eta_c+1} & u_i \le 0.5, \\ \left(\frac{1}{2(1-u_i)}\right)^{1/\eta_c+1} & u_i > 0.5 \end{cases} \quad (15)$$

where u_i is a uniform random number in the range $[0, 1]$ and η_c a user-defined parameter distribution.

2) Non-Uniform Mutation: The main purpose of any mutation operator is to maintain diversity among the individuals in a population. In this research, we use non-uniform mutation that decreases the step size and increases the probability that the amount of mutation will decrease as the number of generation increases. This prevents the population from stagnating in the early stages of evolution and then allows the GA to fine-tune the solution in later stages. In it, a child is mutated according to

$$x'_{i,j}(g) = x_{i,j}(g) + \delta_{i,j}(g) \quad (16)$$

$$\delta_{i,j}(g) = \begin{cases} (x_i^{\max} - x_{i,j}(g))\left(1 - [u(g)]^{1-(t/N_G)^b}\right) & u \le 0.5, \\ (x_i^{\min} - x_{i,j}(g))\left(1 - [u(g)]^{1-(g/N_G)^b}\right) & u > 0.5 \end{cases}$$
$$i \in N_x \text{ and } j = 1 \ldots N_D \quad (17)$$

where $u(g)$ is a random number ($\in [0, 1]$), and g and N_G the current generation number and maximum number of generations, respectively. The speed of the step length can be controlled by choosing different "b" values and, in this research, is set to 5 [41].

C. Differential Evolution

DE has also two search operators known as mutation and crossover, but there operations are little different from GA operators. DE presents the solutions in a vector form. Once a new solution is generated from parents, it is called a candidate and new candidates for the next generation are selected, if, and only if, their fitness values are better than that of the corresponding parent's fitness. In the following subsections, we discuss the DE search operators used in this research.

1) Mutation and Crossover: Over the last two decades, many mutation operators have been proposed [42]. However, as it has been proven that there is no single one that can perform well for all types of test problems [43]. Therefore, in this research, two mutation operators are considered. Regarding crossover operators, it has been found that binomial is better than exponential crossover [42] while, in the selection operator, an offspring is selected if it is better than its parent.

Therefore, the first mutation operator we used, in this paper, facilitates the population diversity while the second one improves the convergence rate [44]. Consequently, the new offspring is generated as follows:

$$\vec{v}_{z,t} = \begin{cases} \vec{x}_{r_3,t} + F_z(\vec{x}_{r_1,t} - \vec{x}_{r_2,t}), \\ \quad if \; rand_1 \le Cr_z \text{ and } rand_2 \le prob_1, \\ \vec{x}_{i,t} + F_z((\vec{x}_{r_1,t} - \vec{x}_{r_2,t}) + (\vec{x}_{best,t} - \vec{x}_{i,t})), \\ \quad if \; rand_1 \le Cr_z \text{ and } rand_2 > prob_1, \\ \vec{x}_{i,j} \; otherwise. \end{cases} \quad (18)$$

In the mutation events, three parents are selected randomly from the entire population such that $z \ne r_1 \ne r_2 \ne r_3$. Here, F_z is the amplification factor for the mutation operator, Cr_z the crossover rate, and $prob_1$ a predefined probability (here, it is set to a value of 0.5) of choosing the methodology for generating new individuals from the current one. The selection process of F_z and Cr_z are self-adaptively calculated, as described in the next subsection.

2) Self-Adaptive Amplification Factor and Crossover Rate: It is a fact that DE's performance depends on its control parameters but selecting them is a combinatorial optimization problem. Therefore, a self-adaptive mechanism is deployed in this paper [44].

Initially, for each individual in the population, two sets of control parameters, $\dot{F} \in N(0.5, 0.1)$ and $\dot{C}r \in N(0.5, 0.1)$, are generated using normal distributions with mean and standard deviation values of 0.5 and 0.1, respectively. Then, to generate new offspring as per (18), F_z and Cr_z are calculated as follows:

$$F_z = \begin{cases} \dot{F}_{r_1} + rand_1(\dot{F}_{r_2} - \dot{F}_{r_3}), \text{ if } (rand_2 < \tau_1), \\ rand_3, \text{ otherwise} \end{cases} \tag{19}$$

$$Cr_z = \begin{cases} \dot{C}r_{r_1} + rand_4(\dot{C}r_{r_2} - \dot{C}r_{r_3}), \text{ if } (rand_5 < \tau_1), \\ rand_6, \text{ otherwise} \end{cases} \tag{20}$$

where $rand_\tau \in [0, 1]$ for $\tau = 1, 2, \ldots, 6$ and $\tau_1 = 0.75$ [44]. If values are less than 0.1 or larger than 1, they are truncated to 0.1 and 1, respectively.

Once the offspring is generated after evolving mutation and crossover, a greedy selection scheme is used as follows:

$$\vec{\mathbf{x}}_{i,g+1} = \begin{cases} \vec{\mathbf{v}}_{i,g+1}, \text{ if } f(\vec{\mathbf{v}}_{i,g+1}) < f(\vec{\mathbf{x}}_{i,g}), \\ \vec{\mathbf{x}}_{i,g}, \text{otherwise} \end{cases} \quad \forall i \in N_D. \tag{21}$$

If the offspring is better than the parents, it is accepted for the next generation. In addition, the parent's respective $\dot{F}_z$ and $\dot{C}r_z$ are replaced by the offspring's contributed F_z and Cr_z. This process is repeated until all the individuals are selected. Therefore, at the end of the current generation, the only better-performing individuals ($\vec{x}$) and their corresponding F_z and Cr_z are placed for the next generation evaluation.

D. Selection Process

In most EAs, a feasible solution is considered better than an infeasible one during the course of evolution [45]. Additionally, of any two infeasible solutions, the one with the minimum constraints violation (*CV*) is considered the best. Although this mechanism does not add additional parameters, there is a risk of losing some solutions which have good objective function values but marginally violate the constraints [45]. In this regard, an additional objective function based on the amount of *CV* for the population members is considered. Then, considering these two objectives (original and additional), the solutions are ranked using a non-dominated and crowding distance (*CD*) mechanism [45] by examining their fitness values, *CV* and *CD* neighboring solutions, i.e., the solutions with better fitness values, lower *CV*, and non-crowded are given higher ranks than those with better fitness values but in a crowded area.

E. Heuristic for DED Constraints

It has already been mentioned that DED is a nonlinear constrained optimization problem involving a number of equality and inequality constraints. The solutions generated by EAs may not satisfy all constraints, especially equality (demand balance) and dynamic (ramp limits) ones. Even if a feasible solution is obtained in one generation, it may become infeasible after crossover and mutation in another generation. This situation becomes even worse when many equality constraints are involved. A great deal of research has been undertaken into dealing with equality constraints, including penalty function integration [36], slack generation consideration [5], and local search consideration [35]. However, these approaches are not adequate for handling a chain of equality constraints, as is the case in DED problems. A few researchers have used SQP to deal with equality constraints and increase the convergence rate [35] but, although this approach returns a feasible solution after a long run, it loses significant diversity.

In this paper, we have proposed a heuristic to transform the infeasible solutions into feasible solutions. In the process, the 24-h load cycle is divided into 24-hourly sub-problems, and allocate production to meet the load demand in each hour starting from different random hours. Although the allocation can be started from the first hour of the operational cycle, as done in [36], the allocation can be infeasible at a later stage due to ramp constraint and any significant changes in demand (i.e., peak demand period). Note that the generation limit in any hour depends on the generation of the immediate past hour. The heuristic consists of the following steps.

Step 1) Arrange the decision variables (x) into a matrix form as

$$P = \begin{bmatrix} P_1^1 & P_2^1 & \ldots & P_{N_T}^1 \\ P_1^2 & P_2^2 .. & .. & P_{N_T}^2 \\ \vdots & \vdots & \vdots & \vdots \\ P_1^T & P_2^T .. & .. & P_{N_T}^T \end{bmatrix}. \tag{22}$$

Step 2) Randomly select an hour ($t \in T$) and generation ($P_t \in P$) at that hour, and start the forward process.

Step 3) Set $P_{i,t}^{\max} = P_i^{\max}$ and $P_{i,t}^{\min} = P_i^{\min}$.

Step 3.1) Check $P_{i,t}^{\min} \leq P_i^t \leq P_{i,t}^{\max}$ $\forall i$ and, if a unit is infeasible, freeze it using (26).

Step 3.2) Check feasibility at the tth hour as follows:

$$\left| \sum_{i=1}^{N_T} P_i^t - (P_{D_t} + P_{loss_t}) \right| \leq \varepsilon. \tag{23}$$

Here, ε is a tolerance limit which has a large value at the early stages of evolutionary process and is reduced to 1e-06 (an acceptable limit as of [8]) over the generations [46] such as

$$\varepsilon(g) = \begin{cases} \varepsilon(0)\left(1 - \frac{g}{N_{G_c}}\right) 0 < g < N_{G_c}, \\ 0, \quad g \geq N_{G_c} \end{cases}$$
$$\varepsilon(0) = \sigma * CV \tag{24}$$

where σ is a constant that determines the initial preserving $CV(\varepsilon(0))$, and g and N_{G_c} the current and cut-off generations, respectively. The cut-off generation indicates that there are no infeasible solutions in the population.

If the solution is feasible, go to step 3.4; otherwise, go to the next step.

Step 3.3) Obtain a random permutation of N_T and generate a random sequence of the operating units as $R_s = \{r_1, r_2, \ldots, r_{N_T}\}$ to satisfy the equality constraints. Now, choose the first generator $(r = r_1)$ as the slack generator to balance the residual load, as decided by the rest of the generator's known output, as

$$P_r^t = (P_{D_t} + P_{loss_t}) - \sum_{\substack{i=1 \\ i \neq r}}^{N_T} P_i^t \quad (25)$$

$$\begin{aligned} &if\ P_r^t < P_{r,t}^{\min}, \quad P_r^t = P_{r,t}^{\min} \\ &elseif\ P_r^t > P_{r,t}^{\max},\ P_r^t = P_{r,t}^{\max}\ end, \\ &r \in i \quad \forall i. \end{aligned} \quad (26)$$

Check feasibility using (23) and (26) and the look-ahead demand constraint which determines the $(t+1)$th hour generation range that must satisfy the $(t+1)$th hour load demand and can be mathematically formulated as

$$\sum_{i=1}^{N_T} P_{i,t+1}^{\min} \leq (P_{D_{t+1}} + P_{loss_{t+1}}) \leq \sum_{i=1}^{N_T} P_{i,t+1}^{\max} \quad (27)$$

where $i \in N_T,\ t \in T$ and

$$P_{i,t+1}^{\max} = \min[P_i^{\max}, \quad (P_{i,t}^t + UR_i)] \quad (28)$$

$$P_{i,t+1}^{\min} = \max[P_i^{\min}, \quad (P_{i,t}^t - DR_i)]. \quad (29)$$

If the solution of $\vec{P}^t$ is still infeasible, recalculate (25) considering the next random slack generator $(r = r_2)$ from the R_s vector. This process is repeated until a feasible solution which satisfies (23), (26), and (27) is found. Then, the new operating range at the $(t+1)$th hour is updated using (28) and (29), and set to $t = t+1$.

Step 3.4) Repeat steps 3.1 to 3.3 and obtain a feasible solution at the tth hour. As this process is repeated until $t = T$, the P matrix is updated from the t to T hours and the rest of the hours determined using the backward process which is applied to obtain feasible solutions at the $(t-1)$th to first hour as follows.

Step 4) Set $t = t-1$ and update the capacity range of $P_i^t, \forall i$ using

$$P_{i,t-1}^{\max} = \min[P_i^{\max}, \quad (P_{i,t}^t + UR_i)] \quad (30)$$

$$P_{i,t-1}^{\min} = \max[P_i^{\min}, \quad (P_{i,t}^t - DR_i)]. \quad (31)$$

Step 4.1) Calculate the feasible solution at the tth hour using the process described in steps 3.1 to 3.3 and then repeat steps 4 and 4.1 until $t = 1$.

Step 5) Reconstruct $\vec{x}$ from the calculated matrix $(P_i^t\ \forall i, t)$ using (11).

Step 6) Return a feasible $\vec{x}$ to the algorithm.

As this proposed heuristic does not place any priority on a unit or particular hour, it will help to maintain the diversity of solutions expected in EAs.

F. Diversity Mechanism

In fact, any EA can become stuck in local solutions, especially those for DED problems. To tackle this, if the average fitness function of the current population does not improve for a predefined number of generations, some individuals are randomly replaced as

$$\begin{aligned} &if |\min(f_g - f_{g-k})| < \zeta \\ &\quad x'_{i,j} = \begin{cases} x_{i,j}^{\min} + (x_{i,j}^{\max} - x_{i,j}^{\min}) rand\ if\ rand \leq \phi, \\ x_{i,j} \quad otherwise \end{cases} \\ &\quad \forall i = 1, 2, \ldots N_x, \forall_j = \frac{N_s}{2}, \frac{Ns}{2} + 1, \ldots . . N_s \\ &endif \end{aligned} \quad (32)$$

where f_g and f_{g-k} are the best fitness values at the gth and $(g-k)$th generations, respectively, k and ζ the tolerance factors, that is, we want to tolerate a kth (assume 100) number of generations by changing the fitness value within ζ (assume 0.001) and ϕ a constant (we set it to 20%) to represent the number of individuals to be randomly replaced.

G. Steps of the Proposed Algorithm

The proposed enhanced GA (E-GA) and DE (E-DE) are applied to the DED problem with VPE according to the following steps.

Step 1) Generate an initial population based on (11).

Step 2) Satisfy system constraints (3)–(10) using the heuristic described in Section III-E.

Step 3) Evaluate the fitness values using the formula in (1).

Step 4) Create child populations using the crossover and mutation operators described in Sections III-B and III-C for GA and DE, respectively.

Step 5) Select the best individuals from both the parent and child populations using the selection process described in Section III-D.

Step 6) Modify the infeasible individuals (if any) to satisfy the constraints using the proposed heuristic.

Step 7) If required, apply the diversity mechanism.

Step 8) If a stopping criterion is met, stop; otherwise, go to step 3.

IV. Experimental Results

For our experimental study, we take a few test problems from the literature that involve up to 150 thermal units for a 24-h planning horizon with a 1-h long time period. Based on the availability of data, these problems can be solved both with and without consideration of power-loss constraints. The problems we solve are briefly described in the following.

Case 1: a 5-unit problem without power loss [36];

Case 2: a 5-unit problem with power loss using the loss coefficients (B) [36];

Case 3: a 10-unit system without power loss [8];

Case 4: a 10-unit system with power loss [8];

Case 5: a 30-unit system generated by combining three 10-unit systems of Case 3 without power loss [8];

Case 6: a 100-unit system generated by combining ten 10-unit systems without power loss [8]; and

Case 7: a 150-unit thermal system without power loss [5].

For a fairer comparison, we select the same SRs as in [8], whereby the 1-h SR is set to 5% of the load and the 10-min one to $2/6 \times 5\%$ of the load. The relaxation factor of the equality constraints (ε) is set to 1e-6 and the cut-off generation (T_c) to

TABLE I
SUMMARY OF SOLUTIONS FOR 5-UNIT SYSTEM WITH LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
SA [14]	47356.00	NR	NR	NR
APSO [15]	44678.00	NR	NR	NR
GA [13]	44862.00	44922.00	45894.00	NR
PSO [13]	44253.00	45657.00	46403.00	NR
ABC [13]	44046.00	44065.00	44219.00	NR
AIS [16]	44385.00	44759.00	45554.00	NR
Hybrid PSO [36]	43223.00	43732.00	44252.00	274.95
E-GA	42528.90	42580.60	42638.40	30.16
E-DE	**42528.70**	**42571.20**	**42664.50**	**36.90**

TABLE II
SUMMARY OF SOLUTIONS FOR 10-UNIT SYSTEM WITH LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
EP [17]	1054685	1057323	NR	NR
EP-SQP [17]	1052668	1053771	NR	NR
MHEP-SQP [34]	1050054	1052349	NR	NR
DGPSO [22]	1049167	1051725	NR	NR
IPSO [18]	1046275	1048154	NR	NR
AIS [16]	1045715	1047050	1048431	NR
ECE [30]	1043989	1044963	1046805	NR
ABC [13]	1043381	1044963	1046805	NR
TVACIPSO [19]	1041066	1042118	1043625	NR
EBSO [23]	1038915	1039188	1039272	NR
CSAPSO [21]	1038251	1039543	NR	NR
SAMFA [28]	1037698	1037938	1039199	NR
MTLA [29]	1037489	1037712	1038090	NR
MIQP [8]	1038376	NR	NR	NR
E-GA	1036460	1037020	1037430	251.83
E-DE	**1036280**	**1036310**	**1036380**	**51.31**

200. The GA parameters, the probability of crossover, distribution index (η), and probability of mutation are set to 0.9, 3, and 0.1, respectively. We set the population sizes to 100 for the 5-, 10-, and 30-unit, and 200 for the 100- and 150-unit problems, and the maximum number of generations to 4000 for all cases. Thirty independent runs are performed for each test case and the solutions recorded and compared with the results from the-state-of-the-art algorithms.

The algorithms are implemented on a desktop personal computer with a 3.4-GHz Intel Core i7 processor with 16 GB of RAM using the MATLAB (R2012b) environment. The algorithm runs until the number of generations is higher than 4000 (criterion 1) or the best fitness value is no longer improved in 100 generations (criterion 2) or the average fitness value is no longer improved in 100 generations (criterion 3).

A. DED With TL

As power TL cannot be avoided in a power distribution system, it is important to consider it when scheduling generating units. In this paper, due to data unavailability, the TL is considered for only two cases (1 and 3). Their results are compared with those from E-GA and E-DE as well as state-of-the-art algorithms. The solutions obtained for the 5-unit and 10-unit problems with SR constraints using the proposed algorithms (E-DE and E-GA) are presented in Tables I and II, respectively, along with the results from the state-of-the-art algorithms. It can be seen that the proposed algorithms are able to obtain much better results than the state-of-the-art algorithms. Additionally, E-DE is found to be better than E-GA.

TABLE III
SUMMARY OF SOLUTIONS FOR 5-UNIT SYSTEM WITHOUT LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
E-GA	42524.40	42565.90	42630.80	26.77
E-DE	**42523.60**	**42524.80**	**42621.60**	**28.87**

TABLE IV
SUMMARY OF SOLUTIONS FOR 10-UNIT SYSTEM WITHOUT LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
EP [17]	1048638	NR	NR	NR
SQP [17]	1051163	NR	NR	NR
EP-SQP [17]	1031746	1035748	NR	NR
MHEP-SQP [34]	1028924	1031179	NR	NR
AIS [16]	1021980	1023156	1024973	NR
GA [13]	1033481	1038014	1042606	NR
ABC [13]	1021576	1022686	1024316	NR
DE [26]	1036756	1040586	1452558	3225.8
CDE [26]	1019123	1020870	1023115	1310.7
MDE [25]	1031612	1033630	NR	NR
CSDE [27]	1023432	1026475	1027634	NR
Hybrid DE [25]	1031077	NR	NR	NR
HS [31]	1046726	NR	NR	NR
HHS [31]	1019091	NR	NR	NR
CE [30]	1022702	1024024	NR	NR
ECE [30]	1022272	1023335	NR	NR
PSO [13]	1027679	1031716	1034340	NR
IPSO [18]	1023807	1026863	NR	NR
ICPSO [20]	1019072	1020027	NR	NR
PSO-SQP [7]	1027334	1028546	NR	NR
ICA [32]	1018468	1019291	1021796	NR
Hybrid PSO [36]	1018159	1019850	1021813	826.94
MIQP [8]	1016601	NR	NR	NR
E-GA	1016360	1016710	1016880	221.11
E-DE	**1016160**	**1016260**	**1016420**	**69.93**

TABLE V
SUMMARY OF SOLUTIONS FOR 30-UNIT SYSTEM WITHOUT LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
EP [34]	3164531	3200171	NR	NR
EP-SQP [34]	3159024	3169093	NR	NR
MHEP-SQP [34]	3151445	3157438	NR	NR
DE [26]	3162997	3173102	NR	NR
CDE [26]	3083930	3090542	NR	NR
CE [30]	3086110	3088870	NR	NR
ECE [30]	3084649	3087847	NR	NR
IPSO [18]	3090570	3096900	NR	NR
ICPSO [20]	3064497	3071588	NR	NR
Hybrid PSO [36]	3062144	3067277	NR	2177.6
MIQP [8]	3049359	NR	NR	NR
E-GA	3049110	3049550	3051150	879.62
E-DE	**3046110**	**3046640**	**3046970**	**227.87**

B. DED Without TL

We solve the 5-, 10-, 30-, 100-, and 150-unit DED problems without TLs and the results obtained from our approaches, along with those from some others in the literature, are presented in Table III–VII in which it is clear that our algorithms outperform all the others.

TABLE VI
SUMMARY OF SOLUTIONS FOR 100-UNIT SYSTEM WITHOUT LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
GA [28]	10908741	11584628	11987675	NR
PSO [28]	10366076	10766385	11310279	NR
FA [28]	10197269	10419457	11216243	NR
SAFA [28]	10183819	10286043	10388958	NR
SAMFA [28]	10170104	10171876	10179061	NR
MIQP [28]	10170508	NR	NR	NR
E-GA	10170343	10174764	10180669	3978.57
E-DE	**10158600**	**10165800**	**10168300**	**3362.58**

TABLE VII
SUMMARY OF SOLUTIONS FOR 150-UNIT SYSTEM WITHOUT LOSS

Method	Production cost ($)			
	Minimum	Average	Maximum	Std. dev.
BA [5]	15287005	15291497	15296855	NR
SALBA [5]	15256663	15258781	15260355	NR
E-GA	15260000	15266300	15267100	2529.31
E-DE	**15247900**	**15259800**	**15260100**	**1117.66**

TABLE VIII
SUMMARY OF COMPUTATIONAL COST FOR DIFFERENT PROBLEMS

Problem size	E-GA		E-DE	
	Maximum generation	CPU time (min)	Maximum generation	CPU time (min)
5 units	665	4.7	635	2.8
10 units	1984	12.8	2652	11.7
30 units	1327	39.2	3126	83.2
100 units	2736	118.23	2354	112.43
150 units	4000	187.21	4000	157.03

The computational costs of different approaches for different problems are presented in Table VIII, in which it can be seen that E-DE is better than E-GA for all problems because it provides better quality solutions and requires less computational time. Note that, for the large 150-unit DED problem, both GA and DE reach the prescribed maximum number of generations, which indicates that their fitness values improve very slowly and means that the solutions could be further improved at a higher computational cost.

V. ANALYSIS OF DIFFERENT COMPONENTS

In this section, we have analyzed the effect of different components of the algorithms such as heuristic, mutation, self-adaptation, and selection process for our algorithms. For the analysis, we have mainly considered a 10-unit (case-3) problem as a representative case.

A. Effect of Heuristic

The proposed heuristic transforms the infeasible solutions into good quality feasible solutions. To demonstrate the effect of the proposed heuristic, the average of best objective function values over 30 independent runs of each variant are recorded in Table IX. From this table, it can be seen that GA without the heuristic does not find a single feasible solution, even after 4000 generations, and although DE is able to find a few feasible solutions, the quality is poor. When the heuristic is applied starting from the first hour (HFS), both algorithms are able to obtain feasible solutions and the solutions are better than the same for without heuristic if any feasible solution is obtained. However, with the proposed heuristic that is with a random starting hour (HRS), the solutions are better than those obtained with HFS.

TABLE IX
AVERAGE OBJECTIVE VALUES OBTAINED BY GA AND DE WITH AND WITHOUT HEURISTIC

Algorithm	Type of heuristic	Pop. size × Max. gen.	Objective value
GA	None	100 × 4000	Infeasible
	HFS	100 × 4000	1018190
	HRS	100 × 4000	**1016360**
DE	None	100 × 4000	1076540
	HFS	100 × 4000	1017370
	HRS	100 × 4000	**1016160**

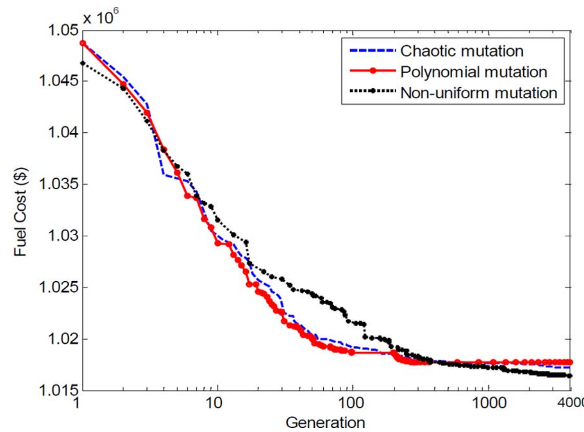

Fig. 1. Effect of non-uniform mutation in GA for solving DED problem.

B. Effect of Mutation

In this subsection, we have analyzed the performance of non-uniform, chaotic, and polynomial mutation operators with GA. A sample of 4000 generations with identical parameters is set for all variants, and the convergence plots for the best fitness values of each variant are shown in Fig. 1. Although the non-uniform mutation variant has a slow convergence rate in early generations, it is able to obtain better results at the end of the evolutionary process, i.e., GA with a non-uniform mutation operator obtains an objective function value of $1 016 360 while, with polynomial and chaotic mutation operators obtain $1 017 710 and $1 017 170, respectively. In conclusion, non-uniform mutation is dominant over the other two mutation operators for solving DED problems.

C. Effect of Self-Adaptation

To demonstrate that the self-adaptation strategy used in this paper provides better results than that with fixed-parameters, a DE algorithm with two different mutation and crossover rates is considered, and its results compared with those obtained from the algorithm with self-adaptive mechanism. The average values, of best objective solutions, obtained from 30 runs for each variant are shown in Table X, in which it is clear that using the self-adaptive method to calculate DE parameters is beneficial in terms of the objective values obtained.

TABLE X
COMPARISON OF AVERAGE FITNESS VALUES FOR DIFFERENT TYPES OF MUTATION AND CROSSOVER RATES

Type of F_z and Cr_z	Parameter values	Objective value
Fixed	$F_z = 0.9, Cr_z = 0.1$	1016380
	$F_z = 0.5, Cr_z = 0.5$	1016570
Adaptive	$\dot{F}_z = 0.5 \sim 0.95$, $\dot{C}r_z = 0.5 \sim 0.95$	**1016160**

TABLE XI
AVERAGE FUNCTION VALUES OBTAINED BY GA AND DE WITH DIFFERENT SELECTION TECHNIQUES

Problem	EAs	Selection approach	
		Traditional	Proposed
10 units	E-DE	1016460	**1016160**
	E-GA	1017810	**1016360**
30 units	E-DE	3048460	**3046110**
	E-GA	3053931	**3049110**

D. Effect of Selection Process

In this section, we have shown the superiority of proposed non-dominated-based selection process over the traditional technique. The experiments were conducted with 10- and 30-unit DEDs (without loss) up to 4000 maximum generations with a population size of 100. The results are presented in Table XI, in which it is clear that the proposed selection mechanism outperforms the traditional one for both GA and DE.

VI. SCHEDULING UNDER ROLLING HORIZON FRAMEWORK

In this section, we have demonstrated the use of our proposed algorithm for scheduling on a rolling horizon basis [47]. In many real-life situations, the scheduling has to be updated with the change of data and availability of new information. This is also applicable to DED scheduling, where the demand and operational data may change for future periods within the planning horizon. In this paper, we have considered a planning horizon of 24 h, where each period is 1 h long. Such a planning horizon will cover a full-cycle of power demand, which is also long enough to generate feasible scheduling under ramp constraints. However, the length of planning horizon can be shortened, if needed, without modifying the algorithm. The length of each period can also be reduced for dealing with sensitive data (frequent changes) effectively or for real-time scheduling. The algorithm can also be applied for reactive rescheduling, where the schedule is revised due to any changes that take place in any input at any point in time. In the process, at the beginning, the schedule is generated for 24 h, that is for 1 to 24 h with an intention that only the production plan for period one will be implemented. At the end of period 1, the schedule is generated for 2 to 25, considering any changes/updates in data or input that may be experienced in period 1. At the end of period 2, the schedule is generated for 2 to 26, and so on. Such a scheduling process will provide a better operational plan.

To demonstrate the application of the rolling horizon process, we have considered a 5-unit test problem, and generated random demands with 5% standard deviation from the forecasted demand. The demand data are shown in Fig. 2. The generating units are considered as committed for all runs. Based on the new data, the second and subsequent runs are performed by our proposed methodology. The simulation has run 30 times and the best fitness values are reported in Table XII. From the table, it is seen that the E-DE has provided better results comparing to other algorithms for each run. From this experience, we say that te DED model can be dynamically implemented in practice using rolling horizon framework as the procedure always uses the most recent information, while the data are updated at every single period. Hence, the proposed approach in conjunction with rolling horizon framework will enhance practice of DED problem solving.

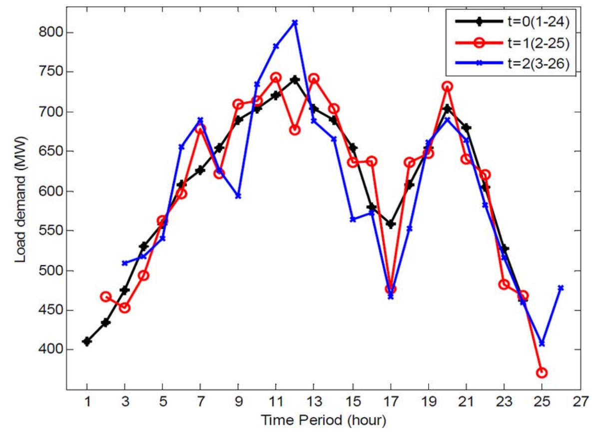

Fig. 2. Load demands for different scenarios.

TABLE XII
5-UNIT TEST RESULTS OBTAINED FROM THE ROLLING-HORIZON FRAMEWORK

Run	Schedule period (hour)	Fuel cost ($)			
		GA	DE	E-GA	E-DE
at $t = 0$	1-24	55800.70	46691.60	42539.40	**41943.10**
at $t = 0$	2-25	51281.00	46817.30	43835.10	**43289.10**
at $t = 0$	3-26	50165.60	46565.90	43775.50	**43169.20**

VII. CONCLUSION AND FUTURE WORK

In this paper, we demonstrated that a real-parameter enhanced GA with a non-uniform mutation and a self-adaptive enhanced DE exhibited superior performances in solving DED problems. In this approach, a random sequential technique was used to consider periodic simpler sub-problems in order to satisfy the equality constraints and dynamic ramp constraints. A dynamic relaxation factor for the equality constraints was set to preserve a few marginally infeasible solutions in order to enhance the convergence rate. In addition, a selection mechanism for preserving the best and non-crowded individuals in order to avoid a local solution was discussed. A parametric analysis explicitly showed the effect of different components used in this paper. Applications of both the E-GA and E-DE algorithms in a number of test problems taken from recent literature, which included TLs, revealed their remarkably better performances.

In future work, we intend to apply this approach using other population-based algorithms to solve other DED problems including renewable sources and emission effects.

References

[1] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York, NY, USA: Wiley, 2012.

[2] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1325–1332, Aug. 1993.

[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] X. Xia and A. M. Elaiw, "Optimal dynamic economic dispatch of generation: A review," *Electr. Power Syst. Res.*, vol. 80, no. 8, pp. 975–986, 2010.

[5] T. Niknam, R. Azizipanah-Abarghooee, M. Zare, and B. Bahmani-Firouzi, "Reserve constrained dynamic environmental/economic dispatch: A new multiobjective self-adaptive learning bat algorithm," *IEEE Syst. J.*, vol. 7, no. 4, pp. 763–776, Dec. 2013.

[6] M. Basu, "Modified particle swarm optimization for nonconvex economic dispatch problems," *Int. J. Electr. Power Energy Syst.*, vol. 69, no. 0, pp. 304–312, 2015.

[7] T. A. Victoire and A. E. Jeyakumar, "Reserve constrained dynamic dispatch of units with valve-point effects," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1273–1282, Aug. 2005.

[8] M. Q. Wang, H. B. Gooi, S. X. Chen, and S. Lu, "A mixed integer quadratic programming for dynamic economic dispatch with valve point effect," *IEEE Trans. Power Syst.*, vol. 99, pp. 1–10, 2014.

[9] C. B. Somuah and N. Khunaizi, "Application of linear programming redispatch technique to dynamic generation allocation," *IEEE Trans. Power Syst.*, vol. 5, no. 1, pp. 20–26, Feb. 1990.

[10] C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," *IEEE Trans. Power App. Syst.*, vol. PAS-103, no. 6, pp. 1170–1175, 1984.

[11] G. Irisarri, L. M. Kimball, K. A. Clements, A. Bagchi, and P. W. Davis, "Economic dispatch with network and ramping constraints via interior point methods," *IEEE Trans. Power Syst.*, vol. 13, no. 1, pp. 236–242, Feb. 1998.

[12] G. Binetti, A. Davoudi, D. Naso, B. Turchiano, and F. L. Lewis, "A distributed auction-based algorithm for the nonconvex economic dispatch problem," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1124–1132, 2014.

[13] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial bee colony algorithm for units with valve-point effect," *Eur. Trans. Electr. Power*, vol. 21, no. 1, pp. 70–81, 2011.

[14] C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, and M. Basu, "Simulated annealing technique for dynamic economic dispatch," *Electr. Power Compon. Syst.*, vol. 34, no. 5, pp. 577–586, 2006.

[15] B. K. Panigrahi, V. R. Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Convers. Manage.*, vol. 49, no. 6, pp. 1407–1415, 2008.

[16] S. Hemamalini and S. P. Simon, "Dynamic economic dispatch using artificial immune system for units with valve-point effect," *Int. J. Electr. Power Energy Syst.*, vol. 33, no. 4, pp. 868–874, 2011.

[17] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function," *IEEE Trans. Power Syst.*, vol. 22, no. 4, p. 77, Nov. 2002.

[18] 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.

[19] B. Mohammadi-ivatloo, A. Rabiee, and M. Ehsan, "Time-varying acceleration coefficients IPSO for solving dynamic economic dispatch with non-smooth cost function," *Energy Convers. Manage.*, vol. 56, no. 0, pp. 175–183, 2012.

[20] 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, 2010.

[21] 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, 2011.

[22] T. A. Victoire and A. E. Jeyakumar, "Deterministically guided PSO for dynamic dispatch considering valve-point effect," *Electr. Power Syst. Res.*, vol. 73, no. 3, pp. 313–322, 2005.

[23] T. Niknam and F. Golestaneh, "Enhanced bee swarm optimization algorithm for dynamic economic dispatch," *IEEE Syst. J.*, vol. 7, no. 4, pp. 754–762, 2013.

[24] S. S. Balamurugan R., "An improved differential evolution based dynamic economic dispatch with nonsmooth fuel cost function," *J. Electr. Syst.*, vol. 01, 2007.

[25] 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, 2008.

[26] 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.

[27] 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, 2011.

[28] T. Niknam, R. Azizipanah-Abarghooee, and A. Roosta, "Reserve constrained dynamic economic dispatch: A new fast self-adaptive modified firefly algorithm," *IEEE Syst. J.*, vol. 6, no. 4, pp. 635–646, 2012.

[29] T. Niknam, R. Azizipanah-Abarghooee, and J. Aghaei, "A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 749–763, May 2013.

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

[31] V. Ravikumar Pandi and B. K. Panigrahi, "Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm," *Expert Syst. Applicat.*, vol. 38, no. 7, pp. 8509–8514, 2011.

[32] B. Mohammadi-ivatloo, A. Rabiee, A. Soroudi, and M. Ehsan, "Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch," *Energy*, vol. 44, no. 1, pp. 228–240, 2012.

[33] M. Basu, "Hybridization of bee colony optimization and sequential quadratic programming for dynamic economic dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 44, no. 1, pp. 591–596, 2013.

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

[35] A. M. Elaiw, X. Xia, and A. M. Shehata, "Hybrid DE-SQP and hybrid PSO-SQP methods for solving dynamic economic emission dispatch problem with valve-point effects," *Electr. Power Syst. Res.*, vol. 103, no. 0, pp. 192–200, 2013.

[36] Y. Zhang, D. wei Gong, N. Geng, and X. yan Sun, "Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects," *Appl. Soft Comput.*, vol. 18, no. 0, pp. 248–260, 2014.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Computat.*, vol. 6, no. 2, pp. 182–197, 2002.

[38] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimiz.*, vol. 11, no. 4, pp. 341–359, 1997.

[39] K. Q. Ye, "Orthogonal column Latin hypercubes and their application in computer experiments," *J. Amer. Statist. Assoc.*, vol. 93, no. 444, pp. 1430–1439, Dec. 1998.

[40] S. Elsayed, R. Sarker, and D. Essam, "A comparative study of different variants of genetic algorithms for constrained optimization," in *Simulated Evolution and Learning*. Berlin, Germany: Springer, 2010, pp. 177–186, Lecture Notes in Computer Science.

[41] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Adaptive configuration of evolutionary algorithms for constrained optimization," *Appl. Math. Computat.*, vol. 222, pp. 680–711, 2013.

[42] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Computat.*, vol. 15, no. 1, pp. 4–31, 2011.

[43] R. Sarker, S. Elsayed, and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," *IEEE Trans. Evol. Computat.*, to be published.

[44] S. Elsayed, R. Sarker, D. Essam, and N. M. Hamza, "Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization," in *CEC2014*, China, 2014.

[45] T. Ray, H. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," in *Constraint-Handling in Evolutionary Optimization*, E. Mezura-Montes, Ed. Berlin, Germany: Springer, 2009, pp. 145–165, Studies in Computational Intelligence.

[46] T. Takahama and S. Sakai, "Efficient constrained optimization by the constrained rank-based differential evolution," in *Proc. IEEE Congr. Evol. Computation (CEC)*, 2012, pp. 1–8.

[47] A. Lorca and X. A. Sun, "Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind," *IEEE Trans. Power Syst.*, to be published.

M. F. Zaman (S'15) received the B.Sc. and M.Sc. Eng. degrees from the Department of Electrical and Electronic Engineering at Rajshahi University of Engineering and Technology, Bangladesh, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree in the School of Engineering and IT, University of New South Wales, Canberra, Australia.

He served as a Lecturer and an Assistant Professor at Rajshahi University of Engineering and Technology. His research interests include economic operation of power system generators, renewable energy integration, optimization algorithms, evolutionary computation, and multi-objective optimization.

Saber M. Elsayed (M'10) received the Bachelor's degree in information systems and technology from Zagazig University, Egypt, in 2004, the Master's degree in computer science from Menoufiya University, Egypt, in 2008, and the Ph.D. degree in computer science from the University of New South Wales, Canberra, Australia, in 2012.

Currently, he is a research associate with the School of Engineering and Information Technology, University of New South Wales at Canberra. From 2005 to 2009, he was a teaching assistant with the Department of Operations Research and Decision Support, the Faculty of Computers and Informatics, Zagazig University, Egypt. His research interests include the areas of evolutionary algorithms, constrained and unconstrained optimization, multi-objective optimization, portfolio optimization, big data, and cyber-security using computational intelligence.

Dr. Elsayed was the winner of the "IEEE-CEC2011's Competition Testing Evolutionary Algorithms on Real-world Numerical Optimization Problems". He is a member of the IEEE Computer Society. He was also a member of the program committee of several international conferences. He is a reviewer in several international journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Tapabrata Ray (M'12) received the Ph.D. degree from the Indian Institute of Technology (IIT) Kharagpur.

He is an ARC Future Fellow with the School of Engineering and Information Technology, University of New South Wales (UNSW), Canberra, Australia. He is the founder and leader of the Multidisciplinary Design Optimization Research Group at UNSW, Canberra. He has published extensively in the area of evolutionary computation and design optimization. His research areas include multidisciplinary design optimization, evolutionary computation, multiobjective optimization, constrained optimization, robust design optimization, surrogate assisted optimization, shape representation and optimization, and bioinspired models for optimization.

Ruhul A. Sarker (M'03) received the Ph.D. degree from Dalhousie University, Halifax, NS, Canada, in 1992.

He is currently an Associate Professor in the School of Engineering and IT, University of New South Wales, Canberra, Australia. His main research interests are evolutionary optimization and applied operations research. He is the lead author of the book *Optimization Modelling: A Practical Approach* (Boca Raton, FL, USA: CRC, 2007). He has published more than 230 refereed articles in the international journals, edited books, and conference proceedings.

Prof. Sarker is currently an associate editor of *Memetic Computing Journal* and *Flexible Service and Manufacturing Journal*. He is a member of INFORMS.