Adaptive range composite differential evolution for fast optimal reactive power dispatch

Niu, Ming; Xu, Ning Zhou; Dong, He Nan; Ge, Yang Yang; Liu, Yi Tao; Ngin, Hoon Tong

2021

Niu, M., Xu, N. Z., Dong, H. N., Ge, Y. Y., Liu, Y. T., & Ngin, H. T. (2021). Adaptive range composite differential evolution for fast optimal reactive power dispatch. IEEE Access, 9, 20117-20126. doi:10.1109/ACCESS.2021.3053640

https://hdl.handle.net/10356/146655

https://doi.org/10.1109/ACCESS.2021.3053640

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 13, 2021, accepted January 18, 2021, date of publication January 22, 2021, date of current version February 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053640

Adaptive Range Composite Differential Evolution for Fast Optimal Reactive Power Dispatch

MING NIU[1], NING ZHOU XU[2], (Member, IEEE), HE NAN DONG[1], YANG YANG GE[1], YI TAO LIU[1], AND HOON TONG NGIN[2]

[1]Electric Power Research Institute of State Grid Liaoning Electric Power Co., Ltd., Shenyang 110006, China
[2]Experimental Power Grid Center, Energy Research Institute @ NTU, Nanyang Technological University, Singapore 627590

Corresponding author: Ning Zhou Xu (11901711r@connect.polyu.hk)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1505400, and in part by the Energy Market Authority and National Science Foundation Singapore under Grant NRF2017EWT-EP003-038.

ABSTRACT This paper proposes a novel adaptive range composite differential evolution (ARCoDE) algorithm to efficiently and accurately solve optimal reactive power dispatch (ORPD) problem. Because of a novel adaptive range strategy for control parameters, the proposed ARCoDE possesses superior exploration and exploitation capabilities that can efficiently handle the ORPD problem involving complicated constraints and discrete and continuous variables. This has been demonstrated in case studies using the IEEE optimal power flow testbeds considering complex wind and demand scenarios. The superior performance of ARCoDE has been further validated through comparisons with several award-winning algorithms in 2014 IEEE Competition on ''Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems'', given limited iterations of in evolutionary optimization process.

INDEX TERMS Control parameter adaptation, differential evolution, optimal reactive power dispatch.

I. INTRODUCTION

The reactive power dispatch is critical to ensure the security and economy of power system operation. Similar to optimal power flow (OPF), optimal reactive power dispatch (ORPD) problem is a complicated mixed-integer non-linear optimization problem involving many constraints and discrete/ continuous decision variables [1]. Without assumptions such as convexity, differentiability and continuity, traditional techniques including linear programming, non-linear programming, and interior point method may not handle these problems well [2]-[4]. In addition, the performance of these methods is highly affected by the initial solution guess.

In view of the above issues, a variety of heuristic optimization algorithms (HOAs) have been proposed to solve OPF and ORPD problems [5], including e.g. genetic algorithm (GA), [6], [7], evolutionary programming (EP) [8], particle swarm optimization (PSO) [9], [10], differential evolution (DE) [11], seeker optimization [12], mean-variance mapping optimization (MVMO) [13], quantum-inspired evolutionary algorithm (QEA) [14], etc. In practice, fast OPF computation is needed for e.g. power flow management [15], reactive source control of wind farm [13], and so on. Therefore, it is practically valuable to develop highly efficient HOAs to fulfill practical operation needs. In addition, the OPF and ORPD formulations are becoming even more complicated due to integration of renewable energies, which also motivates applications of HOAs.

The major concerns of HOAs include the convergence speed and control parameters selection. The former can be enhanced by introducing exploitive recombination strategies, but the robustness (i.e. the method should obtain good solutions, in reasonable times and not too sensitive to changes in parameters) of the algorithm may be compromised accordingly. The latter can be handled by different adaptive or self-adaptive mechanisms to shorten the tedious trial-and-error procedure for fine tuning control parameters. However, these adaptive or self-adaptive strategies can still provide unsatisfactory parameters for practical applications, where only limited numbers of function evaluations are allowed due to the critical time requirement. For example, the active or reactive power dispatch in power system can be conducted in every 15 minutes, asking for a fast optimization solution. Considering the above concerns, this

The associate editor coordinating the review of this manuscript and approving it for publication was Jiayong Li.

paper proposes a novel adaptive range composite differential evolution (ARCoDE) algorithm targeting at practical applications such as the fast ORPD. The proposed ARCoDE utilizes the concept of compositing different types of trial vector generation strategies [16], which is able to provide a decent balance for the algorithm between the exploration and exploitation capabilities. In addition, a novel control parameter range adaptation mechanism is proposed to enable a highly efficient adaptive tuning of control parameters. These novel properties effectively support ARCoDE to conquer the difficulties introduced by the limited numbers of function evaluations due to the critical time requirements in many practical applications including ORPD problem.

The rest of this paper is organized as follows. Section II briefly introduces the basic formulation of ORPD problem. Section III reviews the state-of-the-art techniques of DE algorithms. The proposed ARCoDE algorithm is detailed in Section IV. Experimental results are reported in Section V. Finally, Section VI concludes this paper

II. ORPD PROBLEM FORMULATION

The ORPD problem is a classical but complicated mixed-integer non-linear optimization problem involving many constraints and discrete/continuous decision variables. In general, the application of HOAs to solve ORPD problems may be constrained by the relatively long convergence time. In addition, the fine-tuning of control parameters needs numerous amounts of trial-and-error tests. Moreover, due to the uncertainties introduced by the increased penetration of renewable energy generation including PV and wind power [17], the solutions provided by HOAs are less likely robust.

Typically, the ORPD problem aims at the minimization of the total active power losses in transmission networks as,

$$\begin{aligned} \mathit{Minimize}: f_Q &= \sum_{k\in(i,j)} P_{\mathrm{loss},k} \\ &= \sum_{i=1}^{L}\sum_{j=1}^{L}\left[g_{ij}\left(|V_i|^2+|V_j|^2-2|V_i||V_j|cos\left(\delta_i-\delta_j\right)\right)\right] \end{aligned} \tag{1}$$

where f_Q is the sum of active power loss and minimized in terms of decision variables x and u; $P_{\mathrm{loss},k}$ is the active power transmission loss for branch k; L is the number of transmission lines. x represents the control variables including voltages at generation buses V_{G}, reactive power compensation of the shunt capacitors and inductors Q_{C}, and transformer tap settings Q_{C},

$$x^\top = [V_{\mathrm{G}}, Q_{\mathrm{C}}, T] \tag{2}$$

and u denotes the dependent variables consisting of voltage of load buses V_{PQ}, generator reactive power outputs Q_{G} and transmission line flow S_{L}, defined as,

$$u^\top = [V_{\mathrm{L}}, Q_{\mathrm{G}}, S_{\mathrm{L}}] \tag{3}$$

Constraints including the equality and inequality ones are involved in ORPD problem. The equality constraints are mainly non-linear power flow equations for power balance, defined as,

$$P_{\mathrm{G}_i}-P_{\mathrm{D}_i}-P^{\mathrm{B}}_{i,\mathrm{loss}}=0, \quad Q_{\mathrm{G}_i}-Q_{\mathrm{D}_i}-Q^{\mathrm{B}}_{i,\mathrm{loss}}=0, \ i\in N_{\mathrm{B}} \tag{4}$$

where P_{G_i} and Q_{G_i} are the injected active and reactive power at bus i respectively, P_{D_i} and Q_{D_i} represent the active and reactive power demands at bus i respectively, $P^{\mathrm{B}}_{i,\mathrm{loss}}$ and $Q^{\mathrm{B}}_{i,\mathrm{loss}}$ are the active and reactive power losses at bus i respectively, and N_{B} is the total number of buses. The inequality constraints include,

$$V_i^{\min} \le V_i \le V_i^{\max}, \quad i\in N_{\mathrm{B}}, \tag{5}$$
$$Q_{\mathrm{G}_i}^{\min} \le Q_{\mathrm{G}_i} \le Q_{\mathrm{G}_i}^{\max}, \quad i\in N_{\mathrm{G}}, \tag{6}$$
$$Q_{\mathrm{C}_i}^{\min} \le Q_{\mathrm{C}_i} \le Q_{\mathrm{C}_i}^{\max}, i\in N_{\mathrm{C}}, \tag{7}$$
$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i\in N_{\mathrm{T}}, \tag{8}$$
$$S_{\mathrm{L}_i} \le S_{\mathrm{L}_i}^{\max}, \quad i\in N_{\mathrm{B}}, \tag{9}$$

where constraints (5)—(9) define the limits for bus voltage V_i, reactive power generation Q_{G_i}, reactive power compensation Q_{C_i}, transformer tap position T_i and branch power flow S_{L_i}, respectively; N_{G}, N_{C}, and N_{T} denote the total number of generator buses, shunt capacitors and inductors installation buses, and transformer taps respectively.

Traditionally, the ORPD problem can be solved by classical optimization algorithms such as linear programming, non-linear programming, and interior point method, however, the results obtained can hardly become the genuine global optimum due to the involved problem simplifications. In contrast, HOAs can be applied directly to solve the original formulations without simplifications. In addition, HOAs are the only category of methods that are capable of global optimal search for mixed integer non-differentiable and non-convex problems. However, the concerns of solution robustness and computation time limit their practical applicability, and these concerns are particularly attended in developing the proposed ARCoDE algorithm.

III. OVERVIEW OF DIFFERENTIAL EVOLUTION

The DE algorithm is a stochastic population-based optimization algorithm for real-valued parameters and functions. The core of DE is a scheme for generating trial solution vectors by weighing the difference vector between two population members and then adding that to a third member. If the resulting vector yields a smaller objective value than its target vector, it will be prioritized in the evolutionary optimization process. The algorithm comprises the following four steps: initialization, mutation, crossover, and selection.

Initialization: DE begins with a randomly initiated population of NP D-dimensional real-valued vectors. The below notation represents the ith vector of the population at generation G:

$$x = \left[x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,D}\right] \tag{10}$$

Since each variable may have a certain range, the jth component of the ith vector can be initialized as

$$x_{i,j} = x_j^{\min} + rand_{i,j}[0, 1] \cdot (x_j^{\max} - x_j^{\min}) \tag{11}$$

Mutation: The DE mutation maintains the population diversity and provides information necessary to steer the optimization. One of the simplest DE mutation operators generates a mutated vector for each target vector X_i of generation G, according to

$$V_i = X_{r_1} + F \cdot (X_{r_2} - X_{r_3}), \tag{12}$$

where r_1, r_2, and r_3 are distinct integers randomly chosen from $[1, NP]$ and different from i, F is the mutation constant controlling the amplification of the differential variation.

Crossover: To enhance the population diversity, the mutated vector is then mixed with a predetermined target vector to form the so-called trial vector, often referred to as crossover. Specifically, the trial vector is formed as follows:

$$U_i = [u_{i,1}, u_{i,2}, \cdots, u_{i,D}], \tag{13}$$

where

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_{i,j}[0, 1] \le C_R \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases} \tag{14}$$

and j_{rand} is an integer randomly chosen from $[1, 2, \cdots, D]$ to ensure $u_{i,j}$ receive at least one component from $v_{i,j}$. The crossover constant C_R controls the population diversity.

Selection: This step determines whether the target or the trial vector survive to the next generation according to,

$$X_i^{G+1} = \begin{cases} U_i^G & \text{if } f(U_i^G) \le f(X_i^G) \\ X_i^G & \text{otherwise,} \end{cases} \tag{15}$$

where f is the objective to be minimized. Equation (15) ensures the fitness of the population can either improve or remain the same, but never deteriorate.

Recognizing that DE performance depends on its trial vector generation strategies and control parameter settings, many DE variants have been proposed. With respect to the trial vector generation strategies, Fan and Lampinen [18] propose a trigonometric mutation operator to accelerate the DE convergence. Their mutation operator enhances the local search, since it moves the new trial vector toward the best one of three individuals chosen for mutation. Mezura-Montes *et al.* [19] propose a novel mutation operator named "current-to-best/1", which incorporates the information of the best solution in the current population and the current parent to create a new trial vector. Feoktistov and Janaqi [20] classify mutation operators into four categories according to the way they use the objective function values. It has been observed that "current-to-best/1" strategy performs poorly on exploring the search space when solving multimodal problems. Recently, much effort has been made to improve the performance of this strategy. Das *et al.* [21] improve the "current-to-best/1" strategy by introducing a local neighbourhood model, in which each vector is mutated by using the best individual solution found so far in its small neighbourhood. Zhang and Sanderson [22] propose the "current-to-pbest/1" strategy. Instead of only adopting the best individual in the "current-to-best/1" strategy, their strategy also utilizes the information of other good solutions. Moreover, the recently generated inferior solutions are incorporated in this strategy. Yong *et al.* [16] study the DE performance of combine several effective trial vector generation strategies with some suitable control parameter settings, and propose a composite DE (CoDE) to randomly combine three selected recombination strategies and three control parameters settings to generate trial vectors.

Many attempts have also been made to improve the DE convergence speed and robustness of solutions by tuning the control parameters such as NP, F, and C_R. Storn and Price [23] argue that the three parameters are not difficult to set for good performance, suggesting $NP \in [5D, 10D]$, F should be 0.5 as a good initial choice and $F < 0.4$ or $F > 1.0$ will lead to performance degradation, and C_R can be set to 0.1 or 0.9. In contrast, in [24], it is shown that DE performance is very sensitive to control parameters and suggests $NP \in [3D, 8D]$, It is proved that F should not be smaller than a problem-dependent threshold in order to prevent premature convergence, and if $F > 1.0$, the convergence speed will decrease. Therefore, a good initial F be 0.6 and $C_R \in [0.3, 0.9]$ is suggested. Despite many different suggestions for control parameters, consensus has been reached that $F \in [0.4, 1.0]$, and C_R should be either close to 1.0 or 0.0 depending on the characteristics of problems.

Some smart adaptive strategies are developed to best tune the control parameters during DE evolution. Two schemes are introduced to adapt F, with one scheme varying F randomly, and the other linearly reducing F from a predefined maximal value [21]. A self-adaptive DE (jDE) is proposed in [25], where both F and Cr are randomly tuned according to certain probabilities. An adaptive differential evolution with optional external archive (JADE) (proposed by Zhang and Sanderson [22], utilizes normal and Cauchy distributions to generate F and C_R for each target vector, respectively. In addition, JADE makes use of recent successful F and C_R for generating new ones. Unlike the above methods, self-adaptive differential evolution (SaDE), proposed in [26], adaptively adjusts its trial vector generation strategies and control parameters simultaneously by learning from the previous search.

IV. ADAPTIVE RANGE COMPOSITE DE FOR ORPD

As reviewed in Section III, different generation strategies for trial vector and the control parameter tuning have been extensively investigated. However, those methods are still found unpromising when dealing with the practical ORPD problem that demands fast speed as well as solution robustness. This is mainly due to that ORPD in practice are conducted in short time intervals just allowing limited numbers of function evaluations for those strategies to well adapt the trial vector and control parameters. As such, this paper proposes

TABLE 1. Pseudo-code of ARCoDE.

Input: *NP*: the number of candidates for each generation.
Max_FES: maximum number of function evaluations.
The strategy candidate pool: "DE/best/2/bin" and "DE/rand/2/bin.
Initial ranges of the control parameters: $F^{er} = [0.7, 0.9]$; $F^{et} = [0.5, 0.7]$; $C_r^{er} = [0.8, 1]$; $C_r^{et} = [0, 0.2]$.
Initial probabilities of each range: $P_{F/Cr}^{er} = P_{F/Cr}^{et} = 0.5$.
(1) $G = 0$; Initialize the candidate population $P_0 = \{\vec{X}_{1,0}, \cdots, \vec{X}_{NP,0}\}$ by uniformly sampling within the feasible region;
(2) Calculate the objective values for all the candidates $f(\vec{X}_{1,0}), \cdots, f(\vec{X}_{NP,0})$;
Evaluate the constraint violation $g(\vec{X}_{1,0}), \cdots, g(\vec{X}_{NP,0})$;
(3) $FES = NP$;
(4) **while** $FES < Max\_FES$ do
(5) $P_{G+1} = \phi$;
(6) **for** $i = 1{:}NP$ do
(7) Calculate parameter range probabilities $P_{F/Cr}^{er/et}$ using equation (4.14) and update the success and fail memory. Then apply Roulette Wheel selection to select the ranges;
(8) Generate two candidate vectors $\vec{u}_{i\_1,G}$ and $\vec{u}_{i\_2,G}$ for the target $\vec{X}_{1,G}$ based on the two candidate vector generation mechanisms with control parameters determined according to the ranges obtained by Step (7);
(9) Calculate the objective values and the constraint violation values of the two candidates $\vec{u}_{i\_1,G}$ and $\vec{u}_{i\_2,G}$;
(10) Choose the best trial vector from the two trial vectors $\vec{u}_{i\_1,G}$ and $\vec{u}_{i\_2,G}$, and the target vector $\vec{X}_{1,G}$ according to Deb's selection criterion;
(11) $FES = FES + 2$;
(12) **end for**
(13) $G = G + 1$;
(14) **end while**
Output: the candidate with the best objective function value or the smallest constraint violation value in the population

a novel ARCoDE method aiming at efficiently solve fast ORPD problem. The primary idea is to randomly combine two trial vector generation strategies with two adaptive ranges of control parameters at each generation in creating new trial vectors.

Different from the original CoDE, the new algorithm features a faster convergence while remains a consistent population diversity during evolutions. To allow fast convergence given limited function evaluations, two trail vector generation strategies including DE/best/2/bin and DE/rand/2/bin [23] are employed in the study. Instead of setting specific values for control parameters, explorative and exploitative ranges for F and C_R respectively are innovatively introduced in the proposed algorithm. During the evolution process, the algorithm gradually adapts the chosen probabilities and the sizes of these ranges. At each generation, each trial vector generation strategy from the strategy pool is used to create a new trial vector with the algorithm control parameters chosen from the ranges of F and C_R according to its feedback probability. Accordingly, two trial vectors are generated for each target vector. Then the best one enters the next generation if it is better than its target vector. Table 1 presents the pseudo-code of ARCoDE, details of which are illustrated below.

A. TRIAL VECTOR GENERATION STRATEGIES

To allow fast convergence, the greedy combination strategies that benefit from their fast convergence by incorporating the best solution information in the evolutionary search are considered. Consequently, the DE/best/2/bin is selected as one of the candidate recombination strategies in the pool. However, such strategy may easily lead the evolutionary of the population to local optimum. To prevent the premature convergence by such a greedy strategy, the DE/rand/2/bin strategy is also included to further enhance the explorative capability of the proposed algorithm. In the DE/rand/2/bin, two difference vectors are added to the base vector, which might lead to better perturbation than the strategies with only one difference vector e.g. DE/rand/1/bin.

B. CONTROL PARAMETER ADAPTATION

As discussed above, solving ORPD problems in practice may need limited function evaluations for fast convergence. The existing adaptive strategies e.g. jDE, JADE, and SaDE can hardly guarantee to gauge suitable values for F and C_R in such a small number of function evaluations. Instead, it is more promising to evolve suitable ranges for these control parameters during the iteration process. In general, a large F can make the mutant vectors distributed widely in the search space and can increase the population diversity. In contrast, a small F makes the search focus on neighborhoods of the current solutions, thus speeding up the convergence. On the other side, a large C_R makes the trial vector very different from the target vector, since the trial vector will inherit little information from the target vector. Consequently, the diversity of the offspring population can be maintained. A small C_R is very suitable for separable problems, since in such case the trial vector may be different from the target by only one component.

Motivated by the above observations, explorative and exploitative ranges for F and C_R, respectively are defined in this paper based on the characteristic of these control parameters, defined as follow,

$$\text{Explorative range of } F: \; F^{er} = [0.7, 0.9], \tag{16}$$

$$\text{Exploitative range of } F: \; F^{et} = [0.5, 0.7], \tag{17}$$

$$\text{Explorative range of } C_r: \; C_r^{er} = [0.8, 1], \text{ and} \tag{18}$$

$$\text{Exploitative range of } C_r: \; C_r^{et} = [0, 0.2]. \tag{19}$$

The initial probabilities of applying different range to each individual are set to 0.5, i.e. $P_{F/C_r}^{er/et} = 0.5$. Therefore, each range has equal probability to be applied to every individual in the initial population. According to the probability, Roulette Wheel selection is applied to select the range for each individual in the current population. Thereafter, an F/C_r value will be randomly selected within this range and assigned to the corresponding individual. After evaluation of all newly generated trial vectors, the number of trial vectors generated by different ranges while successfully entering the next generation is recorded as $NS_{F/C_r}^{er/et}$ and the numbers of trial vectors discarded is recorded as $NS_{F/C_r}^{er/et}$. These numbers are accumulated within a specified number of generations, call the learning period. Then the probability is updated as:

$$P_{F/C_r}^{er/et} = \frac{NS_{F/C_r}^{er/et}}{NS_{F/C_r}^{er/et} + NF_{F/C_r}^{er/et}} \tag{20}$$

which represents the percentage of the success rate of trial vectors generated by each range during the learning period. Therefore, the probabilities of applying each range are updated every generation, after the learning period. When the evolutionary process reaches a pre-defined set point, the range with lower percentage of success rate is discarded and the range with higher percentage of success rate will be split in half, and then the above operation is repeated with this new range. This adaptation procedure is capable to gradually evolve suitable ranges for F and C_R within a relatively small number of function evaluations.

C. CONSTRAINT HANDLING

In this work, the constraint handling methods are applied based on superiority of feasible solutions proposed by Deb [27]. Deb's selection criterion has no parameter to fine-tune, which is one of the main motivation of our work—no fine-tuning of parameters as much as possible. Hence, this constraints handling technique is incorporated as follows:

During the selection procedure, the vector A is compared to vector B in the current population considering both the objective value and constraint violations. Vector A will replace vector B and enter the population of the next generation if any of the following conditions is true.

1) Vector A is feasible and vector B is not.
2) Vectors A and B are both feasible and vector A has smaller objective value than vector B.
3) Vectors A and B are both infeasible, but vector A has a smaller overall constraint violation.

V. NUMERICAL RESULTS

The 41-bus offshore wind power plant (WPP) ORPD test case in the 2014 IEEE Competition on "Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problem" is used to study the performance of the proposed ARCoDE. This test case consists of 18 continuous variables associated to wind generator reactive power set-points, 2 discrete variables associated to stepwise adjustable on-load transformers' tap position, a discrete variable defining the stepwise adjustment of a regulated capacitor, and a continuous variable defining the adjustment of reactor. In the fast ORPD problem, the reactive power requirements corresponding to the actual operating condition are defined as stepwise changes of reactive power requirements (qref) results in 96 scenarios, some of which turn out to be hard-to-solve optimization tasks. According to (1)—(9), the target of the problem is to minimize the total active power transmission losses while fulfilling constraints, given limited numbers of function evaluations. The WPP ORPD problem contains 96 scenarios, among which the 13 most challenging scenarios are selected for the test purpose in this paper. These 13 scenarios are particularly selected since their feasible solutions can hardly be found by those award-winning algorithms in the competition. The topology of the offshore WPP system is shown in Fig. 1, and a detailed description of test cases and the competition results can be found in [28].

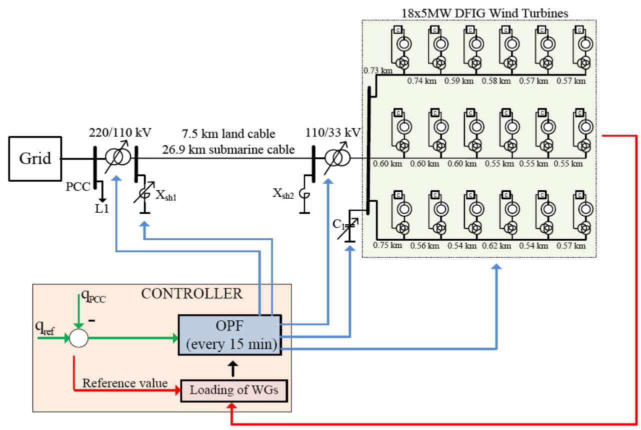

FIGURE 1. The 41-bus offshore wind power plant (WPP) ORPD test case [13], [28].

A. NUMERICAL RESULTS OF ARCODE ON THE TEST CASES

The test environment is a DELL Desktop Workstation with Intel (R) Xeon (R) CPU E5-2650 v2 @2.60GHz RAM 64GB. Table 2 shows the numerical results (including the best solutions, objective values, the sum of constraint violations, and the average computation time through 31 trials) obtained by the proposed ARCoDE algorithm for the 13 test cases, where WGi_Q represent the wind generator reactive power set-points, and $i = 1, 2, \cdots, 18$; OLTC_T$_j$ represent the tap position of stepwise adjustable on-load transformers, and $j = 1, 2$; C_1 represents the stepwise adjustment of capacitor; Xsh1 represents the adjustment of reactor; obj_best represents the best objective value (power loss) according to (1) obtained by ARCoDE, and gvar_best represents the best sum of different constraint violations. It is observed that the computation of each scenario is fairly fast for about 60s, reflected by the average computational time.

B. COMPARISON WITH THE AWARD-WINNING ALGORITHMS IN THE COMPETITION

The mean and standard deviation of fitness values (evaluated by the benchmark program provided by [28]) from ARCoDE are compared with those from the top 3 ranking algorithms in the IEEE Competition on "Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problem", i.e. improved $(\mu + \lambda)$-constrained differential evolution (ICDE) [29], differential evolution particle swarm optimization (DEEPSO) [30] and MVMO. The number of function evaluations in all these methods is set to 10000 according to the competition rules. Such a short

TABLE 2. Best Results Obtained by ARCoDE on the 13 Test Scenarios.

Scenario	50	51	52	53	54	55	56	75	76	77	78	79	80
WG1_Q (MVar)	1.644	1.633	1.650	1.650	1.650	1.650	1.543	-1.546	-1.650	-1.650	-1.650	-1.650	-1.650
WG2_Q (MVar)	1.647	1.647	1.650	1.649	1.650	1.650	1.637	-1.604	-1.650	-1.650	-1.650	-1.650	-1.650
WG3_Q (MVar)	1.519	1.648	1.650	1.650	1.650	1.650	1.611	-0.891	-1.650	-1.650	-1.650	-1.650	-1.650
WG4_Q (MVar)	1.616	1.601	1.650	1.650	1.650	1.650	1.645	-0.816	-1.650	-1.650	-1.650	-1.650	-1.650
WG5_Q (MVar)	1.594	1.643	1.650	1.650	1.650	1.650	1.648	-0.601	-1.650	-1.650	-1.650	-1.650	-1.650
WG6_Q (MVar)	1.643	1.571	1.650	1.650	1.650	1.650	1.641	-0.394	-1.650	-1.650	-1.650	-1.650	-1.650
WG7_Q (MVar)	1.616	1.650	1.650	1.650	1.650	1.650	1.607	-1.636	-1.650	-1.650	-1.650	-1.650	-1.650
WG8_Q (MVar)	1.645	1.650	1.650	1.650	1.650	1.650	1.626	-1.222	-1.650	-1.650	-1.650	-1.650	-1.650
WG9_Q (MVar)	1.630	1.618	1.650	1.650	1.650	1.650	1.633	-1.543	-1.650	-1.650	-1.650	-1.650	-1.650
WG10_Q (MVar)	1.626	1.649	1.650	1.650	1.650	1.650	1.603	-1.353	-1.650	-1.650	-1.650	-1.650	-1.650
WG11_Q (MVar)	1.612	1.639	1.650	1.650	1.650	1.650	1.620	-1.141	-1.650	-1.650	-1.650	-1.650	-1.650
WG12_Q (MVar)	1.628	1.633	1.650	1.650	1.650	1.650	1.555	-1.100	-1.650	-1.650	-1.650	-1.650	-1.650
WG13_Q (MVar)	1.615	1.645	1.650	1.650	1.650	1.650	1.640	-0.907	-1.650	-1.650	-1.650	-1.650	-1.650
WG14_Q (MVar)	1.604	1.646	1.650	1.650	1.650	1.650	1.626	-1.541	-1.650	-1.650	-1.650	-1.650	-1.650
WG15_Q (MVar)	1.594	1.638	1.650	1.650	1.650	1.650	1.618	-1.507	-1.650	-1.650	-1.650	-1.650	-1.650
WG16_Q (MVar)	1.575	1.610	1.650	1.650	1.650	1.650	1.640	-1.378	-1.650	-1.650	-1.650	-1.650	-1.650
WG17_Q (MVar)	1.625	1.627	1.650	1.650	1.650	1.650	1.586	0.759	-1.650	-1.650	-1.650	-1.650	-1.650
WG18_Q (MVar)	1.606	1.643	1.650	1.650	1.650	1.650	1.567	-0.854	-1.650	-1.650	-1.650	-1.650	-1.650
OLTC_T1	1.0993	1.0993	1.0993	1.0993	1.0993	1.0993	1.0993	0.9669	0.9669	0.9669	0.9669	0.9669	0.9669
OLTC_T2	0.8700	0.8700	0.8700	0.8700	0.8700	0.8700	0.8700	1.1083	1.1300	1.1300	1.1300	1.1300	1.1300
C_1 (MVar)	-12.100	-12.100	-12.100	-12.100	-12.100	-12.100	-12.100	-4.033	-4.033	-4.033	-4.033	-4.033	-4.033
X_{sh1} (MVar)	0	0	0	0	0	0	0	9.8965	9.8965	9.8965	9.8965	9.8965	9.8965
obj_best (MW)	1.433	1.379	1.280	1.216	1.258	1.261	1.440	2.011	2.637	2.637	2.637	2.637	2.637
gvar_best	0	0	0	0	0	0	0	0	0	0	0	0	0
Computation time* (s)	64.128	63.910	64.522	64.085	64.294	65.230	66.425	55.990	66.266	65.469	63.703	63.221	66.484

*: The average value through 31 running trials

function evaluations makes it extremely difficult to find feasible solutions of the complicated ORPD problem for HOAs.

The experimental results are given in Table 3. All the results are obtained from 31 independent trials. The last three rows of Table 3 summarize the experimental results. On these 13 test scenarios, ARCoDE performs significantly better than ICDE, which is developed by the authors previously. The performance of ARCoDE and DEEPSO are quite similar. For test scenarios 76 and 79, MVMO shows a better performance in terms of optimality and robustness than the other 3 competitors. However, it should be noted that the proposed ARCoDE does not need a fine-tuning pre-defined control parameter setting, which indicates a significant advantage over the other three algorithms. In Table 3, "$-$","$+$", and "$\sim$" denote that the performance of the corresponding algorithm is better than, worsen than, and similar to that of ARCoDE. E.g. MVMO wins ARCoDE in 3 scenarios, but loses in 1. They perform similarly in the rest 9 scenarios out of the total 13. Notably, ARCoDE needs no local search operator that can add computational complexities and is found existing in DEEPSO and MVMO. In terms of the constraint handling, ARCoDE is much simpler than ICDE, which utilizes the concept of multi-objective optimization. The preference of feasible solutions makes the constraint handling part of ARCoDE less computationally complex.

In summary, the proposed ARCoDE is better, or at least no worse than the three competitors. The evolution of the mean fitness values derived from ICDE, DEEPSO, MVMO, and ARCoDE versus the number of function evaluation is plotted in Fig. 2 for some typical test scenarios.

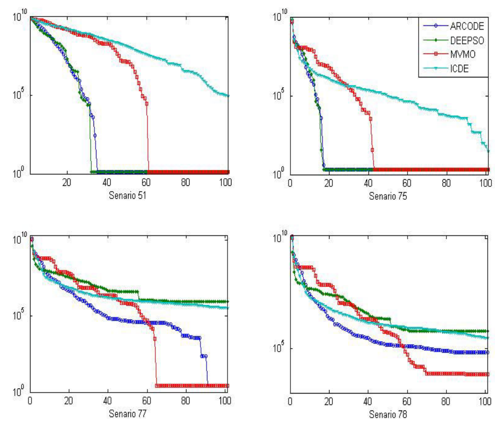

FIGURE 2. Evolution of the mean fitness value derived from modified ICDE, DEEPSO, MVMO, and ARCoDE versus the number of FES on four test cases.

C. COMPARISON WITH THREE STATE-OF-THE ART DE ALGORITHMS

To demonstrate the effectiveness of the novel proposed adaptive range of control parameter settings, ARCoDE is also compared with three other state-of-the-art adaptive DEs, i.e., jDE, JADE, and SaDE. In jDE, JADE, and SaDE, the control parameters F and Cr are self-adapted during the evolution. In our experiments, the same parameter setting is used for these three methods as in their original papers. The number of function evaluations in all these is 10000, and each method is executed 31 times on each test cases. Table 4 summarizes the experimental results.

TABLE 3. Experiment Results of ICDE, DEEPSO, MVMO and ARCoDE Over 31 Independent Trials.

Scenario	ICDE Mean (Std)	DEEPSO Mean (Std)	MVMO Mean (Std)	ARCoDE Mean (Std)
50	2.428E+04(4.567E+04)	1.422E+00(0.004E+00)	1.430E+00(0.016E+00)	1.437E+00(0.003E+00)
51	9.079E+04(1.096E+05)	1.373E+00(6.384E-04)	1.380E+00(0.010E+00)	1.382E+00(0.003E+00)
52	3.760E+05(2.803E+05)	1.280E+00(3.920E-05)	1.280E+00(5.043E-05)	1.280E+00(7.112E-06)
53	4.482E+05(2.999E+05)	1.216E+00(5.257E-05)	1.216E+00(7.427E-05)	1.216E+00(8.466E-06)
54	3.770E+05(2.412E+05)	1.258E+00(2.251E-05)	1.258E+00(2.719E-05)	1.258E+00(3.225E-06)
55	2.944E+05(1.089E+05)	1.261E+00(4.576E-05)	1.261E+00(4.534E-05)	1.261E+00(5.289E-06)
56	9.737E+03(2.293E+04)	1.428E+00(0.003E+00)	1.435E+00(0.016E+00)	1.445E+00(0.002E+00)
75	2.968E+01(1.471E+02)	7.550E+01(4.092E+02)	2.022E+00(0.003E+00)	2.0116E+00(5.296E-04)
76	3.895E+05(2.278E+05)	1.270E+06(5.761E+06)	2.637E+00(3.740E-08)	4.444E+05(2.328E+06)
77	3.281E+05(1.986E+05)	9.233E+05(5.118E+06)	2.637E+00(2.497E-08)	2.637E+00(1.796E-08)
78	3.186E+05(1.941E+05)	6.464E+05(3.600E+06)	7.086E+03(3.944E+04)	7.016E+04(3.841E+05)
79	2.937E+05(1.696E+05)	2.008E+04(1.118E+05)	2.637E+00(3.005E-08)	5.776E+05(3.002E+06)
80	3.341E+05(1.841E+05)	2.637E+00(7.478E-08)	2.637E+00(3.005E-08)	2.637E+00(1.356E-05)
-	0	3	3	N.A.
+	13	2	1	N.A.
~	0	8	9	N.A.

"Mean" and "Std" indicate the average and standard deviation of the function fitness values obtained in 31 runs, respectively. "-","+", and "~" denote that the performance of the corresponding algorithm is better than, worsen than, and similar to that of ARCoDE, respectively.

TABLE 4. Experimental Results of jDE, JADE, SaDE and ARCoDE Over 31 Independent Trials.

Senario	jDE Mean (Std)	JADE Mean (Std)	SADE Mean (Std)	Adaptive range CODE Mean (Std)
50	3.579E+09(9.754E+08)	1.450E+03(1.987E+03)	9.291E+02(1.747E+03)	1.437E+00(0.003E+00)
51	4.299E+09(1.083E+09)	6.951E+05(1.200E+06)	2.044E+06(3.206E+06)	1.382E+00(0.003E+00)
52	4.777E+09(1.056E+09)	4.755E+06(90.96E+06)	9.533E+06(9.080E+06)	1.280E+00(7.112E-06)
53	4.761E+09(1.222E+09)	3.338E+06(7.740E+06)	6.235E+06(7.514E+06)	1.216E+00(8.466E-06)
54	5.103E+09(9.391E+08)	6.839E+06(1.012E+07)	1.011E+07(1.111E+07)	1.258E+00(3.225E-06)
55	4.665E+09(9.477E+08)	5.051E+06(9.511E+06)	5.490E+06(5.254E+06)	1.261E+00(5.289E-06)
56	3.725E+09(1.258E+09)	1.273E+03(1.743E+03)	1.039E+05(5.726E+05)	1.445E+00(0.002E+00)
75	1.174E+07(7.458E+06)	2.012E+00(3.834E-04)	2.012E+00(2.719E-04)	2.0116E+00(5.296E-04)
76	4.436E+07(5.195E+07)	2.402E+07(2.580E+07)	1.772E+07(1.498E+07)	4.444E+05(2.328E+06)
77	3.846E+07(1.541E+07)	1.595E+07(2.488E+07)	2.150E+07(2.570E+07)	2.637E+00(1.796E-08)
78	4.109E+07(2.084E+07)	1.981E+07(2.163E+07)	1.975E+07(2.334E+07)	7.016E+04(3.841E+05)
79	3.891E+07(2.575E+07)	2.005E+07(2.579E+07)	2.262E+07(3.546E+07)	5.776E+05(3.002E+06)
80	3.533E+07(1.624E+07)	1.809E+07(2.496E+07)	2.185E+07(2.537E+07)	2.637E+00(1.356E-05)
-	0	0	0	N.A.
+	13	12	12	N.A.
~	0	1	1	N.A.

"Mean" and "Std" indicate the average and standard deviation of the function fitness values obtained in 31 runs, respectively. "-","+", and "~" denote that the performance of the corresponding algorithm is better than, worsen than, and similar to that of ARCoDE, respectively.

Overall, ARCoDE significantly outperforms jDE, JADE, and SaDE in terms of the optimality and robustness of solutions. As the assumption described in Section IV, when dealing with practical ORPD kind of problems, these adaptive strategies may not guarantee to evolve to a satisfied control parameter setting within a limited numbers of function evaluations. However, by relaxing the search criterion of control parameters from a specific value to a range, ARCoDE presents a promising application for fast ORPD problems. The evolution of the mean fitness values derived from modified jDE, JADE, SaDE, and ARCoDE versus the number of function evaluation is plotted in Fig. 3 for some typical test scenarios.

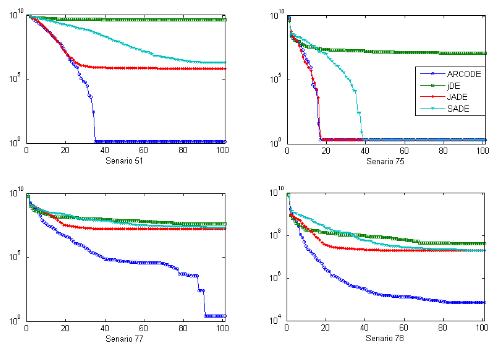

FIGURE 3. Evolution of the mean fitness value derived from jDE, JADE, SADE, and ARCoDE versus the number of FES on four test cases.

D. COMPARISON WITH FOUR VARIANTS OF CODE

The proposed ARCoDE is also compared with four variants of CODE with different trial vector generation strategies. For each variant, we attempt to provide a good balance between their exploration and exploitation ability by merging the greedy generation strategies (i.e., DE/best/2/bin, DE/current-best) and explorative-oriented generation strategies

TABLE 5. Experimental Results of Four Variants of CODE and ARCoDE Over 31 Independent Trials.

Senario	Original CODE Mean (Std)	CODE (Best/2/, rand/2/, current-rand) Mean (Std)	CODE (Rand/2/, current-best) Mean (Std)	CODE (Best/2/, current-rand) Mean (Std)	Adaptive range CODE (best/2/, rand/2/) Mean (Std)
50	8.561E+06(2.107E+07)	2.203E+06(8.913E+06)	2.849E+07(6.614E+07)	7.144E+06(1.680E+07)	1.437E+00(0.003E+00)
51	2.844E+07(5.167E+07)	1.344E+07(3.028E+07)	1.943E+07(4.309E+07)	3.463E+07(8.172E+07)	1.382E+00(0.003E+00)
52	3.438E+07(8.116E+07)	2.540E+07(4.201E+07)	2.878E+07(4.536E+07)	4.472E+07(8.200E+07)	1.280E+00(7.112E-06)
53	5.258E+07(1.085E+08)	1.967E+07(1.944E+07)	1.577E+08(1.763E+08)	3.723E+07(8.293E+07)	1.216E+00(8.466E-06)
54	3.955E+07(5.350E+07)	3.654E+07(8.163E+07)	8.834E+07(1.471E+08)	3.834E+07(8.164E+07)	1.258E+00(3.225E-06)
55	4.816E+07(6.290E+07)	2.953E+07(8.284E+07)	3.790E+07(5.516E+07)	3.223E+07(4.977E+07)	1.261E+00(5.289E-06)
56	9.971E+06(.3759E+07)	1.067E+07(3.714E+07)	3.056E+07(7.468E+07)	2.033E+06(8.237E+06)	1.445E+00(0.002E+00)
75	1.032E+06(5.749E+06)	1.726E+06(9.612E+06)	2.052E+06(9.303E+06)	7.860E+04(4.374E+05)	2.0116E+00(5.296E-04)
76	1.260E+08(1.531E+08)	1.497E+08(2.562E+08)	2.175E+08(2.430E+08)	1.293E+08(1.567E+08)	4.444E+05(2.328E+06)
77	2.286E+08(4.301E+08)	1.170E+08(1.413E+08)	2.187E+08(3.059E+08)	1.151E+08(1.389E+08)	2.637E+00(1.796E-08)
78	1.724E+08(2.848E+08)	1.405E+08(1.577E+08)	2.452E+08(2.407E+08)	1.399E+08(1.724E+08)	7.016E+04(3.841E+05)
79	1.253E+08(2.418E+08)	1.698E+08(2.839E+08)	2.126E+08(2.676E+08)	6.824E+07(8.234E+07)	5.776E+05(3.002E+06)
80	1.746E+08(2.053E+08)	1.551E+08(2.293E+08)	2.348E+08(3.178E+08)	1.650E+08(2.473E+08)	2.637E+00(1.356E-05)
-	0	0	0	0	N.A.
+	13	13	13	13	N.A.
~	0	0	0	0	N.A.

"Mean" and "Std" indicate the average and standard deviation of the function fitness values obtained in 31 runs, respectively. "-","+", and "~" denote that the performance of the corresponding algorithm is better than, worsen than, and similar to that of ARCoDE, respectively.

TABLE 6. Comparison of ARCoDE with Respect to ICDE, DEEPSO, MVMO, jDE, JADE, and SaDE in Terms of Feasible Rate.

Scenario	Feasible rate (%)						
	ICDE	DEEPSO	MVMO	jDE	JADE	SADE	ARCODE
50	25.8	100	100	0	64.5	77.4	100
51	6.5	100	100	0	74.2	58.1	100
52	0	100	100	0	77.4	0	100
53	0	100	100	0	83.9	0	100
54	0	100	100	0	67.7	0	100
55	0	100	100	0	77.4	0	100
56	32.3	100	100	0	64.5	67.7	100
75	93.5	96.8	100	100	100	100	100
76	0	87.1	100	0	29.0	0	93.5
77	0	93.5	100	0	54.8	0	100
78	0	96.8	96.8	0	38.7	0	90.3
79	0	96.8	100	0	35.5	0	87.1
80	0	100	100	0	35.5	0	100

(i.e., DE/rand/2/bin, DE/current-rand). 31 trials were carried out on 13 test scenarios. The number of function evaluations in all these methods is set to 10000. As shown in Table 5, the proposed ARCoDE shows its significant advantages over the typical CODE so as its variants in terms of the optimality and the robustness of solutions. The experimental results demonstrates once again the better applicability of ARCoDE for OPRD problems.

E. COMPARISON IN TERMS OF FEASIBLE RATE

As described in Section II, the ORPD problem involves a lot of equality and inequality constraints. Therefore, it is crucial for HOAs to provide stable feasible solutions i.e. candidate solutions fulfilling all constraints, when dealing with the ORPD problem. In terms of rate of feasible solutions, ARCoDE is compared against 6 methods as shown in Table 6. For all the test scenarios, the feasible rates are obtained through 31 trials. It can be seen from Table 6, ARCoDE gets 100% feasible rate for 10 of the 13 test scenarios. The feasible rate performance of ARCoDE is only worse than MVMO for scenario #76, 78, and 79, and superior to the other methods for all scenarios. It should be pointed out that ARCoDE does not need a fine-tuned control parameter setting for trial vector generations. In addition, the constraint handling part of ARCoDE involves no pre-defined parameter settings. These characteristics indeed enables the ease of its application in many practical problems.

In summary, the overall performance of ARCoDE is highly competitive with the six methods compared. It is therefore convinced that the proposed ARCoDE can provide superior optimization performance for ORPD problems.

VI. CONCLUSION

Many attempts in using HOAs for solving ORPD problems have been reported in the literature. These experiences reveal that for solving fast OPRD problem, a faster convergence and the robustness of solution should be provided by a well-designed HOAs. The novel ARCoDE algorithm, proposed in this paper, represents one of the first attempts along this direction. It employs two trail vector generation strategies and a novel control parameter range adaptation strategy. The structure of ARCoDE is simpler and it is easier to implement.

The experimental studies in this paper are carried out on the benchmark test cases of IEEE Competition on ''Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems''. The proposed ARCoDE is compared with the three award winning algorithms and three state-of-the-art adaptive DE algorithms. The experimental results demonstrate the overall performance of ARCoDE is better, or at least no worse than the other award winning algorithms. In addition, the effectiveness of the combination of the selected trial vector generation strategies and the novel proposed adaptive range of control parameters is experimentally studied. The experimental results show that the fast convergence rate and the robustness of ARCoDE make it promising for solving fast ORPD problems.

REFERENCES

[1] C. M. Huang, S. J. Chen, Y. C. Huang, and H. T. Yang, "Comparative study of evolutionary computation methods for active-reactive power dispatch," *IET Gener., Transmiss. Distrib.*, vol. 6, no. 7, pp. 636–645, Jul. 2012.

[2] N. Deeb and S. M. Shahidehpour, "Linear reactive power optimization in a large power network using the decomposition approach," *IEEE Trans. Power Syst.*, vol. 5, no. 2, pp. 428–438, May 1990.

[3] K. Y. Lee, Y. M. Park, and J. L. Ortiz, "A united approach to optimal real and reactive power dispatch," *IEEE Power Eng. Rev.*, vol. PER-5, no. 5, pp. 42–43, May 1985.

[4] K. C. Almeida and F. S. Senna, "Optimal active-reactive power dispatch under competition via bilevel programming," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2345–2354, Nov. 2011.

[5] M. Niu, C. Wan, and Z. Xu, "A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems," *J. Modern Power Syst. Clean Energy*, vol. 2, no. 4, pp. 289–297, Dec. 2014.

[6] C. Wan, Z. Xu, and P. Pinson, "Direct interval forecasting of wind power," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4877–4878, Nov. 2013.

[7] P. Subbaraj and P. N. Rajnarayanan, "Optimal reactive power dispatch using self-adaptive real coded genetic algorithm," *Electr. Power Syst. Res.*, vol. 79, no. 2, pp. 374–381, Feb. 2009.

[8] J. Yuryevich and K. P. Wong, "Evolutionary programming based optimal power flow algorithm," *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1245–1250, Nov. 1999.

[9] Y. Jiang, C. Wan, J. Wang, Y. Song, and Z. Y. Dong, "Stochastic receding horizon control of active distribution networks with distributed renewables," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1325–1341, Mar. 2019.

[10] B. Zhao, C. X. Guo, and Y. J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1070–1078, May 2005.

[11] M. Varadarajan and K. S. Swarup, "Solving multi-objective optimal power flow using differential evolution," *IET Gener., Transmiss. Distrib.*, vol. 2, no. 5, pp. 720–730, Sep. 2008.

[12] Z. Cao, C. Wan, Z. Zhang, F. Li, and Y. Song, "Hybrid ensemble deep learning for deterministic and probabilistic low-voltage load forecasting," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1881–1897, May 2020.

[13] H. V. Pham, J. L. Rueda, and I. Erlich, "Online optimal control of reactive sources in wind power plants," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 608–616, Apr. 2014.

[14] J. G. Vlachogiannis and K. Y. Lee, "Quantum-inspired evolutionary algorithm for real and reactive power dispatch," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1627–1636, Nov. 2008.

[15] M. J. Dolan, E. M. Davidson, I. Kockar, G. W. Ault, and S. D. J. McArthur, "Distribution power flow management utilizing an online optimal power flow technique," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 790–799, May 2012.

[16] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[17] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, May 2014.

[18] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *J. Global Optim.*, vol. 27, no. 1, pp. 105–129, 2003.

[19] E. Mezura-Montes, J. Velazquez-Reyes, and C. A. C. Coello, "Modified differential evolution for constrained optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 25–32.

[20] C. Wan, J. Wang, J. Lin, Y. Song, and Z. Y. Dong, "Nonparametric prediction intervals of wind power via linear programming," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1074–1076, Jan. 2018.

[21] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2005, pp. 991–998.

[22] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[23] R. Storn and K. Price, *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. Berkeley, CA, USA: ICSI, 1995.

[24] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Adv. Intell. Syst., Fuzzy Syst., Evol. Comput.*, vol. 10, no. 10, pp. 293–298, 2002.

[25] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[26] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[27] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.

[28] *IEEE PES Working Group on Modern Heuristic Optimization*. [Online]. Available: https://www.uni-due.de/ieee-wgmho/competition2014

[29] G. Jia, Y. Wang, Z. Cai, and Y. Jin, "An improved ($\mu+\lambda$)-constrained differential evolution for constrained optimization," *Inf. Sci.*, vol. 222, pp. 302–322, Feb. 2013.

[30] R. P. Q. Alves, "Stochastic location of FACTS devices in electric power transmission networks," M.S. thesis, Faculdade Engenharia, Universidade Porto, Porto, Portugal, 2013.

MING NIU received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2010, and the M.S. and Ph.D. degrees in electrical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2012 and 2020, respectively.

He is currently a Postdoctoral Fellow with the State Grid Liaoning Electric Power Company Electric Power Research Institute, Shenyang, China. His research interests include renewable energy and reactive power compensation and planning.

NING ZHOU XU (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Hefei University of Technology, Hefei, China, in 2008 and 2011, respectively, and the Ph.D. degree in electrical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2015.

He is currently a Scientist with the Experimental Power Grid Center, Energy Research Institute @ NTU, Nanyang Technological University, Singapore. His research has been concerned with energy storage systems, power system reliability, and the synergy between EVs and the grid.

HE NAN DONG received the M.S. and Ph.D. degrees in power electronics and drives from the Shenyang University of Technology, Shenyang, Liaoning, China, in 2013 and 2018, respectively.

He is currently a Senior Engineer with the State Grid Liaoning Electric Power Company Electric Power Research Institute, Shenyang. His research interests include the key technology and power quality of microgrid.

YANG YANG GE received the M.S. and Ph.D. degrees in electrical engineering from the Shenyang University of Technology, Shenyang, Liaoning, China.

He is currently a Research Fellow with the State Grid Liaoning Electric Power Company Electric Power Research Institute, Shenyang. His research has been concerned with grid integration of renewable energy and energy storage systems.

YI TAO LIU received the M.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, China.

He is currently the Head of the State Grid Liaoning Electric Power Company Electric Power Research Institute, Shenyang, Liaoning, China. His research interests include renewable energy, power system reliability, active distribution networks, and integrated energy systems.

HOON TONG NGIN received the B.E. degree (Hons.) and the Ph.D. degree in electrical engineering from the National University of Singapore, Singapore.

He has over 18 years of track record in growing the research and development innovation ecosystem in Singapore's energy and infocomm sectors. His experience range from technology strategic planning, conceptualisation and delivery of research and development and test-bedding programmes in emerging growth areas, and forging strategic partnerships with multi-national corporations, local enterprises, and economic agencies to commercialise research and development efforts. He has managed over 200 million funding and over 30 research and development projects. They include large scale demonstration projects in micro-grids, energy storage, solar forecasting, and electric vehicle. Others include the application of emerging technologies, such as data analytics, artificial intelligence, blockchain, and drones/robotics, in power sector specific areas, such as power plant efficiency improvement, gas pipeline monitoring, smart grids/micro-grids, distributed energy resource management systems, and cyber-physical security enhancements.