Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

Website: http://AIMsciences.org
pp. **X–XX**

A MODIFIED DIFFERENTIAL EVOLUTION BASED SOLUTION TECHNIQUE FOR ECONOMIC DISPATCH PROBLEMS

Md. Abul Kalam Azad and Edite M.G.P. Fernandes

Algoritmi R&D Centre
School of Engineering, University of Minho
4710-057 Braga, Portugal

(Communicated by the associate editor name)

Abstract. Economic dispatch (ED) plays one of the major roles in power generation systems. The objective of economic dispatch problem is to find the optimal combination of power dispatches from different power generating units in a given time period to minimize the total generation cost while satisfying the specified constraints. Due to valve-point loading effects the objective function becomes nondifferentiable and has many local minima in the solution space. Traditional methods may fail to reach the global solution of ED problems. Most of the existing stochastic methods try to make the solution feasible or penalize an infeasible solution with penalty function method. However, to find the appropriate penalty parameter is not an easy task. Differential evolution is a population-based heuristic approach that has been shown to be very efficient to solve global optimization problems with simple bounds. In this paper, we propose a modified differential evolution based solution technique along with a tournament selection that makes pair-wise comparison among feasible and infeasible solutions based on the degree of constraint violation for economic dispatch problems. We reformulate the nonsmooth objective function to a smooth one and add nonlinear inequality constraints to original ED problems. We consider five ED problems and compare the obtained results with existing standard deterministic NLP solvers as well as with other stochastic techniques available in literature.

1. **Introduction.** In real world, as competition increases in the power generation industry, generating companies try to further improve the operating efficiency of their power plants. The application of mathematical optimization techniques has a long history in power generation systems and tangible improvements can still be achieved through the application of more robust solution techniques. Economic dispatch (ED) is one of the major important optimization task in power generation systems. The objective of economic dispatch problem is to find the optimal combination of power dispatches from different power generating units in a given time period to minimize the total generation cost while satisfying the specified load demands and the generating units operating conditions. If the ramp-rate constraints are included in ED, then it is called a dynamic economic dispatch (DED) problem. Generally, the cost function for each generating unit can be represented by a quadratic function and can be solved by traditional gradient-based methods,

2000 *Mathematics Subject Classification.* 90C30, 90C56, 90C59, 90C90.
Key words and phrases. Economic dispatch, Valve-point loading effects, Generator ramp-rate, Differential evolution, Constraint violation, Tournament selection.

such as Lagrangian multipliers. However, in the large power plants, the turbines have steam admission valves and the resulting cost function has additional non-differentiable terms. The inclusion of valve-point loading effects also increases the nonlinearity as well as the number of local minima in the solution space. So it is highly desirable to use derivative-free methods that can cope with a multimodal objective function to converge to a global minimum of the ED problem.

Many traditional solution methods exist to solve the ED problem such as linear programming [36], nonlinear programming [12, 56], quadratic programming [31] and Lagrangian relaxation algorithm [33, 39]. These methods may face some problems when converging to the solution due to nonlinearity and nondifferentiability of the objective cost function. Although one can transform the nonsmooth objective function into a smooth form that can be handled by these methods, the obtained solutions may still be local optima instead of the global one, since convergence is guaranteed only to a local optimum.

In the last decades many stochastic solution methods have been proposed to solve the ED problem due to their ability to converge in probability to the global optima. The stochastic methods are genetic algorithm [13, 55, 61], simulated annealing [48, 63], genetic algorithm/simulated annealing [62], particle swarm optimization [27, 49], evolutionary programming [50, 53, 64], hybrid stochastic search [16], augmented Lagrange Hopfield network [21]. Attaviriyanupap et al. [2] proposed a new methodology for solving the ED problem using evolutionary programming combined with sequential quadratic programming (EP-SQP). Victorie and Jeyakumar [59] proposed a hybrid technique of particle swarm optimization and sequential quadratic programming (PSO-SQP) to solve the ED problem. These methods appear to be efficient in solving the ED problem although they require derivative information to approximate locally the objective function by a quadratic model and the constraints by linear models.

The nonsmooth objective function of the ED problem can be reformulated to a smooth one by adding appropriate nonlinear inequality constraints to the original ED problem. Therefore the existing gradient-based standard deterministic NLP solvers can also be applicable to solve the ED problem. The task of global optimization is to find a solution where the objective function obtains its smallest value, the global minimum. When the objective function has a huge number of local minima, local optimization techniques are likely to get stuck before the global minimum is reached, and some kind of global search is needed to find the global minimum with some reliability. Algorithms for solving global minimization problem can be classified into deterministic methods that guarantee to find a global optimum with a required accuracy and stochastic methods that find the global minimum only with high probability. Usually deterministic methods are used to solve small dimensionality problems. In deterministic methods, the values of the objective function are assumed to be exact, and the computation is completely determined by the values sampled so far.

Nowadays, there are some deterministic solvers which give global solutions. ALGENCAN [8] is a novel global solver based on an augmented Lagrangian framework and the αBB method. BARON [57] stands for Branch And Reduce Optimization Navigator and is a computational system for solving nonconvex optimization problems to global optimality. Couenne [6] stands for Convex Over and Under ENvelopes for Nonlinear Estimation and is a branch and bound algorithm to solve mixed-integer nonlinear programming problems globally. It solves NLP subproblems by using IPOPT [60]. LINDOGlobal [43] is a global solver that employs branch-and-cut methods to break a nonlinearly constrained optimization problem model down into a list of subproblems. MCS stands for Multilevel Coordinate Search [35] and DIRECT stands for DIviding RECTangles [24] are optimization algorithms designed to search for global minima of a real valued objective function.

Available solvers that are able to guarantee convergence only to local solutions are: CONOPT [23] is a feasible path method based on the Generalized Reduced Gradient (GRG) algorithm that is well suited for models with very nonlinear constraints. FilterSQP [25] implements a Sequential Quadratic Programming trust-region algorithm which is suitable for solving large nonlinearly constrained problems. It uses the new concept of "filter" instead of a penalty merit function. IPOPT is an implementation of a primal-dual barrier algorithm with a filter line-search method for nonlinear programming. KNITRO [11] is designed for solving large-scale, smooth nonlinear programming problems. It offers both interior-point and active-set methods. LANCELOT [15] implements an augmented Lagrangian algorithm for nonlinearly constrained optimization problems. It is suitable for large nonlinearly constrained problems. LOQO [58] is based on an infeasible primal-dual interior-point method and solves both convex and nonconvex optimization problems. MINOS [47] is suitable for large constrained problems with a linear or nonlinear objective function and a mixture of linear and nonlinear constraints. SNOPT [29] implements a sequential programming algorithm that uses a smooth augmented Lagrangian merit function and makes explicit provision for infeasibility in the original problem and in the quadratic programming subproblems.

For moderate and large-scale problems, stochastic methods are widely used. Stochastic methods involve function evaluations at a suitably chosen random sample of solutions and subsequent manipulation of the sample to find good local (and hopefully global) minima. Even when the data is exact, it is sometimes beneficial to deliberately introduce randomness into the search process as a mean of speeding convergence and making the algorithm less sensitive to modeling errors. The stochastic methods can be classified as point-to-point search technique, such as Simulated Annealing [41], and population-based search technique such as Genetic Algorithm [34], Ant Colony [22], Differential Evolution [54], Tabu Search [30], Harmony Search [28], Particle Swarm Optimization [40], Evolution Strategy [5], Electromagnetism-like Mechanism [7], Artificial Bee Colony [38].

Differential evolution (DE) is a population-based heuristic approach proposed by Storn and Price [54], that is very efficient to solve nondifferentiable global optimization problems with simple bounds. The DE algorithm has been applied to solve the ED problem. He et al. [32] proposed a hybrid genetic algorithm (HGA) for solving the ED problem with valve-point loading effects that combines the GA algorithm with DE and SQP to improve the performance of HGA. Chiou [14] proposed a variable scaling hybrid differential evolution (VSHDE) for solving the ED problem in large-scale systems. The authors considered the smooth cost function. Balamurugan and Subramanian proposed a differential evolution [3] and an improved differential evolution (IDE) [4] for solving the ED problem with valve-point loading effects.

Since the ED problem can be formulated as general constrained nonlinear programming problem, it is very important to handle both equality and inequality constraints to make the obtained solution feasible. In solution methods for solving the ED problem found in literature, real power outputs are balanced to satisfy equality constraints in some way. In some methods, creating solution is continued until the feasible solution is obtained and hence the method is more time consuming. In the DED problem the ramp-rate constraints are modified to new operating limits of the generating unit. In some cases, penalty function method is used to penalize infeasible solutions. Although the penalty function method is applicable to any type of constraints, its performance is not always satisfactory. Since penalty parameter is always problem dependent, the difficulty of the penalty function method is to find the appropriate penalty parameter which guide the solution method towards the optimum.

Although the ED problems can be solved by standard NLP solvers, they do not always give global optimal solutions because of getting stuck in local optima. So global search techniques, in particular, stochastic techniques are highly desirable to solve ED problems so that better optimal solutions (hopefully global) are obtained. The original version of DE has only three parameters and according to Storn and Price [54] is sensitive to those parameters. It is very important to explore the whole search space as well as to exploit neighborhood of the best solution to improve the quality of the obtained solution in DE. In this paper, we propose a modified differential evolution algorithm for economic dispatch problems with valve-point loading effects and transmission loss considering the self-adaptive technique for control parameters, a modified mutation and a modified selection in order to obtain global solutions. In modified selection, to handle the constraints effectively, a tournament selection based on the feasibility and dominance rules that makes a pair-wise comparison among feasible and infeasible solutions is used. We test the proposed algorithm with various ED problems and compare obtained results with standard deterministic NLP solvers as well as other stochastic techniques available in literature.

The organization of this paper is as follows. We describe the formulation of the economic dispatch problems in Section 2. Section 3 describes the outline of original DE. We describe the proposed modified differential evolution in Section 4. Section 5 outlines the procedures to solve the ED problems with modified differential evolution. Section 6 describes the experimental results and finally we draw the conclusions of this study in Section 7.

2. **Economic Dispatch Problems.** Consider the ED problem with N generating units that generate power for T time periods. The objective function can be represented by the following cost function

$$\min\ f(\mathbf{P}) \equiv \sum_{i=1}^{N}\sum_{t=1}^{T} C_{it}(P_{it}),$$

where P_{it} is the ith unit power output at any time t, $C_{it}(P_{it})$ is the ith unit cost at any time t, and $\mathbf{P}$ is an array of the $N \times T$ components P_{it}.

The objective function of each unit with a quadratic fuel cost function can be expressed as

$$C_{it}(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i,$$

where a_i, b_i and c_i are the cost coefficients of unit i. The above objective function is continuously differentiable.

The standard ED problem minimizes the total fuel cost associated with N generating units in T time periods subject to the following constraints.

1) *Real power balance constraints*: The total power output from N units should be equal to the demand and loss at time t, i.e.,

$$\sum_{i=1}^{N} P_{it} = D_t + L_t, \quad t = 1, 2, \ldots, T \tag{1}$$

where D_t is the total assumed load demand at time period t. L_t is the transmission loss at time period t and it can be calculated by the following equation

$$L_t = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{it} B_{ij} P_{jt}, \tag{2}$$

where $\mathbf{B}$ is the $N \times N$ loss coefficients matrix.

2) *Generating unit ramp rate limits*: Depending on the load demand at time period t, the power output from each unit i can be increased or decreased, so called generating unit ramp-up and ramp-down, respectively, and should be satisfied by the following inequality constraints

$$\begin{aligned} & P_{it} - P_{i(t-1)} \le UR_i \\ & P_{i(t-1)} - P_{it} \le DR_i \\ & i = 1, 2, \ldots, N, \quad t = 2, 3, \ldots, T, \end{aligned} \tag{3}$$

where UR_i and DR_i are the ramp-up and ramp-down limits of the ith unit, respectively.

3) *Real power operating limits*: The power output from unit i should be within minimum and maximum output of that unit

$$P_{i\min} \le P_{it} \le P_{i\max}, \quad i = 1, 2, \ldots, N, \tag{4}$$

where $P_{i\min}$ and $P_{i\max}$ are the minimum and the maximum real power outputs of the ith unit, respectively.

But in reality, the objective function of the ED problem is nondifferentiable at some points due to the valve-point loading effects. Therefore, the objective function is composed of a set of nonsmooth cost functions. When considering valve-point loading effects, the objective function is generally described as a superposition of sinusoidal functions and quadratic functions and given by

$$C_{it}(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i + \left| e_i \sin\left(f_i \left(P_{i\min} - P_{it}\right)\right)\right|,$$

where e_i and f_i are the coefficients of unit i reflecting valve-point loading effects. The nonsmooth objective function can be reformulated as a smooth objective function in the following way

$$C_{it}(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i + \alpha_{it} + \beta_{it}$$

by adding new variables $\alpha_{it}, \beta_{it} \ge 0$ and the following inequality constraints in the original problem

$$\begin{aligned} & \alpha_{it} \ge e_i \sin\left(f_i \left(P_{i\min} - P_{it}\right)\right) \\ & \beta_{it} \ge -e_i \sin\left(f_i \left(P_{i\min} - P_{it}\right)\right). \end{aligned} \tag{5}$$

The general form of the reformulated ED problem as a constrained nonlinear programming problem is

$$\begin{array}{ll} \min & f(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) \\ s.t. & h_k(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) = 0 \quad k=1,2,\ldots,m_1 \\ & g_k(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) \leq 0 \quad k=1,2,\ldots,m_2 \\ & P_{i\min} \leq P_{it} \leq P_{i\max} \quad i=1,2,\ldots,N,\ t=1,2,\ldots,T \\ & 0 \leq \alpha_{it}, \beta_{it} \leq \max(e_i) \quad i=1,2,\ldots,N,\ t=1,2,\ldots,T, \end{array} \tag{6}$$

where $m_1 = T$ and $m_2 = (4N \times T - 2N)$ and hence the total number of constraints has increased to $(T + 4N \times T - 2N)$ and the total number of variables is now $3N \times T$. Thus this type of ED problem can also be solved by existing gradient-based standard deterministic NLP solvers.

3. **Differential Evolution.** Differential evolution is a simple yet powerful evolutionary algorithm for global optimization problem (7) introduced by Storn and Price [54],

$$\min_{\mathbf{x}\in\Omega} \quad f(\mathbf{x}) \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the set of decision variables, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ with $\Omega = \{\mathbf{x} \in \mathbb{R}^n : lb_j \leq x_j \leq ub_j, \quad j = 1, \ldots, n\}$.

The DE algorithm has become more popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is easy to understand. DE is a floating point encoding for global optimization over continuous spaces. It creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better or equal fitness. DE has three parameters: amplification factor of the differential variation F, crossover control parameter CR, and population size NP.

DE is a direct search method which utilizes NP n-dimensional components points. These NP points are called target points in a population. The target point is defined by $\mathbf{x}_{p,z} = (x_{p1,z}, x_{p2,z}, \ldots, x_{pn,z})$, where z is the index of iteration/generation and $p = 1, 2, \ldots, NP$. NP does not change during the optimization process. The initial population at $z = 1$ is generated randomly and should cover the entire search space such as the jth component of the pth target point in a population is generated by

$$x_{pj,1} = lb_j + rand(0,1) \times (ub_j - lb_j), \quad j = 1, 2, \ldots, n,$$

where $rand(0,1)$ represents a uniformly distributed random number within the range [0,1]. After initialization DE employs evolutionary processes of mutation, crossover and selection until a global solution is reached. DE's basic strategies are described in the following.

3.1. **Mutation.** In the successive generations, DE performs mutation to create mutant point $\mathbf{v}_{p,z+1}$ with respect to each target point $\mathbf{x}_{p,z}$ in the current population. The most commonly used mutation that is refereed as DE/rand/1 is

$$\mathbf{v}_{p,z+1} = \mathbf{x}_{r_1,z} + F(\mathbf{x}_{r_2,z} - \mathbf{x}_{r_3,z}) \tag{8}$$

with uniformly chosen random indices r_1, r_2, r_3 from the set $\{1, 2, \ldots, NP\}$, mutually different and $F > 0$. The indices r_1, r_2 and r_3 are also chosen to be different from the running index p, so that NP must be greater or equal to four to allow for this condition. F is a real and constant parameter $\in [0, 2]$ which controls the amplification of the differential variation $(\mathbf{x}_{r_2,z} - \mathbf{x}_{r_3,z})$. $\mathbf{x}_{r_1,z}$ is called the base point. There are other frequently used mutation strategies available in the literature:

DE/rand/2: $\mathbf{v}_{p,z+1} = \mathbf{x}_{r_1,z} + F(\mathbf{x}_{r_2,z} - \mathbf{x}_{r_3,z}) + F(\mathbf{x}_{r_4,z} - \mathbf{x}_{r_5,z})$
DE/best/1: $\mathbf{v}_{p,z+1} = \mathbf{x}_{\text{best},z} + F(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z})$
DE/best/2: $\mathbf{v}_{p,z+1} = \mathbf{x}_{\text{best},z} + F(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z}) + F(\mathbf{x}_{r_3,z} - \mathbf{x}_{r_4,z})$
DE/target-to-best/1: $\mathbf{v}_{p,z+1} = \mathbf{x}_{p,z} + F(\mathbf{x}_{\text{best},z} - \mathbf{x}_{p,z}) + F(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z})$
DE/target-to-rand/1: $\mathbf{v}_{p,z+1} = \mathbf{x}_{p,z} + K(\mathbf{x}_{r_1,z} - \mathbf{x}_{p,z}) + F(\mathbf{x}_{r_2,z} - \mathbf{x}_{r_3,z})$

where $\mathbf{x}_{\text{best},z}$ is the best individual point in the current generation, r_4 and r_5 are mutually exclusive uniform random indices in the range $[1, NP]$ and $K \in [0, 1]$.

3.2. **Crossover.** In order to increase the diversity of the mutant points' components, crossover is introduced. To this end, the crossover point called as trial point $\mathbf{u}_{p,z+1}$ is formed, where

$$\begin{aligned} u_{pj,z+1} &= \begin{cases} v_{pj,z+1} & \text{if } (r_j \leq CR) \quad \text{or} \quad j = s_p \\ x_{pj,z} & \text{if } (r_j > CR) \quad \text{and} \quad j \neq s_p \end{cases}, \\ j &= 1, 2, \ldots, n. \end{aligned} \tag{9}$$

In (9), the random number $r_j \sim \text{U}[0, 1]$ performs the mixing of jth component of points, $CR \in [0, 1]$ is a constant parameter for crossover which has to be determined by the user, and uniformly chosen random index s_p from the set $\{1, 2, \ldots, n\}$ ensures that $\mathbf{u}_{p,z+1}$ gets at least one component from $\mathbf{v}_{p,z+1}$.

3.3. **Selection.** To decide whether or not it should become a target point of generation $z + 1$, the trial point $\mathbf{u}_{p,z+1}$ is compared to the target point $\mathbf{x}_{p,z}$ using the greedy criterion in the following way

$$\mathbf{x}_{p,z+1} = \begin{cases} \mathbf{u}_{p,z+1} & \text{if } f(\mathbf{u}_{p,z+1}) \leq f(\mathbf{x}_{p,z}) \\ \mathbf{x}_{p,z} & \text{otherwise.} \end{cases}$$

The above three operations are repeated generation to generation until a termination criterion is reached.

4. **Proposed Modified Differential Evolution.** According to Storn and Price [54], DE is much more sensitive to the choice of F than it is to the choice of CR. The suggested choices for the three parameters are: (i) $F \in [0.5, 1]$; (ii) $CR \in [0.8, 1]$; and (iii) $NP = 10 \times n$. Recall that n is the dimensionality of the problem. The parameters in DE are kept constant throughout the entire evolutionary process. However, it is not an easy task to set appropriate parameters since these depend on the nature and size of the optimization problems.

DE's performance depends on the amplification factor of differential variation and crossover control parameter. Hence adaptive control parameters have been implemented in DE in order to obtain a competitive algorithm. In most commonly used mutation (8), three points are chosen randomly and the base point is then chosen at random within the three. This has an exploratory effect but it slows down the convergence of DE. Liu and Lampinen [44] proposed a fuzzy adaptive differential evolution algorithm (FADE) that uses fuzzy logic controllers to adapt the parameters for mutation and crossover. Kaelo and Ali [37] proposed a differential evolution algorithm with random localization (DERL). In this algorithm the best point among three is chosen for the base point and remaining two as differential variation for mutation and F is determined from $[-1, -0.4] \cup [0.4, 1]$ for each individual point. Brest et al. [9] proposed self-adaptive control parameters for differential evolution (jDE). Qin et al. [51] proposed a differential evolution algorithm with strategy adaptation (SaDE). Here a pool of different mutation strategies and a pool of control parameters are maintained. Then, for each target point one mutation strategy and corresponding control parameters are randomly assigned. Das et al. [18] proposed a differential evolution using a neighborhood-based mutation operator (DEGL). In DEGL, two mutation strategies are performed and then recombined with a weight factor. Zhang and Sanderson [65] proposed a adaptive differential evolution with optional external archive (JADE). Mallipeddi et al. [45] proposed a differential evolution algorithm with ensemble of parameters and mutation strategies (EPSDE). All these variants of DE have been proposed for solving nonlinear global optimization problems with simple bounds. Ali presented in [1] a recursive topographical differential evolution algorithm in a cluster of particles energy minimization context. A concise yet rather complete review of differential evolution variants is presented by Das and Suganthan in [17].

In population-based solution method, it is very important to obtain optimum solution with minimum time period. Also the algorithm is to be capable to explore the whole search space as well as to exploit around neighborhood of a reference point (this can be the best point). DE has these features but sometimes it is getting stuck at local solution mainly because of using mutation and control parameters. So in this paper, we propose a modified differential evolution (mDE) that is capable of solving economic dispatch problems. The mDE algorithm includes the following modifications:

1. a combination of two mutation strategies with a weight factor;
2. a cyclical usage of the overall best point as base point in mutation;
3. the self-adaptive technique for control parameters;
4. a modified selection (discussed in Section 5.3) to solve the economic dispatch problems.

These modifications allow mDE to maintain its exploratory feature as well as exploit the region around the best point cyclically for each mutant point and at the same time expedite the convergence. The modified differential evolution is outlined below.

4.1. **Modified Mutation.** Mutation plays vital role in DE which explores the entire search space and at the same time expedites convergence. But a proper balance between exploration and exploitation is required for an effective operation. DE/rand/1 mutation strategy explores the entire search space but converges slowly. On the other hand, DE/best/1 strategy exploits around the best point found so far and converges rapidly. With this strategy a local solution may be obtained before the global solution can be reached. Also the amplification factor F is always problem dependent and should not be constant. It enhance the exploration feature of mutation. In this context, here in mDE, we propose a modified mutation that is a mixture of two mutation strategies. The first mutation strategy is a combination of two different strategies with a weight factor and the other one is the cyclical use of DE/best/1 strategy, both with self-adaptive amplification factor F.

Firstly, three points are randomly chosen from the target population at current generation and the best point among these three based on the fitness is used as the base point and remaining two points are used in differential variation to create first mutant point $\mathbf{v}^{rb}_{p,z+1}$

$$\mathbf{v}^{rb}_{p,z+1} = \mathbf{x}_{r_1,z} + F_{p,z+1}(\mathbf{x}_{r_2,z} - \mathbf{x}_{r_3,z}) \tag{10}$$

where $\mathbf{x}_{r_1,z}$ is the best point among the three randomly chosen points. r_2 and r_3 are the indices of remaining two points. $F_{p,z+1}$ is the self-adaptive parameter for amplifying the differential variation (see (14)). This mutation has exploration as well as random localization features.

Secondly, we use DE/rand/1 mutation strategy to create second mutant point $\mathbf{v}^r_{p,z+1}$

$$\mathbf{v}^r_{p,z+1} = \mathbf{x}_{r_4,z} + F_{p,z+1}(\mathbf{x}_{r_5,z} - \mathbf{x}_{r_6,z}) \tag{11}$$

where uniformly chosen random indices r_4, r_5, r_6 from the set $\{1, 2, \ldots, NP\}$ are mutually different and also chosen to be different from the running index p.

Finally, we combine the above two mutant points using a scalar weight factor $\omega \in [0, 1]$ to form actual mutant point $\mathbf{v}_{p,z+1}$

$$\mathbf{v}_{p,z+1} = \omega_{p,z+1}\mathbf{v}^{rb}_{p,z+1} + (1 - \omega_{p,z+1})\mathbf{v}^r_{p,z+1} \tag{12}$$

where $\omega_{p,z+1}$ is a self-adaptive weight factor that balances the combination of the two previous mutant points (also see (14)). Clearly, if $\omega_{p,z+1} = 1$, then the actual mutant point generation reduces to (10) and if $\omega_{p,z+1} = 0$, reduces to (11).

Furthermore, at every R generation, we use the best point found so far as base point and randomly chosen two points for differential variation

$$\mathbf{v}_{p,z+1} = \mathbf{x}_{\text{best}} + F_{p,z+1}(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z}). \tag{13}$$

This modified mutation allows mDE to maintain its exploration and exploitation features as well as expedite the convergence.

4.2. **Self-Adaptive Control Parameters.** In mDE, we use the self-adaptive technique proposed by Brest et al. [9] for control parameters F, CR and ω that generates a different set (F_p, CR_p, ω_p) for each individual point in the population. For $z = 1$, F_p is generated according to $F_{p,1} = F_l + rand(0, 1) \times (F_u - F_l)$, where F_l and F_u are the lower and upper bounds of F, respectively. The values of CR and ω are generated randomly from $[0, 1]$. Then new control parameters for each individual point at next generation $F_{p,z+1}$, $CR_{p,z+1}$ and $\omega_{p,z+1}$ are calculated as

$$\begin{aligned} F_{p,z+1} &= \begin{cases} F_l + \lambda_1 \times (F_u - F_l), & \text{if } \lambda_2 < \tau_1 \\ F_{p,z}, & \text{otherwise} \end{cases} \\ CR_{p,z+1} &= \begin{cases} \lambda_3, & \text{if } \lambda_4 < \tau_2 \\ CR_{p,z}, & \text{otherwise} \end{cases} \\ \omega_{p,z+1} &= \begin{cases} \lambda_5, & \text{if } \lambda_6 < \tau_3 \\ \omega_{p,z}, & \text{otherwise,} \end{cases} \end{aligned} \tag{14}$$

where $\lambda_k \sim \text{U}[0, 1], k = 1, \ldots, 6$ and τ_1, τ_2, τ_3 represent probabilities to adjust parameters F_p, CR_p and ω_p, respectively. $F_l = 0.1$ and $F_u = 1.0$, so the new $F_{p,z+1}$ takes a value from $[0.1, 1.0]$ in a random manner. The new $CR_{p,z+1}$ and $\omega_{p,z+1}$ take values from $[0, 1]$. $F_{p,z+1}$, $CR_{p,z+1}$ and $\omega_{p,z+1}$ are obtained before the mutation is performed. So, they influence the mutation, crossover and selection operations of the new point $\mathbf{x}_{p,z+1}$.

4.3. **Bounds Check.** When generating the mutant point, some components can be generated outside the domain Ω. So, in mDE the bounds of each individual point's component should be checked with the following projection of bounds:

$$u'_{pj,z+1} = \begin{cases} lb_j & \text{if } u'_{pj,z+1} < lb_j \\ ub_j & \text{if } u'_{pj,z+1} > ub_j \\ u'_{pj,z+1} & \text{otherwise.} \end{cases}$$

4.4. **Termination Criterion.** Let $G_{\max}$ be the maximum number of generations. If $f_{\max,z}$ and $f_{\min,z}$ are the maximum and minimum objective function values respectively, attained at generation z then the proposed mDE algorithm terminates if $z > G_{\max}$ or $(f_{\max,z} - f_{\min,z}) \leq \epsilon$, for a very small positive number ϵ.

5. **Solving Economic Dispatch Problems with Modified Differential Evolution.** In this section, the procedures to solve the reformulated ED problem (and the original nonsmooth ED problem as well) by mDE are described. The values of $\mathbf{P}$ that has $(N \times T)$ components of real power outputs are initialized randomly within generating units operating limits. The values of α_{it} and β_{it} are also initialized randomly within $[0, \max(e_i)]$. So in mDE when solving ED problems as previously described $\mathbf{x} = (\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and $n = N \times T$ for all the original problems and $n = 3N \times T$ for the problems that are reformulated.

5.1. **Constraints Handling.** To solve ED problem by mDE the equality constraints (1) and inequality constraints (3) and after reformulation (5) must be satisfied. In the following, the constraints satisfaction by mDE are outlined.

Real Power Balance: To solve ED problem by mDE, it is very important to create a population of individual points that satisfy the real power balance constraints (1). To satisfy these constraints by an individual point, power output of any one unit is selected as the dependent power output P_{lt} randomly [4, 19, 48, 53, 59], where $l \in \{1, 2, \ldots, N\}$. The dependent power output P_{lt} is computed from (15)

$$P_{lt} = D_t + L_t - \sum_{i=1, i\neq l}^{N} P_{it}, \quad t = 1, 2, \ldots, T. \tag{15}$$

The transmission loss L_t is a function of all generating units including that of dependent unit (see (2)), and is given by

$$L_t = \sum_{i=1, i\neq l}^{N} \sum_{j=1, j\neq l}^{N} P_{it} B_{ij} P_{jt} + 2P_{lt} \left(\sum_{i=1, i\neq l}^{N} B_{li} P_{it} \right) + B_{ll} P_{lt}^2. \tag{16}$$

After substituting the value of L_t from (16) into (15) and rearranging, equation (15) becomes

$$B_{ll} P_{lt}^2 + \left(2 \sum_{i=1, i\neq l}^{N} B_{li} P_{it} - 1 \right) P_{lt} + \left(D_t + \sum_{i=1, i\neq l}^{N} \sum_{j=1, j\neq l}^{N} P_{it} B_{ij} P_{jt} - \sum_{i=1, i\neq l}^{N} P_{it} \right) = 0. \tag{17}$$

The equation (17) is a quadratic equation and the value of P_{lt} can be easily calculated and must satisfy the constraints (3) and (4).

Inequality Constraints: Since mDE is a population based technique, to satisfy the inequality constraints of the ED problem (6), the constraint violation of an individual point in a population is defined by $\max\{0, g_k(\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})\}, k = 1, 2, \ldots, m_2$. For an individual point, if all constraints are satisfied the violation should be zero, otherwise it should be the sum of constraint violation. So the sum of constraint violation of an individual is calculated by using the following equation

$$\phi(\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{k=1}^{m_2} \max\left\{0, g_k(\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})\right\}. \tag{18}$$

5.2. **Constraints Handling Techniques.** In the last decades many constraints handling techniques have been proposed using stochastic solution methods. For ED problems, equality constraints can be handled using (15). The most used technique for constrained nonlinear programming problem is penalty function method, where a penalty term is added to the objective function with a positive penalty for infeasible solutions. This technique has been used for solving ED problems [2, 3, 4, 27, 32]. However, the performance of this technique is not always satisfactory due to the difficult task concerned with choosing an appropriate value for the penalty parameter. Deb [20] proposed an efficient constraints handling technique based on feasibility and dominance rules. This technique is easy to implement and the objective function need not be evaluated for infeasible individual. Runarsson and Yao [52] proposed stochastic and global competitive ranking methods for constrained problems. Here both the objective function and the constraints are evaluated for all individuals in a population and each individual is ranked relative to the entire population using objective and constraint violation values separately. Mallipeddi and Suganthan [46] proposed an ensemble of constraints handling techniques, where each of them has its own population and generates new trial points. These points compete with each other to generate the next populations.

Due to its simplicity the herein implemented technique for constraints handling rely on the feasibility and dominance rules.

5.3. **Modified Selection in mDE.** In original DE, an individual point in a population replaces the parent only if it has better or equal objective function value meaning that a trial point replaces the target point if the objective function value of the trial point is smaller than or equal to that of the target point, and becomes a member in a population at generation $z+1$. Since the ED problem is a constrained problem, in mDE we propose a modified selection of an individual point at generation $z+1$ based on the feasibility and dominance rules of that individual point, as proposed by Deb [20] and therein called the tournament selection.

In this technique, first the constraint violation ϕ is calculated for all the individuals. Then the objective function f is evaluated only for feasible individuals. Let $(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta})_z$ and $(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta})_{z+1}$ be the target and trial points, respectively. Two individuals are compared at a time to select which one will be the member in a population at $z+1$, the following criteria are always enforced:

1. any feasible point is preferred to any infeasible point;
2. between two feasible points, one having better objective function value is preferred;
3. between two infeasible points, one having smaller constraint violation is preferred.

In this technique the feasible individuals or the individuals having smaller constraint violation are always selected for next generation target points.

5.4. **Fitness Calculation.** The fitness function of a target point is calculated as follows [20]

$$\Phi(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) = \begin{cases} f(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) & \text{if } \phi(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) = 0 \\ f_{\max\_feas} + \phi(\mathbf{P},\boldsymbol{\alpha},\boldsymbol{\beta}) & \text{otherwise,} \end{cases} \tag{19}$$

where $f_{\max\_feas}$ is the objective function value of the worst feasible target point. When all target points are infeasible then its value is set to zero. In mDE, this fitness function is used only to choose the best and worst individual points in a target population.

5.5. **The mDE Algorithm.** The algorithm of the herein proposed modified differential evolution to solve ED problems is described in the following:

Step 1: Set the values of parameters NP, $G_{\max}$, R, F_l, F_u, τ_1, τ_2, τ_3, ϵ.
Step 2: Set $z = 1$. Initialize the target population $\mathbf{x}_{p,1}, p = 1, 2, \ldots, NP$, and $F_{p,1}$, $CR_{p,1}$ and $\omega_{p,1}$.
Step 3: Balance the demand constraint, calculate constraint violation ϕ using (18) and fitness function Φ using (19).
Step 4: Identify $f_{\max,z}$ and $f_{\min,z}$ and set $f_{\text{best}} = f_{\min,z}$ and $\mathbf{x}_{\text{best}} = \mathbf{x}_{\min,z}$.
Step 5: If termination condition is met stop. Otherwise set $z = z + 1$.
Step 6: Compute new control parameters $F_{p,z}$, $CR_{p,z}$ and $\omega_{p,z}$ using (14).
Step 7: Compute mutant point $\mathbf{v}_{p,z}$:
If $\text{MOD}(z, R) = 0$ compute mutant point using (13), otherwise using (12).
Step 8: Perform crossover to make trial point $\mathbf{u}_{p,z}$ using (9).
Step 9: Check the domains of the trial point using (4).
Step 10: Balance the demand constraint, calculate constraint violation ϕ.
Step 11: Perform modified selection (discussed in Section 5.3).
Step 12: Calculate fitness function Φ.
Step 13: Go to step 4.

6. **Experimental Results.** We code mDE in C with AMPL [26] interfacing and compile with Microsoft Visual Studio 9.0 compiler in a PC having 2.5 GHz Intel Core 2 Duo processor and 4 GB RAM. We set the value of parameter $NP = \min(100, 10n)$ so that the population size depends on the dimension of the problem. We also set $\tau_1 = \tau_2 = \tau_3 = 0.1$ as suggested in [9] and $\epsilon = 10^{-6}$. We set different value for $G_{\max}$ depending on the test problem. We considered five test problems of economic dispatch from literature. The characteristics of the five test problems are outlined in Table 1.

TABLE 1. Characteristics of ED problems

Prob.	No. of Variables	No. of Constraints		
		Equality	Inequality	Total
P1	3	1	0	1
P2	3(6)*	1	6	7
P3	13(26)*	1	26	27
P4	40	1	0	1
P5	120(240)*	24	470	494

*Additional variables indicated in parentheses to reformulate nonsmooth objective function

We model these problems (original and reformulated) in AMPL and GAMS [10] modeling systems[1]. We take input data from literature. We compared the results obtained by mDE with existing deterministic global NLP solvers, existing deterministic local NLP solvers and other stochastic global techniques from literature.

[1]To access these models please visit http://www.norg.uminho.pt/emgpf/problems.htm

At first we will show the effectiveness of using (13) corresponding to R cyclically. We considered two benchmark constrained test problems g06 and g10 from [42]. We run mDE for two cases. In "case 1" we used (13) at every R generations and in "case 2" we used (13) at $G_{\max}/R$ and then at every R. In both cases we set $R = 10$. Thirty independent runs were carried out. The comparison of obtained results are shown in Table 2. In the table "f_{opt}" means the best known optimal solution. "f_{best}" represents the best obtained solution, "f_{avg}" represents the average of the best objective function values and "std. dev." represents the standard deviation of the best objective function values, after 30 runs. From the table it is shown that, for the two test problems, mDE case 2 gives better results. Figures 1 and 2 show the profile of average of the best objective function values at different generations for g06 and g10, respectively. From these figures it is also shown that, mDE case 2 gives better performance. Hereafter we will use mDE to denote the case 2.

TABLE 2. Comparison of results of g06 and g10

Prob.	f_{opt}	$G_{\max}$	case 1			case 2		
			f_{best}	f_{avg}	std. dev.	f_{best}	f_{avg}	std. dev.
g06	-6961.81	500	-6961.82	-6741.13	6.58E+02	-6961.82	-6961.82	5.48E-06
g10	7049.25	2000	7049.46	7138.91	7.82E+01	7049.25	7083.69	4.93E+01

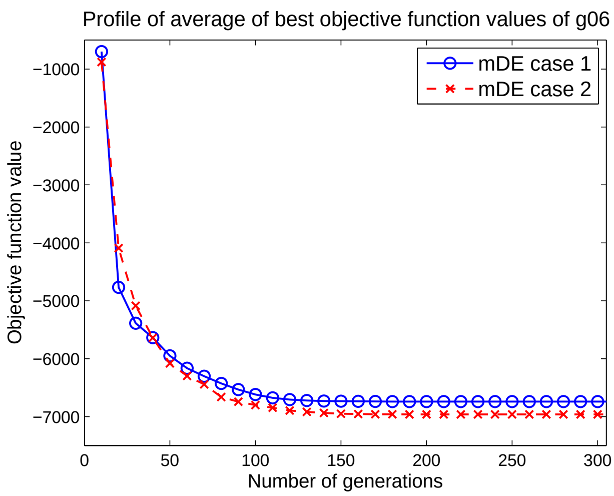

FIGURE 1. Profile of average of best objective function values of g06 at different generations

6.1. Comparison with Deterministic Global and Local Solvers.

We run mDE 30 times and report the best objective function value. We run standard NLP solvers in NEOS server[2]. We emphasize that here the reformulated smooth ED problems are solved. The best objective function value obtained over the 30 runs by mDE and the solutions obtained by the deterministic global NLP solvers are shown in Table 3. The solutions obtained by the deterministic local NLP solvers

[2] http://neos-server.org/neos/

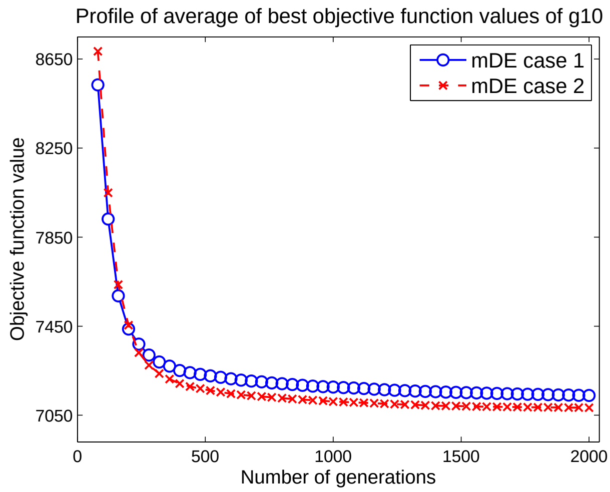

FIGURE 2. Profile of average of best objective function values of g10 at different generations

TABLE 3. Comparison of best objective obtained by global NLP solvers and mDE

Solvers	P1		P2		P3		P4		P5	
	Obj.	Iter.	Obj.	Iter.	Obj.	Iter.	Obj.	Iter.	Obj.	Iter.
ALGENCAN	3619.75	6	8419.97	8	25410.85	8	141127.39	7	45005.00	12
BARON	3619.75	–	–	–	–	–	141127.39	–	–	–
Couenne	–	–	8234.07	208	24169.92	4893	141127.39	152	–	–
LINDOGlobal	3619.75	104	8382.73	205	25404.54	709	141127.39	376	47368.69	17102
mDE	3619.75	35	8234.07	500	24169.92	5000	141127.39	1908	46620.85	20000

(–) indicates no information, infeasible or run time error by the solver

are displayed in Table 4. In both tables "Obj." is the obtained solution and "Iter." represents the number of iterations/generations required to achieve the solution with a specified accuracy. We used the default values given by each solver.

TABLE 4. Comparison of best objective obtained by local NLP solvers

Solvers	P1		P2		P3		P4		P5	
	Obj.	Iter.	Obj.	Iter.	Obj.	Iter.	Obj.	Iter.	Obj.	Iter.
CONOPT	3619.75	11	8493.88	19	25458.80	129	141127.39	24	46410.54	391
FilterSQP	3619.75	7	8416.98	8	25416.94	33	141127.39	4	–	–
IPOPT	3619.75	9	8562.41	142	25466.62	189	141127.39	17	47693.02	466
KNITRO	3619.75	4	–	–	25404.54	365	141127.39	11	47720.57	1788
LANCELOT	3619.75	13	8250.20	53	25404.54	104	141127.39	18	45503.98	196
LOQO	3619.75	17	8234.07	201	25405.36	202	141127.44	22	41539.76	500
MINOS	3619.75	7	8499.45	3	25122.78	44	141127.39	39	47290.49	1960
SNOPT	3619.75	11	8668.24	5	25592.81	29	141127.39	66	46501.88	1778

(–) indicates no information, infeasible or run time error by the solver

The ED problem P1 has three generating units with hourly demand 300 and transmission loss. Data is listed in [55]. The objective function of this problem is smooth. We set $G_{\max} = 100$ in mDE. From Table 3 and Table 4 it is shown that all solvers gave the optimal objective function value of 3919.75 although Couenne did not give a feasible solution, as it got stuck at lower bounds.

P2 has three generating units with hourly demand 850 [53]. The objective function of this problem is nonsmooth. After reformulating, the problem has nine variables and seven constraints. In mDE, we set $G_{\max} = 500$. It is shown that solvers mDE, Couenne and LOQO gave the optimal objective function value of 8234.07, and the other solvers did not give the optimal. In this problem, there exists the trigonometric function "sine" and in fact, BARON cannot be directly applied to problems with trigonometric functions. KNITRO did not give a feasible solution even when setting $xtol = 10^{-20}$.

Problem P3 has 13 generating units with hourly demand 2520 [53]. The objective function of this problem is nonsmooth. After reformulating, the number of variables are 39 and constraints are 27. We set $G_{\max} = 5000$. Couenne and mDE gave the optimal objective function value of 24169.92. In this problem, BARON also did not evaluate the "sine" functions.

The ED problem P4 has 40 generating units with hourly demand 10500 [14]. The objective function of this problem is smooth. We set $G_{\max} = 3000$ in mDE. From Table 3 and Table 4 it is shown that all solvers except LOQO gave the optimal objective function value of 141127.39.

P5 has five generating units with 24 hours demand and with transmission loss [3]. The objective function of this problem is nonsmooth. After reformulating, the number of variables are 360 and constraints are 494. In mDE, we set $G_{\max} =$ 20000. LOQO gave the best objective function value of 41539.76, ALGENCAN gave 45005.00 and mDE gave 46620.85. Here BARON also did not evaluate the "sine" functions, Couenne exited after run time error and FilterSQP did not give a feasible solution.

From the above discussion it is to be noted that although mDE takes more iterations to solve each problem, the quality of the solutions are in general better than those of the most deterministic NLP solvers. In fact, the number of iterations is not a fair and definite measure of comparison since the amount of work done at each iteration is different from one solver to the other. We did not compare the performance of all solvers based on solution time because of different machines used.

6.2. **Comparison with Stochastic Techniques.** Since some stochastic global techniques available in literature have been applied to standard ED problems, we also run mDE to solve those original nonsmooth ED problems. For fair comparison we also coded jDE [9] with modified selection (discussed in Section 5.3) in C with AMPL interfacing. jDE is based on the DE/rand/1 mutation strategy with self-adaptive control parameters for F and CR. Then we compared the obtained results by mDE with the results obtained by jDE and other techniques reported in literature. Here we also set the above mentioned values of parameters for mDE and jDE. It is to be noted that the stochastic techniques available in literature were not applied to all the above mentioned ED problems. So we compared the result of each problem separately.

For problem P1, we set $G_{\max} = 100$ for mDE and jDE. Thirty independent runs were carried out and at every $G_{\max}/20$ the obtained best objective function value was reported and made average. We plot the profile of average of the best objective function values by jDE and mDE at different generations in Figure 3. The best result obtained after 30 runs and the result by GA [55] are shown in Table 5.

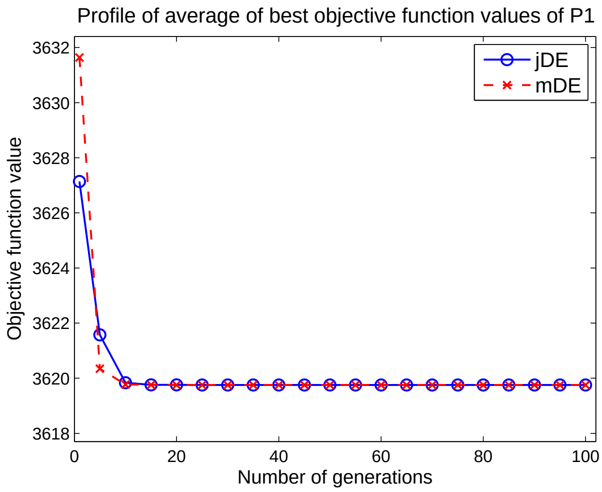

FIGURE 3. Profile of average of best objective function values of P1 at different generations

TABLE 5. Comparison of results of P1

	GA	jDE	mDE
Obj.	3619.76	3619.76	3619.76

When solving problem P2, we set $G_{\max} = 100$. Figure 4 contains the plot of average of the best objective function values at different generations after 30 runs. The best result obtained after 30 runs and the results by IEP [50], IFEP [53], EP [64], EP-SQP [2], MPSO [49] and PSO-SQP [59] are shown in Table 6.

TABLE 6. Comparison of results of P2

	IEP	IFEP	EP	EP-SQP	MPSO	PSO-SQP	jDE	mDE
Obj.	8234.09	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07

We set $G_{\max} = 1000$ in problem P3. Figure 5 shows the plot of average of the best objective function values at different generations. The best result obtained

TABLE 7. Comparison of results of P3

	HSS	GA-SA	EP-SQP	PSO-SQP	HGA	jDE	mDE
Obj.	24275.00	24275.71	24266.44	24261.05	24169.92	24169.93	24169.92

by mDE and jDE after 30 runs, as well as the results by HSS [16], GA-SA [62], EP-SQP [2], PSO-SQP [59] and HGA [32] are shown in Table 7.

For problem P4, we set $G_{\max} = 3000$ for jDE and mDE. We plot the profile of average of the best objective function values by jDE and mDE at every $G_{\max}/20$

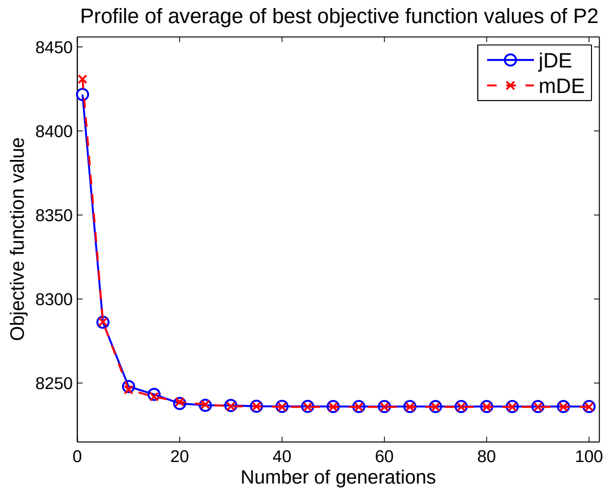

FIGURE 4. Profile of average of best objective function values of P2 at different generations

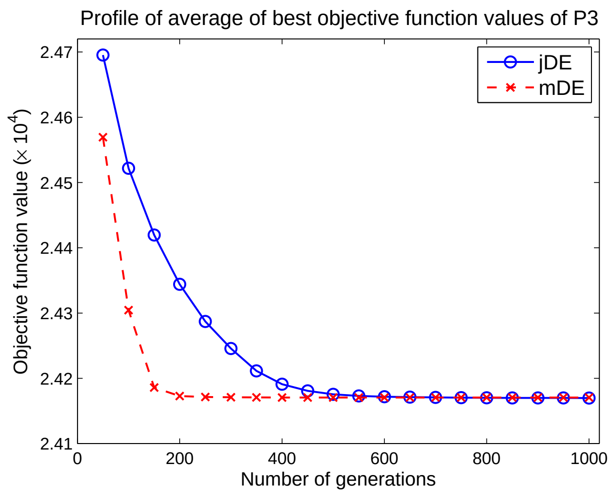

FIGURE 5. Profile of average of best objective function values of P3 at different generations

generation in Figure 6. The best result obtained after 30 runs and the results by SA, GA, HDE, VSHDE available in [14] and ALHN [21] are shown in Table 8.

TABLE 8. Comparison of results of P4

	SA	GA	HDE	VSHDE	ALHN	jDE	mDE
Obj.	164069.36	144486.02	143955.83	143943.90	143926.90	141127.39	141127.39

Finally, we set $G_{\max} = 10000$ when solving problem P5. We plot the profile of average of the best objective function values at different generations, as shown in

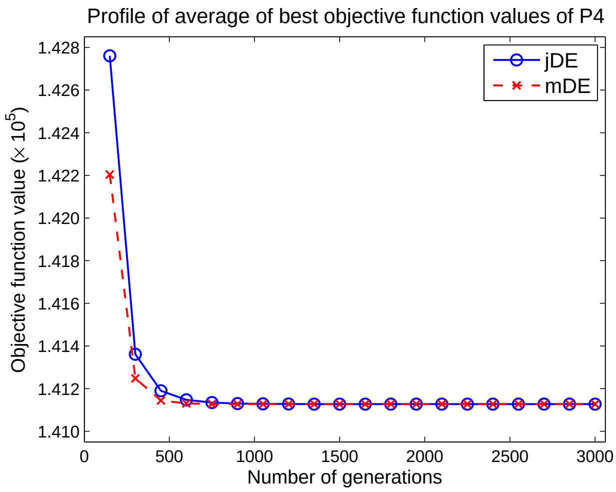

FIGURE 6. Profile of average of best objective function values of P4 at different generations

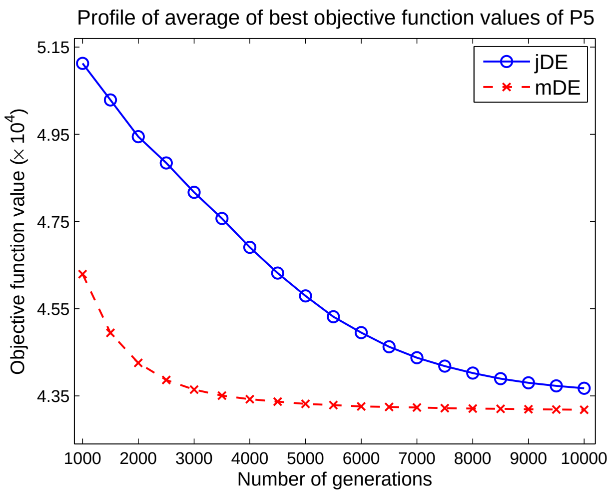

FIGURE 7. Profile of average of best objective function values of P5 at different generations

TABLE 9. Comparison of results of P5

	SA	IDE	DE	jDE	mDE
Obj.	47356.00	45800.00	43213.00	43248.20	43057.83

Figure 7. The best result obtained after 30 runs and the results by SA [48], IDE [4] and DE [3] are shown in Table 9.

From the above tables it is shown that the proposed mDE is rather competitive when compared with other stochastic techniques for solving standard economic dispatch problems. From the tables and figures it is also shown that mDE outperforms jDE.

7. **Conclusions.** In real world, power generation companies try to improve the operating efficiency of their power plants and tangible improvements can still be achieved through the application of more robust solution techniques. In this paper, to make DE algorithm more efficient to handle the constraints in economic dispatch problems with valve-point loading effects and transmission loss, a modified differential evolution (mDE) algorithm has been proposed. The modifications focus on the self-adaptive techniques for control parameters, a modified mutation and a modified selection.

We emphasize the modifications that mostly influence the efficiency of the algorithm. The mixed modified mutation with self-adaptive amplification factor in mDE algorithm aims at exploring the whole search space (when using recombination of two mutation strategies) and exploiting the neighborhood of the best point found so far (when using DE/best/1 mutation strategy cyclically). The modified selection, to handle the constraints effectively, uses the tournament selection based on feasibility and dominance rules that makes pair-wise comparison among feasible and infeasible solutions based on the degree of constraint violation. A penalty based fitness function that does not require any penalty parameter is used to calculate the fitness of each individual point in a population in order to identify the best and the worst points.

To test the effectiveness of the proposed mDE algorithm, five types of ED problems have been solved in this study and the results show that the herein proposed mDE has a performance that improves over most of the other methods of deterministic and stochastic nature in comparison. Future developments will focus on the extension of the mDE to ED problems with mixed integer variables.

Acknowledgments. The authors thank the anonymous referees, the Associate Editor and the Editor-in-Chief for their valuable comments and suggestions.

The first author acknowledges Ciência 2007 of FCT, Fundação para a Ciência e a Tecnologia (Foundation for Science and Technology), Portugal for the financial support under fellowship grant: C2007-UMINHO-ALGORITMI-04. The second author acknowledges FEDER COMPETE, Programa Operacional Fatores de Competitividade (Operational Programme Thematic Factors of Competitiveness) and FCT for the financial support under project grant: FCOMP-01-0124-FEDER-022674.

REFERENCES

[1] M.M. Ali, *A recursive topographical differential evolution algorithm for potential energy minimization*, J. Ind. Manag. Optim., **6(1)** (2010), 29–46.

[2] 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., **17(2)** (2002), 411–416.

[3] R. Balamurugan and S. Subramanian, *Differential evolution-based dynamic economic dispatch of generating units with valve-point effects*, Electr. Power Compon. Syst., **36** (2008), 828–843.

[4] R. Balamurugan and S. Subramanian, *An improved differential evolution based dynamic economic dispatch with nonsmooth fuel cost function*, J. Electr. Syst., **3(3)** (2007), 151–161.

[5] H.-G. Beyer and H.-P. Schwefel, *Evolution strategies: A comprehensive introduction*, J. Nat. Comput., **1(1)** (2002), 3–52.

[6] P. Belotti, "Couenne: a user's manual", 2009. Available from: http://www.coin-or.org/Couenne
[7] S.I. Birbil and S.C. Fang, *An electromagnetism-like mechanism for global optimization*, J. Glob. Optim., **25** (2003), 263–282.
[8] E.G. Birgin, C.A. Floudas and J.M. Martínez, *Global minimization using an augmented Lagrangian method with variable lower-level constraints*, Math. Program. Ser. A, **125(1)** (2010), 139–162.
[9] J. Brest, S. Greiner, B. Bošković, M. Mernik and V. Žumer, *Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems*, IEEE Trans. Evol. Comput., **10** (2006), 646–657.
[10] A. Brooke, D. Kendrick, A. Meeraus and R. Raman, "GAMS: A User's Guide", Release 2.25, The Scientific Press, South San Francisco, 1992.
[11] R.H. Byrd, J. Nocedal and R.A. Waltz, *KNITRO: An integrated package for nonlinear optimization*, In "Large-Scale Nonlinear Optimization" (eds. G. Pillo and M. Roma), Springer Verlag, (2006), 35–59.
[12] C. Chen, *Non-convex economic dispatch: a direct search approach*, Energy Convers. Manage., **48(1)** (2007), 219–225.
[13] P.H. Chen and H.C. Chang, *Large-scale economic dispatch by genetic algorithm*, IEEE Trans. Power Syst., **10** (1995), 1919–1926.
[14] J.-P. Chiou, *Variable scaling hybrid differential evolution for large-scale economic dispatch problems*, Electr. Power Syst. Res., **77** (2007), 212–218.
[15] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., **28(2)** (1991), 545–572.
[16] D.B. Das and C. Patvardhan, *Solution of economic load dispatch using real coded hybrid stochastic search*. Int. J. Electr. Power Energy Syst., **21** (1999), 165–170.
[17] S. Das and P.N. Suganthan, *Differential evolution: A survey of the state-of-the-art*, IEEE Trans. Evol. Comput., **15** (2011), 4–31.
[18] S. Das, A. Abraham, U.K. Chakraborty and A. Konar, *Differential evolution using a neighborhood-based mutation operator*, IEEE Trans. Evol. Comput., **13** (2009), 526–553.
[19] K. Deb, *Scope of stationary multi-objective evolutionary optimization: A case study on a hydro-thermal power dispatch problem*, J. Glob. Optim., **41(4)** (2008), 479–515.
[20] K. Deb, *An efficient constraint handling method for genetic algorithms*, Comput. Meth. Appl. Mech. Eng., **186** (2000), 311–338.
[21] V.N. Dieu and W. Ongsakul, *Augmented Lagrange Hopfield network for large scale economic dispatch*, Proc. Int. Symp. Electr. Electron. Eng., (2007), 19–26.
[22] M. Dorigo, V. Maniezzo and A. Colorni, *The ant system: optimization by a colony of cooperating agents*, IEEE Trans. Syst. Man Cybern., **26(B)** (1996), 29–41.
[23] A. Drud, "CONOPT: solver manual", ARKI Consulting and Development. Available from: http://www.gams.com/dd/docs/solvers/conopt.pdf
[24] D.E. Finkel and C.T. Kelley, *Additive scaling and the DIRECT algorithm*, J. Glob. Optim., **36** (2006), 597–608.
[25] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., **91(2)** (2002), 239–269.
[26] R. Fourer, D.M. Gay and B.W. Kernighan, "AMPL: A modeling language for mathematical programming", Boyd & Fraser Publishing Co., Massachusetts, 1993.
[27] Z. Gaing, *Particle swarm optimization to solve the economic dispatch considering the generators constraints*, IEEE Trans. Power Syst., **18(3)** (2003), 1187–1195.
[28] Z.W. Geem, J.-H. Kim and G.V. Loganathan, *A new heuristic optimization algorithm: harmony search*, Simulation, **76(2)** (2001), 60–68.
[29] Ph.E. Gill, W. Murray and M.A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev., **47** (2005), 99–131.
[30] F. Glover and M. Laguna, "Tabu Search", Kluwer Academic Publishers, Boston, 1997.
[31] G. Granelli and M. Montagna, *Security-constrained economic dispatch using dual quadratic programming*, Electr. Power Syst. Res., **56(1)** (2000), 71–80.
[32] D. He, F. Wang and Z. Mao, *A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect*, Electr. Power Energy Syst., **30** (2008), 31–38.

[33] K. Hindi and M. Ghani, *Dynamic economic dispatch for large-scale power systems: a Lagrangian relaxation approach*, Electr. Power Syst. Res., **13(1)** (1991), 51–56.

[34] J.H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1997.

[35] W. Huyer and A. Neumaier, *Global optimization by multilevel coordinate search*, J. Glob. Optim., **14** (1999), 331–355.

[36] A. Jabr, H. Coonick and J. Cory, *A homogeneous linear programming algorithm for the security constrained economic dispatch problem*, IEEE Trans. Power Syst., **15(3)** (2000), 930–937.

[37] P. Kaelo and M.M. Ali, *A numerical study of some modified differential evolution algorithms*, Eur. J. Oper. Res., **169** (2006), 1176–1184.

[38] D. Karaboga and B. Basturk, *A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm*, J. Glob. Optim., **39(3)** (2007), 459–471.

[39] A. Keib, H. Ma and J. Hart, *Environmentally constrained economic dispatch using the Lagrangian relaxation method*, IEEE Trans. Power Syst., **9(4)** (1994), 1723–1727.

[40] J. Kennedy, R.C. Eberhart and Y. Shi, "Swarm Intelligence", Morgan Kaufmann, San Francisco, 2001.

[41] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, *Optimization by simulated annealing*. Science, **220** (1983), 671–680.

[42] J.J. Liang, T.P. Runarsson, E.M. -Montes, M. Clerc, P.N. Suganthan, C.A. Coello Coello and K. Deb, *Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization*, Technical Report, 2006.

[43] "LINDOGlobal: solver manual", Lindo Systems, Inc., 2007. Available from: http://www.gams.com/dd/docs/solvers/lindoglobal.pdf

[44] J. Liu and J. Lampinen, *A fuzzy adaptive differential evolution algorithm*, Soft Comput., **9** (2005), 448–462.

[45] R. Mallipeddi, P.N. Suganthan, Q.K. Pan and M.F. Tasgetiren, *Differential evolution algorithm with ensemble of parameters and mutation strategies*, Appl. Soft Comput., **11** (2011), 1679–1696.

[46] R. Mallipeddi and P.N. Suganthan, *Ensemble of constraint handling techniques*, IEEE Trans. Evol. Comput., **14(4)** (2010), 561–579.

[47] B.A. Murtagh and M.A. Saunders, "MINOS 5.5 User's Guide, Report SOL 83-20R", Dept. of Operations Research, Stanford University, 1998.

[48] C.K. Panigrahi, P.K. Chattopadhyay, R.N. Chakrabarti and M. Basu, *Simulated annealing technique for dynamic economic dispatch*, Electr. Power Compon. Syst., **34(5)** (2006), 577–586.

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

[50] Y.-M. Park, J.R. Won and J.B. Park, *New approach to economic load dispatch based on improved evolutionary programming*, Eng. Intell. Syst. Electr. Eng. Commun., **6(2)** (1998), 103–110.

[51] A.K. Qin, V.L. Huang and P.N. Suganthan, *Differential evolution algorithm with strategy adaptation for global numerical optimization*, IEEE Trans. Evol. Comput., **13** (2009), 398–417.

[52] T.P. Runarsson and X. Yao, *Constrained evolutionary optimization – the penalty function approach*, In: R. Sarker, M. Mohammadian and X. Yao (eds.), Evolutionary Optimization: International Series in Operations Research and Management Science, (2003), 87–113.

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

[54] R. Storn and K. Price, *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*, J. Glob. Optim., **11** (1997), 341–359.

[55] C.-T. Su and W.-T. Tyen, *A Genetic Algorithm Approach Employing Floating Point Representation for Economic Dispatch of Electric Power Systems*, Proc. Int. Congr. Model. Simul., (1997), 1444–1449.

[56] S. Takriti and B. Krasenbrink, *A decomposition approach for the fuel-constrained economic power-dispatch problem*, Eur. J. Oper. Res., **112(2)** (1999), 460–466.

[57] M. Tawarmalani and N. Sahinidis, "Convexification and global optimization in continuous and mixed-integer nonlinear programming", Kluwer Academic Publishers, Dordrecht, 2002.

[58] R.J. Vanderbei and D.F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., **13** (1999), 231–252.
[59] T.A.A. Victorie and A.E. Jeyakumar, *Hybrid PSO-SQP for economic dispatch with valve-point effect*, Electr. Power Syst. Res., **71** (2004), 51–59.
[60] A. Wächter and L.T. Biegler, *On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming*, Math. Program., **106(1)** (2006), 25–57.
[61] D.C. Walters and G.B. Sheble, *Genetic algorithm solution of economic dispatch with valve-point loadings*, IEEE Trans. Power Syst., **8(3)** (1993), 1325–1331.
[62] K.P. Wong and Y.W. Wong, *Genetic and genetic/simulated-annealing approaches to economic dispatch*, IEEE Proc. Gener. Transm. Distrib., **141(5)** (1994), 507–513.
[63] K.P. Wong and C.C. Fung, *Simulated annealing based economic dispatch algorithm*, Proc. Inst. Electr. Eng., **140(C)** (1993), 509–515.
[64] H.T. Yang, P.C. Yang and C.L. Huang, *Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions*, IEEE Trans. Power Syst., **11(1)** (1996), 112–117.
[65] J. Zhang and A.C. Sanderson, *JADE: Adaptive differential evolution with optional external archive*, IEEE Trans. Evol. Comput., **13** (2009), 945–958.

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: akazad@dps.uminho.pt
E-mail address: emgpf@dps.uminho.pt