IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 28, 2020, accepted October 4, 2020, date of publication November 3, 2020, date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3034730

A Cauchy-Gaussian Quantum-Behaved Bat Algorithm Applied to Solve the Economic Load Dispatch Problem

FRANÇOIS XAVIER RUGEMA[1], GANGUI YAN[1], (Member, IEEE), SYLVERE MUGEMANYI[2], QI JIA[1], (Member, IEEE), SHANFENG ZHANG[1], AND CHRISTOPHE BANANEZA[3]

[1]School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China
[2]College of Information Engineering, Northeast Electric Power University, Jilin 132012, China
[3]Integrated Polytechnic Regional College Tumba, Rulindo, Rwanda

Corresponding author: Gangui Yan (yangg@neepu.edu.cn)

This work was supported in part by the State Grid Corporation of China, and in part by the National Natural Science Foundation of China under Grant 51277024.

ABSTRACT In this paper, a novel Cauchy-Gaussian quantum-behaved bat algorithm (CGQBA) is applied to solve the economic load dispatch (ELD) problem. The bat algorithm (BA) is an acknowledged metaheuristic optimization algorithm owing to its performance. However, the classical BA presents some weaknesses, such as premature convergence. To withstand the drawbacks of the BA, quantum mechanics theories and Gaussian and Cauchy operators are integrated into the standard BA to enhance its effectiveness. Since the economic load dispatch is a nonlinear, complex and constrained optimization problem, its main objective is to reduce the total generation cost while matching the equality and inequality constraints of the system. The validity of the CGQBA is tested on six standard benchmark functions with different characteristics. The numerical results indicate that the CGQBA is effective and superior to many other algorithms. Moreover, the CGQBA is applied to solve the ELD problems on various test systems including 3,6,20, 40,110 and 160 implemented generating units. The simulation results illustrate the strength of the CGQBA compared with other algorithms recently reported in the literature.

INDEX TERMS Economic load dispatch, bat algorithm, Gaussian operator, Cauchy operator, quantum behavior.

I. INTRODUCTION

Economic load dispatch (ELD) is one of the hot topics in the field of power system optimization [1]. The aim of the economic dispatch is to seek the most favorable allocation of generation units that reduces costs while meeting all the system's inequality and equality constraints [2]. Due to the excessive cost of power generation in fossil fuel power plants, the optimality of economic dispatch is advantageous in terms of saving money [3].

As an optimization problem, the economic dispatch (ED) problem consists of an objective function and various constraints [4]. In previous years, several methods known as conventional methods have been used to solve economic dispatch problems, such as linear programming [5], nonlinear programming [6], quadratic programming [7], dynamic programming [8], interior point programming [9], mixed integer programming [10], the Lagrangian relaxation algorithm [11], the decomposition technique [12], the branch-and-bound method [13], the Newton-Raphson method [14], Lambda iteration [15] and the gradient method [16]. However, these classical methods experience difficulty when finding initial solutions and are prone to local optimal convergence [17]. The cost functions of generators were previously formulated as quadratic or piecewise quadratic functions based on the assumption that the incremental cost curves of the units are monotonically increasing piecewise-linear functions [18]–[20]. The practical ED is nonconvex, nonsmooth and nondifferentiable due to the presence of several constraints, such as prohibited operating zones, ramp rates, multifuel options, valve-point effects and transmission losses [21]–[24]. Conventional optimization techniques are not effective in solving the ED problem; therefore, researchers have developed new optimization techniques

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

known as "metaheuristics", which provide better solutions and overcome the demerits of conventional methods [25]. Moreover, they are applied to solve problems in many areas of engineering [26], [27], as well as other real–world optimization problems [28]–[30].

Various metaheuristic algorithms have been applied for solving ED problems: ant colony optimization (ACO) [31], grey wolf optimization (GWO) [32], the flower pollination algorithm (FPA) [33], the firefly algorithm (FA) [34], the social spider algorithm (SPA) [21], the cuckoo search algorithm (CSA) [3], the symbiotic organism search algorithm (SOS) [35], the bat algorithm (BA) [36], the particle swarm algorithm (PSO) [20], the imperialist competitive algorithm (ICA) [37], ant lion optimization (ALO) [38], the gravitational search algorithm (GSA) [39], the genetic algorithm (GA) [40], teaching-learning based optimization (TLBO) [41], the artificial bee colony algorithm (ABC) [42], and the differential evolution algorithm (DE) [43].

Some of these metaheuristic algorithms possess limitations when dealing with large complex systems in terms of finding the global optimal solutions, preventing local optimal and premature convergence [44]. To address these shortcomings, two solutions have been proposed to improve the performance of metaheuristics: modification and hybridization of algorithms [45].

A study by Elsayed *et al.* [46] suggested a modified social spider algorithm (MSSA) for solving the ELD problem, where the random walk in the basic SSA has been substituted by a chaotic sequence. An improved version of the adaptive differential evolution optimizer (ADE) has been applied for solving the nonconvex ED problem [47]. This presented algorithm shuns the problem of premature convergence and ameliorates the convergence speed. An elitist cuckoo search algorithm was examined in [48]. This variant of the CSA is based on modifying some parameters of the basic CSA (initialization), and it achieves remarkable success in solving the ELD problem. A variant of the charged search system (CSS) called the adaptive CSS (ACSS) was developed in [49], and the modifications performed in this algorithm focused on initialization and random walk. Consequently, the ACSS shows supremacy over the CSS in solving the ELD problem. A modified crow search algorithm (MCSA) was presented in [50], and the MCSA differs from the original CSA in terms of selecting the new crows and tuning the flight length. Phasor particle swarm optimization (PPSO) has been suggested for solving convex and nonconvex/nonsmooth ELD problems, as found in [51]. PPSO is endowed with convergence ability and higher performance due to the substitution of the control parameters of the basic PSO with phasor angles. The author in [52] suggested the solution of large-scale multifuel ED problems considering valve-point effects via a dual-population adaptive differential evolution (DPADE). In this study, a dual-population mechanism was used to enhance the searching capability, and an adaptive technology was utilized to elude the unsuitable parameters and to tune two parameters of great importance. An emended salp swarm algorithm (ESSA) for solving the economic emission power dispatch (EED) problem was suggested in [53]. In this study, the reproduction cycle of salp was integrated into the classical salp swarm algorithm (SSA) to prevent the algorithm from being caught in the local minimum and guarantee the diversity of the swarm. The balance between exploitation and exploration is a feature achieved by the proposed ESSA. The ESSA successfully handles single and multiobjective ED problems and outperforms several methods reported in the literature.

A continuous GRASP (greedy randomized adaptive search procedure) algorithm has been hybridized with differential evolution (DE) for solving ED problems [54]. The proposed algorithm (C-GRASP-DE) is vested with the aptitude for global searching and the ability to avoid local optimal stagnation. A hybrid grey wolf optimization (HGWO) was successfully applied for solving the ED problem in [55]. This HGWO combines the advantages of both GWO and the DE. The crossover and mutation operators of the DE are integrated into the classical GWO algorithm to enhance its efficiency in handling ED problems. A robust hybrid optimization technique was designed for solving the ELD problem with wind uncertainty in [56]. This approach exploits the merits of both the genetic algorithm (GA) and adaptive simulated annealing (ASA). The diversity of the population is preserved by utilizing the nonuniform mutation of the GA. Self-adaptive mutation and crossover frameworks are incorporated along with ASA with the aim of facilitating the selection of the best parameters. Short computational times and convergences rate are the most appealing characteristics of the proposed method.

The hybridization of three metaheuristic algorithms, ant colony optimization (ACO), artificial bee colony algorithm (ABC) and harmonic search (HS), is presented in [44]. In this hybrid algorithm, each algorithm fills its own role. The task of seeking the initial solution set is handled by the ACO algorithm. The ABC algorithm checks and enhances the solutions generated by the ACO algorithm, while the HS algorithm removes the mediocre solutions from the solution set and substitutes them with those of higher quality. The authors in [57] combined the modified PSO (MPSO) and genetic algorithm (MPSO-GA) to solve nonsmooth as well as nonconvex ED problems. The initialization is performed by the GA, and the results are conveyed to the MPSO. The exploration of all search spaces is not necessary in MPSO-GA since this work is assigned to the GA. The results reveal the supremacy of the MPSO-GA over both the MPSO and the GA. A hybrid optimization method that integrates PSO and termite colony optimization (TCO), known as HPSTCO, has been developed and applied for solving the dynamic economic dispatch (DED) problem [58]. In this HPSTCO algorithm, PSO iterations are tasked with global searching, while TCO iterations are assigned to explore the vicinity of the global solution. A study by [59] proposed the hybridization of competitive algorithms (ICAs) and sequential quadratic programming (SQP), known as HIC-SQP, for solving ELD problems with wind power. Although the ICA is a metaheuristic algorithm, it presents the disadvantage of being caught into local optima as the number of imperialists rises. The SQP has been employed to palliate this demerit, and the role of the SQP is to adjust the results of the ICA to improve its performance. A robust hybrid gravitational search algorithm (RHGSA) for handling ED problems was addressed in [60]. The efficiency of the classical GSA was improved thanks to the piecewise linear chaotic map that enhances the global search capability and the sequential quadratic programming that boosts the speed of the local search. The authors in [61] suggested a solution to the ED problem considering smooth cost function features by using a combination of lambda iteration and simulated annealing methods (MHLSA). In the proposed approach, the demerits of the SA, such as poor initialization and difficulty when performing global searches, are alleviated by using lambda iterations. Moreover, this method involves multiple searches to discover the locality with the best-suited global optimal solutions. The metropolis biogeography-based optimization-sequential quadratic programming (MpBBO-SQP) algorithm has been proposed to cope with the weakness of BBO [62]. A metropolis criterion of the simulated annealing (SA) algorithm is introduced in BBO to provide control over migrated individuals, thus improving the exploration quality. The solutions generated by MpBBO are adjusted by SQP to increase the performance of the MpBBO-SQP algorithm. 3, 13 and 40 generating units are used to test the validity of the proposed algorithm.

An enhanced bat algorithm (EBA) for solving the ED problem was presented in [63]. The modifications have been conducted on the classical BA, which contains the inertia weight, population distribution, pulse emission rate and loudness. A modified θ bat algorithm is presented in [64]. The proposed approach transforms the Cartesian search space into polar coordinates as a means for providing strong search capacity. In this method, three modifications are introduced into the BA: 1) Lévy flight and a genetic mutation operator are employed to increase the population diversity, and 2) the loudness parameters are tuned to accelerate the convergence rate. Moreover, an adaptive strategy is adopted to facilitate the selection of the best modification to avoid local optima.

A combination of PSO and the BA has been proposed for solving ED problems [65]. In this hybrid algorithm, the PSO integrates the frequency behavior of the bat algorithm to accelerate its velocity updates. The loudness of the bat algorithm is utilized to address boundary constraint violations as long as the solution improves. The authors in [66] suggested the multiobjective chaotic bat algorithm (MOCBA) to handle the EED problem. The chaotic map is introduced to modify both the loudness and the pulse emission rate with the purpose of preventing premature convergence. A Pareto optimal front has been employed to facilitate the simultaneous minimization of fuel and emission.

Most of the abovementioned papers reveal that the balance between exploration and exploitation is extremely important. A novel algorithmic framework for solving economic load dispatch problem is proposed in this paper. To the best of the authors' knowledge, the Cauchy-Gaussian quantum -behaved bat algorithm has never been applied for solving the ELD problem. The quantum theory is introduced into classical BA to ameliorate the searching capability. Moreover, Gaussian and Cauchy mutations are used to improve the population diversity and to enable the algorithm to escape from local optima. To validate the proposed algorithm, the CGQBA is tested on six benchmark functions and applied to solve ELD problems containing 3, 6, 20, 40, 110 and 160 generating units. The rest of the paper is organized as follows: Section II provides details about the formulation of the ELD problem. Section III explains the Bat Algorithm (BA) and the Cauchy-Gaussian quantum-behaved bat algorithm (CGQBA). Section IV portrays various test systems and the simulation results of the proposed algorithm in comparison with other well-known algorithms reported in the literature. Finally, in Section V, the conclusion of the paper is drawn.

II. ED PROBLEM FORMULATION

This section explains the practical formulation of the ED problem with an objective function subject to constraints [49], [67], [68].

A. OBJECTIVE FUNCTIONS

The quadratic fuel cost function of the thermal units is minimized according to the following expression:

$$\min_{P\in R^{N_g}} F=\sum_{j=1}^{N_g} F_j\left(P_j\right)=\sum_{j=1}^{N_g}\left(a_j+b_jP_j+c_jP_j^2\right) \tag{1}$$

where N_g is the total number of generating units, $F_j\left(P_j\right)$ is the fuel cost of the j^{th} generating unit (in \$/hr), P_j is the power generated by the j^{th} generating unit in MW, and a_j, b_j and c_j are the cost coefficients of the j^{th} generator.

For the case that takes the valve-point loading effect into consideration, the objective function is expressed as follows:

$$\begin{aligned}\min_{P\in R^{N_g}} F &= \sum_{j=1}^{N_g} F_j\left(P_j\right)\\ &= \sum_{j=1}^{N_g}\left(a_j+b_jP_j+c_jP_j^2\right)+\left|e_j\sin\left(f_j\left(P_j^{\min}-P_j\right)\right)\right|\end{aligned} \tag{2}$$

where e_j and f_j are the constants for the valve-point effects of generators.

For the case where multiple fuel options are presented, the fuel cost of the j^{th} generator is given by

$$F_j(P_j) = \begin{cases} a_{j1}P_j^2 + b_{j1}P + c_{j1}P, & \text{fuel } 1, P_j^{\min} \le P_j \le P_{j1} \\ a_{j2}P_j^2 + b_{j2}P + c_{j2}P, & \text{fuel } 2, P_{j1} \le P_j \le P_{j2} \\ \vdots \\ a_{jk}P_j^2 + b_{jk}P + c_{jk}P, & \text{fuel k}, P_{jk-1} \le P_j \le P_j^{\max} \end{cases} \quad (3)$$

Every generator with k fuel options contains k discrete regions.

B. OPTIMIZATION CONSTRAINTS

The equality and inequality constraints of the ED problem are the power-balance equality and power generation limits, which are described by the following two equations:

$$\sum_{j=1}^{N_g} P_j = P_D + P_L \quad (4)$$

$$P_j^{\min} \le P_j \le P_j^{\max} \quad (5)$$

where $P_j, P_D, P_L, P_j^{\min}$ and $P_j^{\max}$ are the generation of the j^{th} generator unit (in MW), the total power demand (in MW), and the minimum and maximum power generation limits of the j^{th} generator, respectively. P_L represents the line losses (in MW), and its value is obtained using B coefficients, given by:

$$P_L = \sum_{j=1}^{N_g}\sum_{i=1}^{N_g} P_j B_{ji} P_i + \sum_{j=1}^{N_g} B_{0j}P_j + B_{00} \quad (6)$$

where P_i and P_j represent the power injection at the i^{th} and j^{th} buses, respectively, and B_{ij} indicates the loss coefficients, which are frequently assumed to be constant under normal operating conditions.

C. PRACTICAL OPERATING CONSTRAINTS OF GENERATORS

1) POZs (PROHIBITED OPERATING ZONES)

Because of the operation of the steam valve or vibrations in the shaft bearings, the operating zones are considered. In practice, operations in such areas must be prevented to attain the best fuel economy [69].The feasible operating zones of unit j are formulated as follows:

$$P_j \in \begin{cases} P_j^{\min} \le P_j \le P_{j,1}^l \\ P_{j,k-1}^u \le P_j \le P_{j,k}^l & k = 2, 3, \ldots n_j, j = 1, 2, \ldots n \\ P_{j,n_j}^u \le P_j \le P_j^{\max}, \end{cases} \quad (7)$$

where n_j, P_{jk}^l, P_{jk}^u are the number of prohibited zones and the lower and upper power outputs of the k^{th} prohibited zone of the j^{th} generator, respectively.

2) RAMP RATE LIMITS

The physical limitations of shutting down and starting up generators restrict ramp rate limits, which are formulated by the following two conditions:

Limitation of the increase in generation:

$$P_j - P_j^0 \le UR_j \quad (8)$$

Limitation of the decrease in generation:

$$P_j^0 - P_j \le DR_j \quad (9)$$

where P_j^0, P_j, UR_j, DR_j are the previous and current power outputs and the downramp and the upramp limits for the j^{th} generator, respectively.

Combining (8) and (9) with (5) leads to the following generation limits:

$$\underline{P_j} \le P_j \le \overline{P_j} \quad (10)$$

in which

$$\underline{P_j} = \max\left(P_j^{\min}, P_j^0 - DR_j\right) \quad (11)$$

and

$$\overline{P_j} = \min\left(P_j^{\max}, P_j^0 + UR_j\right) \quad (12)$$

Combining this with (2), the ED problem can be mathematically described as follows:

$$\min_{P \in R^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g}\left(a_j + bP_j + c_jP_j^2\right) + \left|e_j \sin\left(f_j\left(P_j^{\min} - P_j\right)\right)\right| \quad (13a)$$

$$\text{such that } \sum_{j=1}^{N_g} P_j = P_D + P_L$$
$$\max\left(P_j^{\min}, P_j^0 - DR_j\right) \le P_j \le P_{j,1}^l$$
$$P_{j,k-1}^u \le P_j \le P_{j,k}^l, \; k = 2, 3, \ldots n_j, \; j = 1, 2, \ldots n$$
$$P_{j,n_j}^u \le P_j \le \min\left(P_j^{\max}, P_j^0 + UR_j\right) \quad (13b)$$

III. THE BAT ALGORITHM

A. THE BASIC BAT ALGORITHM

The bat algorithm (BA) was designed by Yang in 2010 [70] and was inspired by the echolocation behavior of microbats while seeking prey, foraging, and avoiding obstacles [71].

The echolocation characteristics of microbats are modeled via three rules as described in [71]:

1) Each microbat uses echolocation to approximate the distances between prey and neighborhoods.
2) Flying is performed to look for prey and is done at random with velocity V^i at position X^i with a predetermined frequency $f^{\min}$, varying wavelength λ and loudness A^0. Bats can spontaneously tune the wavelengths

TABLE 1. List of abbreviations.

OPTIMIZATION TECHNIQUE	ABBREVIATIONS
Artificial Algae Algorithm	AAA[99]
Artificial Cooperative Search Algorithm	ACS[92]
Adaptive Charged System Search	ACSS[49]
Adaptive Differential Evolution with Multiple Mutation Strategies	ADE-MMS[98]
Ameliorated Grey Wolf Optimization	AGWO[107]
Bat Algorithm	BA [63],[88],[97]
Biogeography-Based Optimization	BBO[94],[102],[111]
Biogeography-based Particle Swarm Optimization	BLPSO[96]
Backtracking Search Algorithm	BSA[90],[94]
Chaotic Bat Algorithm	CBA[67]
Cross-Entropy Method and Sequential Quadratic Programming	CE-SQP[87]
Conventional Genetic Algorithm with Multiplier Lagrange	CGA_MU[67]
Cauchy-Gaussian Quantum-Behaved Bat Algorithm	CGQBA
Coulomb's and Franklin's Laws Theory Algorithm Optimizer	CFA[114]
Chaotic JAYA	CJAYA[101]
Chaotic Krill Herd Algorithm	CKH[18]
Continuous Quick Group Search Optimizer	CQGSO[95]
Crow Search Algorithm	CSA[50]
Cuckoo Search Algorithm	CSA[117]
Competitive Swarm Optimizer	CSO[105]
Differential Evolution Algorithm Biogeography-Based Optimization	DE-BBO[111]
Dual-Population Adaptive Differential Evolution	DPADE[98]
Enhanced Bat Algorithm	EBA[97]
Emended Salp Swarm Algorithm	ESSA[53]
Full Mixed-Integer Linear Programming	FMILP[91]
Genetic Algorithm	GA[90]
Genetic Algorithm and Ant colony algorithm for continuous domains	GA-API[90]
Genetic Algorithm with active power optimization method	GA-APO[90]
Genetic Algorithm-Pattern Search and Sequential Quadratic Programming	GA-PS-SQP[90]
Gaussian Quantum-Behaved Bat Algorithm	GQBA
Group Search Optimizer	GSO[67]
Grey Wolf Optimization	GWO[32],[100]
Hybrid Artificial Algae Algorithm	HAAA[99]
Honey Bee Mating Optimization	HBMO[102]
Hopfield Modeling	HM[93]
Improved Genetic Algorithm	IGA[100]
Improved Genetic Algorithm with Multiplier Lagrange	IGA_MU[67]
Interactive Honey Bee Mating Optimization	IHBMO[102]
Self-Adaptive Evolution Algorithm with Improved Mutation Strategy	IMSaDE[98]
Improved Orthogonal Design Particle Swarm Optimization Algorithm	IODPSO[110]
Improved Orthogonal Design Particle Swarm Optimization Algorithm Global	IODPSO-G[110]

TABLE 1. ***(Continued.)*** **List of abbreviations.**

Improved Orthogonal Design Particle Swarm Optimization Algorithm Local	IODPSO-L[110]
Krill Herd Algorithm	KHA[113]
Lambda Iteration	λ-Iteration[97]
Modified θ-Bat Algorithm	M θ-BA[64]
Modified Artificial Bee Colony Algorithm	MABC[102]
Memory-Based Differential Evolution Algorithm	MBDE[98]
Modified Crow Search Algorithm	MCSA[50]
Modified Particle Swarm Optimization	MPSO[100]
Modified Sub-Gradient-Harmony Search algorithm	MSG-HS[90]
Modified Social Spider Algorithm	MSSA[46]
Non inferior Grey Wolf Optimization	NGWO[100]
New Particle Swarm Optimization	NPSO[100]
Natural Rank Harmony Search	NRHS[89]
Natural Tournament Harmony Search	NTHS[89]
Natural Proportional Harmony Search	NPHS[89]
Natural Global-worst Harmony Search	NGHS[89]
Genetic Algorithm with active power optimization based on the Newton's second order approach method	NSOA[90]
Oppositional-based Grey Wolf Optimization	OGWO[104]
Oppositional Invasive Weed Optimization	OIWO[104]
Opposition based Krill Herd Algorithm	OKHA[112]
Orthogonal Learning Competitive Swarm Optimizer	OLCSO[105]
Oppositional Real Coded Chemical Reaction Optimization	ORCCRO[111]
Pseudo-inspired Chaotic Bat Algorithm	PI-CBA[106]
Phasor Particle Swarm Optimization algorithm	PPSO[51]
Pattern Search	PS[90]
Particle Swarm Optimization algorithm	PSO[90]
Particle Swarm Optimization Local Random Search	PSO-LRS[101]
Quantum-Behaved Bat Algorithm	QBA
Quasi-Oppositional Teaching Learning Based Optimization	QOTLBO[32]
Quantum Particle Swarm Optimization Method1	QPSO-M1[109]
Quantum Particle Swarm Optimization	QPSO[109]
Real Coded Chemical Reaction Optimization	RCCRO[116]
Simulated annealing based optimization	SA[50]
Self-Adaptive Differential Evolution Algorithm	SADE[98]
Shuffled Differential Evolution	SDE[32]
Semi Definite Programming	SDP[108]
Species-based quantum particle swarm optimization	SQPSO[103]
Salp Swarm Algorithm	SSA[53]
Social Spider Algorithm	SSA[21]
Teaching–Learning Based Optimization	TLBO[32]

TABLE 2. Benchmark functions data.

Benchmark Functions	n	Search Space	Global Minimum
$f_1(x)=\sum_{i=1}^{n}\left(100\left(x_{i+1}+x^2\right)^2+\left(x_i-1\right)\right)^2$	30	$[-30,30]^n$	0
$f_2(x)=\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)+10\right)^2$	30	$[-5.12,5.12]^n$	0
$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$	30	$[-100,100]^n$	-1.0316285
$f_4(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$[-5,5]^n$	0
$f_5(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	$[-100,100]^n$	0
$f_6(x)=\frac{1}{400}\sum_{i=1}^{n}(x_i-100)^2-\prod_{i=1}^{n}\cos\left(\frac{x_i-100}{\sqrt{i}}\right)+1$	30	$[-600,600]^n$	0

(or frequencies) of their emitted pulses and tune the rate of pulse emission $r_1 \in [0, 1]$ based on the vicinity of their objective.

3) The loudness is supposed to change from a large (positive) A^0 to the least constant value $A^{\min}$.

Each bat i possesses a position X^i, a velocity V^i and a frequency f^i in $d-$ dimensional space, and these characteristics should be updated iteratively towards the current best position as follows:

$$f^i = f^{\min} + r_1\left(f^{\max} - f^{\min}\right) \tag{14}$$

$$V^i(t+1) = V^i(t) + f^i\left(X^i(t) - X^{best}(t)\right) \tag{15}$$

$$X^i(t+1) = X^i(t) + V^i(t+1) \tag{16}$$

where $r_1, f^{\min}, f^{\max}$ and f^i are a uniformly distributed random number in the range $[0, 1]$; the minimum tolerable frequency, the maximum tolerable frequency, and the frequency of the i^{th} bat, respectively. As given in [71], the current ED problem assumes that the values of $f^{\min}$ and $f^{\max}$ are set to 0 and 100, respectively. t is the current iteration number, and X^{best} is the location (solution) that possesses the best fitness in the current population. At initialization, V^i is assumed to be 0.

Each bat owns a new solution that can be generated locally through a random walk as follows:

$$X^{i,new}(t) = X^{old} + \varepsilon A^i(t) \tag{17}$$

where ε is a random number uniformly drawn from [0, 1] and $A^i(t)$ is the loudness.

Once the bat has discovered its prey, the loudness continues to decline while the pulse rate emission continues to rise. The loudness A^i and the pulse emission rate R^i are iteratively updated as follows:

$$A^i(t+1) = \alpha A^i(t) \tag{18}$$

$$R^i(t+1) = R^i(0)\left[1 - \exp(-\gamma t)\right] \tag{19}$$

where $A^i(0) \in [1, 2]$ and $R^i(0) \in [0, 1]$ are randomly generated within their respective limits. For the sake of simplicity, we set $\alpha = \gamma = 0.9$, as in [72]. The pseudocode of the bat algorithm is written as follows [71]:

Initialize the bat population X^i $(i = 1, 2, \ldots, N)$ and V^i;
Define the pulse frequency f^i, pulse rate R^i and loudness A^i;
while $(t < T)$ **do**
Generate new solutions by adjusting the frequency and updating the velocities and positions using equations (14)–(16);
if $(rand > R^i)$ **then**
Select a solution among the best solutions randomly;
Generate a local solution around the selected best solution using equation (17);
end if
if $(rand < A^i$ & &$f\left(X^i\right) < f\left(X^{best}\right))$ **then**
Accept the new solutions;
Increase R^i and reduce A^i using equations (18) and (19);
end if
Rank the bats and find the current X^{best};
$t = t + 1$;
end while
Output the best solution X^{best}

B. QUANTUM-BEHAVED BAT ALGORITHM

The quantum-behaved bat algorithm (QBA) is inspired by [73]–[75]. Some variants of the quantum-behaved bat algorithm are addressed in [76]–[78]. In [76], the frequency equation of the proposed algorithm includes the bats' capability of self-adaptive compensation for Doppler effects in echoes. Moreover, the algorithm formulates the bats' habitat selection as the selection between their quantum behaviors and mechanical behaviors. In [77], the presented algorithm possesses its own way of generating a new solution different from that of the original BA. The position of each bat is determined by both the current optimal solution and the mean best position, and the incorporation of quantum-behaved bats enables improvement of the population diversity and prevents the bats from falling into local minima. The improved version of [77] is addressed in [78].

In our paper, quantum theory is applied to the bat algorithm, and then two mutation operators, Gaussian and Cauchy, are incorporated. In the QBA, the bats possess quantum behavior, and their positions are updated as follows:

$$X^i(t+1) = \begin{cases} X^{best}(t) + \beta\left|M^{best}(t) - X^i(t)\right|\ln\left(1/u\right), & k \ge 0.5 \\ X^{best}(t) - \beta\left|M^{best}(t) - X^i(t)\right|\ln\left(1/u\right), & k < 0.5 \end{cases} \tag{20}$$

where both u and k are random numbers in the range [0,1] generated by the uniform distribution and β is the contraction-expansion coefficient, which can be adjusted for the sake of controlling the convergence speeds of the algorithms. It is defined as

$$\beta = \beta_0 + (T - t)\cdot(\beta_1 - \beta_0)/T \tag{21}$$

where β_0 and β_1 are the initial and final values of β, respectively.

TABLE 3. Comparison of various algorithm mean and standard deviation for benchmark functions [85].

Method		f_1	f_2	f_3	f_4	f_5	f_6
GA	Mean	338.5616	0.6509	9749.9145	−1.0298	7.961	1.0038
	Std.	361.497	0.3594	2594.9593	3.1314×10^{-3}	1.5063	6.7545×10^{-2}
PSO	Mean	37.3582	20.7863	1.1979×10^{-3}	−1.0160	0.4123	0.2323
	Std.	32.1436	5.94	2.1109×10^{-3}	1.2786×10^{-2}	0.25	0.4434
GSO	Mean	49.8359	1.0179	5.7829	−1 031628	0.1078	$3\ 0792\times10^{-2}$
	Std.	30.1771	0.9509	3.6813	0	3.9981×10^{-2}	$3\ 0867\times10^{-2}$
FEP	Mean	5.06	4.6×10^{-2}	5.7829×10^{-2}	−1.03	0.3	1.6×10^{-2}
	Std.	5.87	1.2×10^{-2}	1.6×10^{-2}	4.9×10^{-4}	0.5	2.2×10^{-2}
CEP	Mean	6.17	89	5.0×10^{-2}	−1.03	2	8.6×10^{-2}
	Std.	13.61	23.1	6.6×10^{-2}	4.9×10^{-4}	1.2	0.12
FES	Mean	33.28	0.16	1.4×10^{-3}	−1.0316	5.5×10^{-3}	3.7×10^{-2}
	Std.	43.13	0.33	5.3×10^{-4}	6.0×10^{-7}	6.5×10^{-4}	5.0×10^{-2}
CES	Mean	6.69	70.82	1.3×10^{-4}	−1.0316	0.35	0.38
	Std.	14.45	21.49	8.5×10^{-5}	6.0×10^{-7}	0.42	0.77
RCGA-IMM	Mean	5.9087×10^{-8}	1.2622×10^{-31}	2.9568×10^{-14}	−1.0316284535	1.3619×10^{-4}	1.3310×10^{-3}
	Std.	1.4870×10^{-9}	7.6639×10^{-31}	1.7137×10^{-13}	1.5621×10^{-15}	2.0693×10^{-4}	3.1219×10^{-3}
BA	Mean	90462737.63	258.3475	38479.1266	-0.734473977	72.5943	361.3092
	Std.	53802995.35	52.0387	10344.4701	0.58963	7.8481	91.109
QBA	Mean	22.8296	0	3.41×10^{-31}	-1.031628453	8.36×10^{-18}	0
	Std.	0.30916	0	1.69×10^{-30}	9.67×10^{-10}	4.85×10^{-17}	0
GQBA	Mean	22.6054	0	9.94×10^{-45}	-1.031628453	1.58×10^{-23}	0
	Std.	0.23954	0	4.87×10^{-44}	5.59×10^{-10}	9.80×10^{-23}	0
CGQBA	Mean	**22.3385**	**0**	**5.76×10^{-46}**	**-1.031628452**	**3.49×10^{-25}**	**0**
	Std.	**26.336**	**0**	**2.95×10^{-45}**	**1.09×10^{-9}**	**8.90×10^{-25}**	**0**

T is the maximum number of iterations, and t is the current iteration number.

We adopt $\beta_0 = 1$ and $\beta_1 = 0.5$ as in [74].

$$\begin{aligned} M^{best}(t) &= \left(M^{best,1}(t), M^{best,2}(t), \ldots, M^{best,d}(t)\right) \\ &= \left(\frac{1}{N}\sum_{i=1}^{N} P^{i,1}(t), \frac{1}{N}\sum_{i=1}^{N} P^{i,2}(t), \frac{1}{N} \right. \\ &\quad \left. \times \sum_{i=1}^{N} P^{i,3}(t), \ldots, \frac{1}{N}\sum_{i=1}^{N} P^{i,d}(t)\right) \end{aligned} \tag{22}$$

where M^{best} is the mean best position and represents the mean of all the best positions $P^i(t)$ of the population, $P^i(t)$ represents the current best position of the i^{th} bat. N denotes the size of swarm, and d indicates the dimension of the problem. The pseudocode of the quantum-behaved bat algorithm is shown as follows:

Initialize the bat population $X^i (i = 1, 2, \ldots, N)$ and V^i;
Define the pulse frequency f^i, pulse rate R^i and loudness A^i;
while $(t < T)$ **do**
Generate new solutions by adjusting the frequency and updating the velocities and positions using equations (14)–(16);
if $(rand < p_m)$//
Generate new solutions using equations (20)–(22);
End if

TABLE 4. Optimal generations and cost obtained by the CGQBA for test system 1 with load demand of 850 MW.

Unit	QBA	GQBA	CGQBA
1	300.280891	300.2668632	300.266887
2	399.985961	400	400
3	149.733148	149.7331368	149.733113
Total Power (MW)	850	850	850
Total Cost($/hr)	8234.07888	**8234.071766**	**8234.071766**

If $(rand > R^i)$
Generate a local solution around the selected best solution using equation (17);
end if
if $(rand < A^i \;\&\&\; f(X^i) < f(X^{best}))$ **then**
Accept the new solutions;
Increase R^i and reduce A^i using equations (18) and (19);
end if
Rank the bats and find the current X^{best};
$t = t + 1$;
end while
Output the best solution X^{best}

C. THE CAUCHY-GAUSSIAN QUANTUM-BEHAVED BAT ALGORITHM

Gaussian, Cauchy and exponential probability distributions are more effective than uniform probability functions in terms

TABLE 5. Best solution obtained for test 1.

Method	Best cost($/h)
CGQBA	**8234.07**
GQBA	8234.07
QBA	8234.07
BA[63][88]	8234.07
NTHS[89]	8234.07
NPHS[89]	8234.07
NGHS[89]	8234.07
CE-SQP[87]	8234.07
BSA[90]	8234.07
GA-API[90]	8234.07
GA-PS-SQP[90]	8234.07
PS[90]	8234.1
GA[90]	8237.6

TABLE 6. Optimal generations and cost obtained by the CGQBA for test system 2 with load demand of 283.4 MW.

Unit	QBA	GQBA	CGQBA
1	199.59965	199.59965	199.59964
2	20	20.000005	20
3	23.531683	23.994822	24.14266
4	18.6866101	15.277355	17.62494
5	18.9422879	20.296627	18.583482
6	13.5917104	15.011541	14.348708
Total power(MW)	294.3519	294.18	294.29945
Transmission loss(MW)	10.9519	**10.78**	10.89945
Total cost($/hr)	925.37763	924.91811	**924.90309**

of generating random numbers to update the velocity equation of the classical PSO [79]. Inspired by [80]–[82], in which two or more mutation operators are combined, we find that the incorporation of both Gaussian and Cauchy operators into the quantum–behaved bat algorithm improves its performance when applied to the ELD problem. The explanation of these operators is given below:

First, the Gaussian mutation operator is applied to the quantum bat algorithm. The one-dimensional Gaussian density function is given by the following equation [83], [84]:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]} \tag{23}$$

For $\mu = 0$ and $\sigma = 1$, the Gaussian distributed function is given by equation (24) [83], [84]:

$$g(x) = \frac{1}{\sqrt{2\pi}} e^{\left[\frac{-(x)^2}{2}\right]} \tag{24}$$

TABLE 7. Comparison of fuel costs and statistical results for 50 trial runs for test 2.

Method	Best	Mean	Worst	Std.dev.
CGQBA	**924.903**	924.903	926.923	0.7016
GQBA	924.918	925.341	927.05	0.659
QBA	925.377	926.466	928.11	1.0266
FMILP[91]	925.413	NA	NA	NA
BSA[90]	925.413	925.554	926.299	NA
ACS[92]	925.472	925.764	925.984	NA
MSG-HS[90]	925.641	926.851	928.599	NA
PSO[90]	926.388	925.758	928.427	NA
NSOA[90]	984.94	NA	992.48	NA
GA-APO[90]	996.04	NA	1101.49	NA
GA[90]	996.04	NA	1117.13	NA

In this section, we follow the same line of study as in [79]. The generation of random numbers is achieved by using the absolute value $|\cdot|$ of the Gaussian probability distribution with zero mean and unit variance, i.e., $|N(0, 1)|$, and then mapping to a truncated signal given by $G = 0.33= |N(0, 1)|$. The combination of the QBA with the Gaussian mutation operator is expressed by equation (25). Note that the parameter β of equation (20) has been substituted by G as indicated in equation (25).

$$X^i(t+1) = \begin{cases} X^{best}(t) + G \cdot \left|M^{best}(t) - X^i(t)\right| \cdot \ln(1/u), & k \geq 0.5 \\ X^{best}(t) - G \cdot \left|M^{best}(t) - X^i(t)\right| \cdot \ln(1/u), & k < 0.5 \end{cases} \tag{25}$$

The pseudocode of the Gaussian quantum-behaved bat algorithm is shown as follows:

Initialize the bat population $X^i (i = 1, 2, \ldots, N)$ and V^i;
Define the pulse frequency f^i, pulse rate R^i and loudness A^i;
while $(t < T)$ **do**
Generate new solutions by adjusting the frequency and updating the velocities and positions using equations (14)–(16);
if $(rand < p_m)$
Generate new solutions using equations (22) and (25);
End if
If $(rand > R^i)$
Generate a local solution around the selected best solution using equation (17);
end if
if $(rand < A^i \,\&\&\, f(X^i) < f(X^{best}))$ **then**
Accept the new solutions;
Increase R^i and reduce A^i using equations (18) and (19);
end if
Rank the bats and find the current X^{best};
$t = t + 1$;
end while
Output the best solution X^{best}

TABLE 8. Optimal generations and cost obtained by the CGQBA for test system 3 with load demand of 2500 MW.

Unit	QBA	GQBA	CGQBA
1	455.20312	442.42795	455.9158
2	166.58391	178.23783	155.89431
3	112.56265	106.24014	106.35206
4	92.654954	87.490627	91.13474
5	106.60247	110.34924	106.13723
6	85.901151	80.291207	86.680916
7	92.681221	76.92847	93.696409
8	110.19963	106.012	106.93367
9	79.410645	83.172127	83.655249
10	96.392598	115.4067	119.55964
11	227.84151	255.69318	234.79775
12	273.3376	257.5553	261.00972
13	152.31703	147.58458	150.6505
14	41.901902	56.353802	42.436675
15	165.27139	153.06742	169.24732
16	37.020092	38.469137	38.02095
17	60.912431	60.205659	68.911584
18	82.746576	85.156793	88.216855
19	91.498451	87.871493	66.67797
20	54.155261	56.412857	58.134902
Total power(MW)	2585.1946	2584.9265	2584.0643
Transmission loss(MW)	85.1946	84.926513	**84.066567**
Total cost($/hr)	62456.389	62455.862	**62455.413**

TABLE 9. Comparison of fuel costs and statistical results for 50 trial runs for test system 3.

Method	Best	Mean	Worst	Std. dev.
CGQBA	**62455.41276**	62455.4912	62469.123	1.65
GQBA	62455.86154	62455.8921	62471.521	1.73
QBA	62456.38915	62456.4211	62483.124	1.92
ADE-MMS[98]	62456.50744	62456.5921	62457.0609	0.132035
BLPSO[96]	62456.58	62456.64	62456.65	0.01
EBA[97]	62456.6182	NA	NA	NA
CBA[67]	62456.6328	62456.6348	62501.6714	0.3879
FMILP[91]	62,456.63	NA	NA	NA
CQGSO[95]	62456.6331	62456.6331	62456.6334	NA
CKH[18]	62456.6331	62456.788	62456.8755	NA
GSO[67]	62456.6332	62456.6336	62456.6353	NA
HM[93]	62456.6341	NA	NA	NA
λ-Iteration[93]	62456.6391	NA	NA	NA
BSA[94]	62456.6925	62457.1517	62458.1272	NA
BBO[94]	62456.7793	62456.7928	62456.7928	NA
BA[97]	62456.8042	NA	NA	NA

NA-not applicable/available

Second, the Cauchy mutation possesses the ability to escape from local optima [83] and it is applied to enhance the Gaussian quantum-behaved bat algorithm. The definition of the one-dimensional Cauchy density function is given by the following equation [83], [84]:

TABLE 10. Optimal generations and cost obtained by the CGQBA for test system 4 with load demand of 10500 MW.

Unit	QBA	GQBA	CGQBA
1	96.37558191	108.9402578	38.29980995
2	113.9893614	74.05190127	36.12592729
3	67.89653399	60.03677682	113.6500676
4	94.41242906	129.9375219	187.2948846
5	47	96.69359336	95.13176808
6	120.4674729	139.2308751	105.4054629
7	184.802235	294.9687324	260.8164129
8	209.800654	238.5052594	291.2094761
9	209.7977596	141.6799835	284.6188033
10	299.9709768	132.5661188	299.955829
11	168.729435	195.7018518	168.7961964
12	133.5908606	243.5966166	168.7907498
13	304.5150561	394.279371	498.67897
14	304.5194491	484.0380598	214.7993623
15	478.2026592	214.7601696	304.5096857
16	484.0101884	484.6636678	483.9627022
17	499.9795361	491.6999031	498.9084971
18	489.2788263	446.2404689	489.2764528
19	550	549.9253604	331.7827427
20	549.970328	549.9131324	511.2844827
21	549.9249644	549.9392091	538.3680524
22	549.9441271	543.0820748	528.6901047
23	540.8997709	523.873145	544.5483429
24	549.8243003	523.4191503	548.1112351
25	549.969797	549.9749201	547.9145335
26	549.9224235	523.288375	523.2877928
27	19.80489767	10.0030528	39.82719294
28	17.33344993	25.85507792	34.38120645
29	10.00379861	13.89069239	14.7191821
30	96.97717404	76.12002789	56.43992168
31	190	159.8608411	188.7411189
32	159.2479548	189.9828386	150.7052761
33	190	189.9830997	189.7172242
34	94.33221055	127.0758906	198.5541676
35	90.00258885	90.30890927	172.222451
36	164.7961243	199.987053	165.0439102
37	100.620832	66.9926599	80.43178821
38	94.22494779	99.30141186	25.00096848
39	25	54.35237509	58.71532444
40	549.8612968	511.2795702	511.2818704
Total power (MW)	10500	10500	10500
Total cost ($/hr)	121409.9813	121407.1246	**121406.89**

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{k}\right) \tag{26}$$

$x \in [-\infty, \infty]$, and $k > 0$ is the scale factor.

TABLE 11. Comparison of fuel costs and statistical results for 50 trial runs for test system 4.

Method	Best	Mean	Worst	Std. dev.
PI-CBA[106]	**121403.5812**	121454.52	121497.25	NA
HAAA[99]	121403.7	121425.56	121428.9	5.246522
AGWO[107]	121404.3	121412.3	121446.7	7.504
CGQBA	121406.8912	121411.505	121422.931	3.97777
GQBA	121407.1246	121420.554	121440.312	11.45843
AAA[99]	121407.8	121429.717	NA	11.54862
QBA	121409.9813	121423.246	121438.145	9.060165
MCSA[50]	121412.14	121413.28	121414.324	0.8761
ESSA[53]	121412.5	121517	121450.6	31.0236
PPSO[51]	121412.52	121413.95	121412.59	0.0563
θ-MBA[64]	121412.5355	121412.786	121412.948	NA
SDP[108]	121412.5355	NA	NA	NA
FMILP[91]	121412.54	NA	NA	NA
OIWO[104]	121412.54	NA	NA	NA
CBA[67]	121412.5468	121418.983	121436.15	1.611
SSA[53]	121412.55	NA	NA	NA
SQPSO[103]	121412.57	121455.7	121709.558	49.8076
MABC[102]	121412.68	121431.576	1214493.19	18.16
IHBMO [102]	121412.7533	121875.58	NA	NA
CHK[18]	121412.7553	121412.849	121412.905	0.0758
CE-SQP[76]	121412.88	121423.65	NA	NA
MSSA[58]	121413.46	121466.61	121521.73	28.6932
BA[97]	121,414.91	122,094.67	123,447.70	NA
IODPSO-G[110]	121414.93	121426.42	121416.54	17.75
BSA[94]	121415.6139	121474.882	121524.958	NA
OLCSO [105]	121415.8153	121504.049	121460.778	21.7993
HBMO [102]	121416.03	122019.65	NA	NA
IODPSO-L[110]	121420.98	121431.62	121424.62	18.69
SSA [53]	121426	121728.1	121500.5	71.91175
BBO[102]	121426.95	121508.03	121688.66	NA
QPSO-M1 [109]	121435.59	121490.02	121536.1	29.65
CSO[105]	121467.2964	121760.917	121540.97	55.3805
QPSO [109]	121468	121670.19	122002.98	135.2
NPSO[100]	121704.7391	122221.37	NA	NA
CJAYA[101]	121799.88	122581.85	NA	NA
NGWO[100]	121881.81	122787.77	NA	NA
IGA[100]	121915.93	122811.41	NA	NA
PSO-LRS[101]	122035.7946	122558.457	NA	NA
MPSO[100]	122252.265	NA	NA	NA
GWOII[100]	122430.74	123314.39	NA	NA
GWO[100]	122602.37	124796.61	NA	NA
GWOI[100]	122678.91	125155.07	NA	NA

Then, the Cauchy distributed function is defined by equation (27):

$$f_t(x) = \frac{1}{\pi} \frac{k}{k^2 + x^2} \tag{27}$$

The new candidates are generated by the following equation:

$$x^{i,new} = x^{old} + CA^i(t) \tag{28}$$

TABLE 12. Optimal generations and cost obtained by the CGQBA for test system 5 with load demand of 10500 MW.

Unit	QBA	GQBA	CGQBA
1	114	113.99997	114
2	36	114	114
3	120	60.000018	119.99914
4	190	80.000001	189.98586
5	97	47.000024	96.99607
6	68	68.000006	105.45398
7	300	300	300
8	135	135	299.99996
9	300	300	290.98448
10	300	299.9999	300
11	243.5996	374.99999	94
12	318.3991	94.000015	94.000006
13	499.746	500	394.27773
14	500	484.03916	484.04147
15	500	500	484.09761
16	500	500	484.10492
17	500	489.27939	489.25943
18	489.27942	500	489.22382
19	550	549.99987	513.22255
20	550	550	549.99985
21	550	550	524.25615
22	550	550	550
23	550	534.9643	550
24	550	549.99977	523.31601
25	550	550	550
26	550	550	524.64734
27	10	10.000003	10.001824
28	12.463594	10	10.000531
29	10	10.000002	10.000016
30	47	97	87.84134
31	63.884561	190	190
32	190	190	189.99557
33	190	189.99996	189.99346
34	200	200	200
35	200	90	200
36	94.111945	200	164.85813
37	110	110	109.99996
38	25	110	110
39	110	109.99996	110
40	550	550	549.99955
Total power(MW)	11423.484	11412.282	11362.557
Transmission loss(MW)	923.48423	912.28235	**862.55671**
Total cost($/hr)	136439.46	136298.9	**136109.16**

where C is a random number of Cauchy distributions in the range [0,1]

The pseudocode of the Cauchy-Gaussian quantum-behaved bat algorithm is shown as follows:

TABLE 13. Comparison of fuel costs and statistical results for 50 trial runs for test system 5.

Method	Best	Mean	Worst	Std.de
CGQBA	**136109.164**	136117.79	136128.43	6.819
GQBA	136298.896	136307.58	136320.13	8.312
HAAA[99]	136433.5	136436.6	136443.4	3.342
AAA[99]	136437.2	136446.07	136464.1	4.584
QBA	136439.461	136449.74	136462.46	8.274
OGWO[69]	136440.62	136442.26	136445.98	0.1
GWO[32]	136446.85	136463.96	136492.07	0.098
MCSA[50]	136448.63	136448.72	136448.95	1.101
CSA[50]	136452.487	136453.36	136453.7	2.347
OIWO[104]	136452.677	136452.68	136452.68	NA
OKHA[112]	136575.968	136576.15	136576.64	NA
KHA-IV[113]	136670.37	13667.229	136671.86	NA
ORCCRO[111]	136855.19	NA	NA	NA
DE/BBO[111]	136950.77	NA	NA	NA
BBO[111]	137026.82	NA	NA	NA
QOTLBO[32]	137329.86	NA	NA	NA
ACS[92]	137,413.73	NA	NA	NA
TLBO[32]	137814.17	NA	NA	NA
SDE [32]	138157.46	NA	NA	NA

Initialize the bat population $X^i(i = 1, 2, \ldots, N)$ and V^i;
Define the pulse frequency f^i, pulse rate R^i and loudness A^i;
while $(t < T)$ **do**
Generate new solutions by adjusting the frequency and updating the velocities and positions using equations (14)–(16);
if $(rand < p_m)$
Generate new solutions using equations (22) and (25);
End if
If $(rand > R^i)$
Generate a local solution around the selected best solution using equation (28);
end ifx
if $(rand < A^i \,\&\&\, f\left(X^i\right) < f\left(X^{best}\right))$ **then**
Accept the new solutions;
Increase R^i and reduce A^i using equations (18) and (19);
end if
Rank the bats and find the current X^{best};
$t = t + 1$;
end while
Output the best solution X^{best}

The flowchart of the Cauchy-Gaussian quantum-behaved bat algorithm is given in **Fig. 1.**

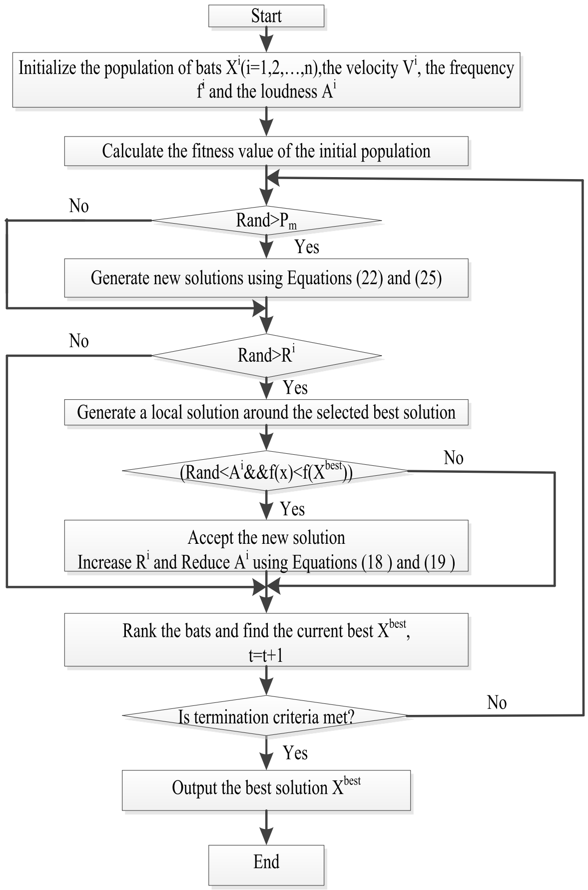

FIGURE 1. The Flowchart of CGQBA.

D. IMPLEMENTATION OF CGQBA TO SOLVE ED PROBLEM

Step 1: Initialize the population of bats which are bat position X^i and velocity. In this case, X^i corresponds to the power $\left(P^i\right)$ generated by the ith generator whereas n is defined as the number of generators. The value of X^i is randomly generated within the clearly defined boundaries $\left[P^{\min}, P^{\max}\right]$, and the initial value of V^i is set to zero.

Step 2: Initialize frequencies f^i, pulse rates R^i and the loudness A^i for each bat.

Step 3: Fix the maximum number of iterations

Step 4: Calculate the fitness values of all the bats utilizing the objective function in **Equation (1)**

Step 5: Generate the new solution by using **Equation (22) and (25)**

Step 6: Generate local solution in the vicinity of the best solution using **Equation (28)**

Step 7: Update both R^i and A^i using Equation (18) and (19), respectively.

Step 8: Verify if all the constraints are respected

Step 9: Repeat steps 1 to 8 until the maximum iteration is achieved.

TABLE 14. Optimal generations and cost obtained by the CGQBA for test system 6 with load demand of 15000 MW.

Unit	BA	QBA	GQBA	CGQBA	Unit	BA	QBA	GQBA	CGQBA	Unit	BA	QBA	GQBA	CGQBA
1	12	2.4	2.4	12	38	70	70	20	70	75	90	90	90	30
2	2.4	2.4	2.4	2.4	39	100	25	25	25	76	12	12	12	50
3	2.4	2.4	12	2.4	40	120	20	120	120	77	450	450	450	160
4	2.4	12	2.4	12	41	180	40	40	180	78	150	150	150	600
5	2.4	2.4	12	2.4	42	50	220	50	220	79	50	200	200	200
6	4	4	4	4	43	440	440	440	440	80	120	20	120	120
7	4	20	20	20	44	560	560	560	560	81	10	10	10	10
8	4	4	4	20	45	150	660	660	660	82	12	12	40	40
9	4	4	4	4	46	700	700	700	700	83	80	80	20	80
10	15.2	76	76	15.2	47	5.4	5.4	5.4	5.4	84	50	200	200	200
11	15.2	76	15.2	76	48	5.4	5.4	5.4	5.4	85	325	325	325	80
12	15.2	76	76	76	49	52	52	8.4	8.4	86	440	440	440	440
13	15.2	15.2	76	76	50	52	8.4	8.4	8.4	87	35	35	35	10
14	100	100	25	25	51	52	52	52	8.4	88	20	20	55	20
15	99.947	25	100	25	52	12	12	12	12	89	20	100	20	20
16	25	100	25	25	53	60	12	12	12	90	220	220	84.992	40
17	155	155	155	155	54	12	12	12	12	91	30	30	140	30
18	155	54.3	155	155	55	12	12	12	12	92	40	40	40	40
19	54.3	155	54.3	155	56	96	25.2	25.2	25.2	93	440	440	440	440
20	155	155	155	155	57	25.2	25.2	25.2	25.2	94	500	500	500	500
21	197	68.9	68.9	68.9	58	100	100	35	100	95	600	600	600	600
22	68.9	68.9	68.9	68.9	59	35	35	35	35	96	700	700	700	700
23	68.9	68.9	68.9	197	60	45	45	120	45	97	3.6	15	3.6	3.6
24	350	350	350	350	61	120	45	120	45	98	15	3.6	15	3.6
25	400	400	400	400	62	67.1	45	120	45	99	4.4	22	4.4	22
26	400	400	400	400	63	185	185	185	54.3	100	4.4	4.4	22	4.4
27	500	140	500	500	64	185	185	185	54.3	101	60	60	60	10
28	500	500	500	140	65	54.3	185	185	185	102	80	41.6	80	80
29	200	50	200	50	66	185	185	54.3	185	103	20.7736	20	20	20
30	25	100	25	100	67	70	70	70	197	104	120	20	20	20
31	10	10	10	50	68	70	70	70	134.2	105	40	40	40	40
32	5	5	5	5	69	70	70	70	70	106	40	40	40	40
33	80	80	80	80	70	360	360	360	360	107	50	50	50	50
34	250	250	250	250	71	400	400	400	400	108	30	30	30	30
35	360	360	360	360	72	400	400	400	400	109	40	40	40	40
36	400	130	130	400	73	60	300	60	60	110	20	20	20	20
37	10	10	40	40	74	250	250	188.31	250					
Total power(MW)											15000	15000	15000	15000
Total cost($/hr)											198048	197975.9	197961.454	**197853.82**

IV. RESULTS AND DISCUSSION

The performance of the GQBA is tested on seven different test systems, including 3-, 6-, 20-, 40-, 110- and 160-unit systems. The comparison between the achieved results for the proposed algorithm after 50 independent trial runs and the results of the recently published algorithms for each test system are reported in their respective **Tables**. The abbreviations of those algorithms are alphabetically ordered in **Table 1**. The number of bats is set to 20 for each of the test systems, and the maximum number of iterations is 1000. For the sake of simplifying the comparison, the best fuel costs among the results are organized in ascending order. MATLAB is used

TABLE 15. Comparison of fuel costs and statistical results for 50 trial runs for test system 6.

Method	Best	Mean	Worst	Std. dev.
CGQBA	**136109.164**	136117.79	136128.431	6.8188
GQBA	136298.896	136307.58	136320.128	8.3119
HAAA[99]	136433.5	136436.6	136443.4	3.3419
AAA[99]	136437.2	136446.07	136464.1	4.584
QBA	136439.461	136449.74	136462.461	8.2744
OGWO[69]	136440.62	136442.26	136445.98	0.1003
GWO[32]	136446.85	136463.96	136492.07	0.098
MCSA[50]	136448.63	136448.72	136448.949	1.1005
CSA[50]	136452.487	136453.36	136453.696	2.3465
OIWO[104]	136452.677	136452.68	136452.677	NA
OKHA[112]	136575.968	136576.15	136576.64	NA
KHA-IV[113]	136670.37	13667.229	136671.865	NA
ORCCRO[111]	136855.19	NA	NA	NA
DE/BBO[111]	136950.77	NA	NA	NA
BBO[111]	137026.82	NA	NA	NA
QOTLBO[32]	137329.86	NA	NA	NA
ACS[92]	137,413.73	NA	NA	NA
TLBO[32]	137814.17	NA	NA	NA
SDE [32]	138157.46	NA	NA	NA

to implement the programs on a personal computer with a 2.16 GHz processor and 4 GB RAM running on Windows 10.

A. BENCHMARK FUNCTION VALIDATION

Six benchmark functions are studied in this section to investigate the performance of the proposed CGQBA. Data for the benchmark functions are taken from [85] and are described in **Table 2**. The proposed CGQBA is applied to the aforementioned benchmark functions, and the mean and standard deviation of the results are provided in **Table 3**. Benchmark function data.

B. TEST SYSTEM 1

This system consists of three generators with a load demand of 850 MW. In this system, the constraints and valve-point load effects are taken into account, whereas the transmission losses are neglected. The system data are taken from [86]. Optimal generations and costs obtained by the QBA, GQBA and CGQBA for Test System 1 are presented in **Table 4.** As shown in **Table 4**, both the GQBA and CGQBA successfully achieve the best solution for the system, which is **$8234.071766/hr.**

A comparison of the statistical results of the QBA and GQBA. The CGQBA and the algorithms available in the literature, the CE-SQP [87], BA [63], [88], NRHS, NTHS, NPHS, NGHS [89], BSA, GA-API, GA-PS-SQP, PS, and GA [90], are provided in **Table 5.** Since the system size is small, the results show that a large number of algorithms converge to the same optimal solution.

The convergence characteristics of the PSO, BA, QBA, GQBA and CGQBA algorithms are illustrated in **Fig. 2.** The figure reveals that the CGQBA performs better than other methods because it converges to the optimal solution in early iterations.

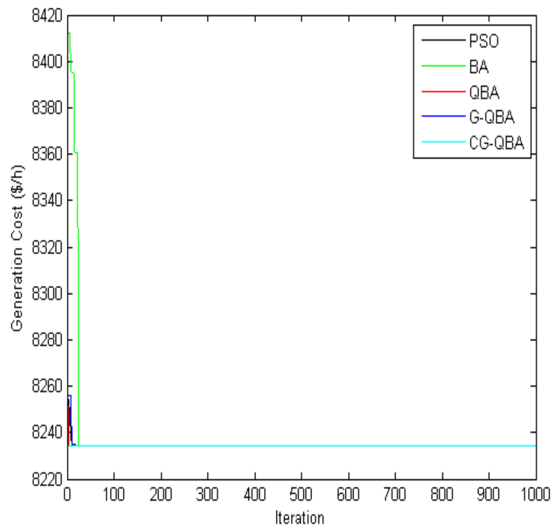

FIGURE 2. Convergence characteristic of the PSO, BA, QBA, GQBA and CGQBA for the test system 1.

C. TEST SYSTEM 2

This system comprises six generating units supplying a load demand of 283.4 MW. Transmission losses are included. The data are taken from [20]. **Table 6** provides the optimal generations and costs obtained. The best fuel cost and the corresponding transmission loss achieved by the CGQBA are **$924.90309 /hr and 10.8994507 MW,** respectively.

Table 7 presents a comparison of the statistical results of the GQBA, QBA and the other reported algorithms (the BSA, MSG-HS, PSO, NSOA, GA-APO, GA [90], FMILP [91] and ACS [92]). It is shown that the proposed CGQBA yields the best fuel cost compared to those obtained by these algorithms.

Fig. 3 shows the convergence behavior of the generation cost versus the iteration number for the PSO, BA, QBA, GQBA and CGQBA algorithms. It is seen in **Fig. 3** that both the GQBA and CGQBA obtain better convergence quality when compared to other methods, but the CGQBA achieves the more optimal solution than that of the GQBA.

D. TEST SYSTEM 3

This system consists of twenty generators with a load demand of 2500 MW. Transmission losses are considered in this system. The data are taken from [93]. As shown in **Table 8,** the best fuel cost and the corresponding transmission loss obtained by the CGQBA are **$62455.413/hr** and **84.066567 MW,** respectively.

TABLE 16. Optimal generations and cost obtained by the CGQBA for test system 7 with load demand of 43200 MW.

Unit	QBA	GQBA	CGQBA	Unit	QBA	GQBA	CGQBA	Unit	QBA	GQBA	CGQBA
1	227.0933	233.81598	216.5964	55	274.414	273.851	309.152	109	410.594	396.436	409.1712
2	195.9504	195.63533	198.2955	56	246.252	248.846	251.378	110	272.675	254.262	324.5515
3	282.1459	285.62708	284.7251	57	287.776	280.595	281.515	111	219.921	225.726	220.475
4	230.5025	243.1311	251.5097	58	239.239	241.753	240.72	112	205.74	218.888	206.6354
5	280.0996	257.28591	297.7732	59	419.206	403.438	412.143	113	266.662	272.763	287.208
6	226.9591	247.84226	227.4371	60	336.036	291.233	287.867	114	226.607	249.305	241.0079
7	269.144	271.50048	281.6746	61	225.392	226.686	233.38	115	277.139	307.724	307.0386
8	237.1294	241.39337	228.6532	62	210.34	212.082	201.943	116	242.325	236.906	233.7432
9	408.3086	388.16335	402.7713	63	298.915	259.421	292.388	117	311.863	285.117	291.4959
10	287.9162	268.1595	312.1391	64	231.47	236.519	235.541	118	242.604	248.22	233.3913
11	222.8895	235.68783	237.7522	65	295.457	290.801	224.775	119	412.165	391.064	419.5338
12	197.1797	214.87292	190.078	66	232.226	238.421	248.043	120	316.936	269.688	273.2213
13	276.0141	292.40361	305.8601	67	268.592	287.789	310.961	121	215.587	228.631	223.4569
14	229.962	237.35957	234.4421	68	240.453	243.737	243.396	122	196.579	210.002	196.6607
15	278.5164	272.01704	250.9043	69	405.538	391.796	412.768	123	254.317	312.267	297.0651
16	237.2567	246.32485	221.4511	70	309.807	289.466	290.177	124	223.918	246.071	251.1424
17	300.9728	264.77794	302.0016	71	228.565	235.396	215.507	125	301.106	260.671	241.1329
18	232.2865	248.78486	237.9486	72	205.382	201.969	211.581	126	227.679	250.653	248.1056
19	392.212	387.02355	401.2882	73	286.349	281.244	292.84	127	279.824	279.069	301.3867
20	312.1737	296.72638	326.5094	74	245.256	239.996	245.952	128	233.759	238.162	233.5128
21	226.6501	231.77791	231.9214	75	310.456	275.725	251.897	129	420.084	412.934	404.0258
22	197.063	211.4354	218.5696	76	233.795	244.834	241.226	130	296.415	264.57	299.8126
23	292.0757	280.06253	274.754	77	284.113	262.265	242.646	131	225.468	237.173	228.3763
24	237.4966	250.44281	221.6351	78	235.465	256.772	234.033	132	214.568	217.126	204.9498
25	270.2077	278.02622	231.4776	79	420.615	387.347	407.473	133	304.402	289.697	292.1551
26	239.0637	247.02845	242.3073	80	295.774	301.053	332.835	134	243.486	236.133	253.3197
27	315.8967	273.79491	282.1908	81	227.375	233.288	219.002	135	286.398	297.03	297.2293
28	228.6293	253.16721	234.6506	82	209.531	204.549	206.572	136	242.853	247.032	251.1881
29	409.4493	408.74665	410.2228	83	293.338	296.93	307.039	137	263.238	289.323	264.5257
30	299.7111	294.96737	288.6169	84	235.589	248.64	233.378	138	239.815	246.224	246.582
31	238.4905	224.31581	224.8278	85	276.704	275.855	320.139	139	431.825	400.332	437.3552
32	202.5605	216.78405	181.2199	86	242.191	233.194	238.296	140	318.332	283.437	254.2257
33	301.0306	293.39699	290.3968	87	291.338	311.037	283.008	141	228.312	236.552	220.1992
34	243.4042	248.88884	238.9507	88	239.895	246.349	241.16	142	196.444	211.144	194.6555
35	312.1566	242.21789	306.3575	89	415.556	388.766	398.357	143	286.516	263.691	317.5469
36	235.2102	246.50897	233.3766	90	278.199	296.815	321.416	144	232.511	245.693	245.7314
37	287.3945	281.27257	244.8311	91	228.81	230.231	228.852	145	292.133	292.798	258.2503
38	235.6049	240.20485	240.083	92	200.404	217.919	196.351	146	240.159	246.009	236.6494
39	391.1708	406.5467	399.198	93	278.972	309.635	275.114	147	258.739	280.041	286.9398
40	290.1982	292.80229	265.0282	94	239.499	258.588	241.157	148	219.256	245.911	236.2101
41	224.4584	230.6574	229.5243	95	295.343	280.274	274.845	149	400.718	383.469	405.3713
42	196.8461	211.09853	210.0145	96	237.914	245.153	238.48	150	256.386	315	288.918
43	257.7563	240.65049	235.7187	97	266.824	284.199	305.748	151	225.591	236.067	221.5838
44	234.4088	241.2455	238.4017	98	238.721	240.37	228.46	152	196.919	213.41	216.2399
45	260.8812	291.46462	302.2315	99	405.117	404.213	423.269	153	310.978	285.294	263.0905

TABLE 16. ***(Continued.)*** **Optimal generations and cost obtained by the CGQBA for test system 7 with load demand of 43200 MW.**

46	228.2023	246.90082	235.0699	100	279.788	289.698	281.924	154	240.646	234.265	235.5983
47	278.8206	266.45183	301.2073	101	237.677	232.469	229.336	155	248.105	289.577	300.1288
48	243.6625	249.16848	241.0797	102	209.102	209.986	192.28	156	245.818	243.671	246.9543
49	419.7121	405.67279	414.1948	103	281.095	260.167	287.194	157	307.999	289.942	288.7732
50	320.4	309.10795	308.6949	104	231.558	237.493	230.856	158	239.115	243.056	248.0565
51	219.6904	231.22114	221.0311	105	265.294	262.654	290.264	159	414.291	397.477	417.7958
52	210.4332	206.37627	193.1934	106	238.294	245.829	227.676	160	311.462	279.136	311.1688
53	289.7025	271.74649	261.8134	107	285.576	283.312	260.486				
54	232.3848	240.44574	243.017	108	230.894	241.386	248.021				
Total power(MW)									43200	43200	43200
Total cost($/hr)									10002	9996.14	**9994.324**

TABLE 17. **Comparison of fuel costs and statistical results for 50 trial runs for test system 7.**

Method	Best	Mean	Worst	Std. dev.
ADE-MMS[98]	**9764.9754**	9905.535	10250.09	140.2813
ACSS [49]	9979.8281	9983.098	9985.697	1.1472
DPADE[52]	9982.5603	9982.676	9982.865	0.0654
CGQBA	9994.3235	9998.234	10026	2.14
PI-CBA[106]	9995.805	10029.08	10069.74	
GQBA	9996.1387	9997.289	9999.124	0.888992
CSA[117]	9996.639	9996.639	10014.02	4.9268
QBA	10001.991	10002.88	10004.98	0.892378
CBA[67]	10002.86	10006.33	10045.23	3.2106
ORCCRO[111]	10004.2	10004.21	10004.45	NA
DE/BBO[111]	10007.05	10007.06	10010.26	NA
BBO[111]	10008.71	10009.16	10010.59	NA
RCCRO[116]	10009.518	10009.52	10009.58	NA
IGA_MU[67]	10042.474	10042.47	NA	NA
MBDE[98]	10082.877	10401.58	10811.46	223.9989
IMSaDE[98]	10129.224	10345.21	10569.6	121.4946
CGA_MU[67]	10143.724	10143.72	NA	NA
SADE [98]	10147.443	10281.39	10481.4	80.6311

The CGQBA yields the lowest cost in comparison with those of the other methods (the CKH [18], GSO, CBA [67], FMILP [91], λ-Iteration, HM [93], BSA, BBO [94], CQGSO [95], BLPSO [96] BA, EBA [97] and ADE-MMS [98]), as seen in **Table 9.**

The convergence characteristics of the PSO, BA, QBA, GQBA and CGQBA for Test System 3 are illustrated in **Fig. 4.** It is shown that the CGQBA obtains the best convergence property compared to the other methods as it converges to the optimal solution earlier.

E. TEST SYSTEM 4

This system comprises forty generators supplying a demand of 10500 MW and incorporating the valve-point loading

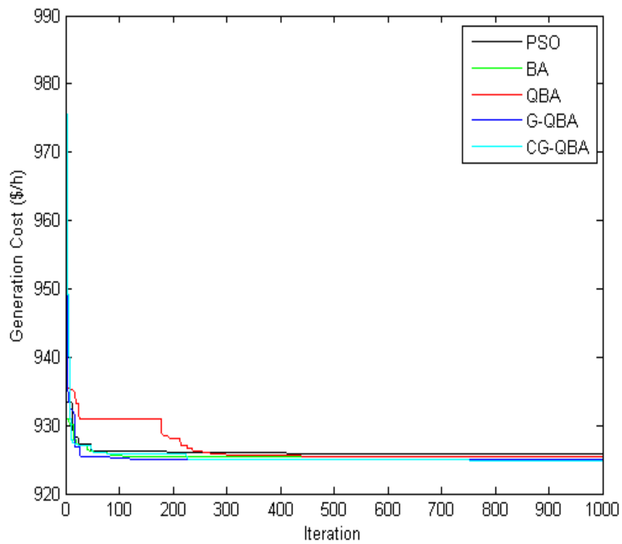

FIGURE 3. **Convergence characteristic of the PSO, BA, QBA, GQBA and CGQBA for the test system 2.**

effects. The transmission losses are neglected. The system data are taken from [86]. As shown in **Table 10**, the optimal cost obtained by the CGQBA is **$121406.8912/hr**.

Table 11 provides the comparison between the results obtained by the CGQBA and those obtained by the other methods (the CHK [18], SSA [21], MSSA [46], CSA, MCSA [50], PPSO [51], SSA, ESSA [53], θ-MBA [64], CBA [67], CE-SQP [87], BA [97], FMILP [91], BSA [94], AAA, HAAA [99], IGA, MPSO, GWO, GWOI, GWOII, NGWO, NPSO [100], CJAYA, PSO-LRS [101], BBO, HBMO, IHBMO, MABC [102], SQPSO [103], OIWO [104], CSO, OLCSO [105], PI-CBA [106], AGWO [107], SDP [108],QPSO, QPSO-M1 [109], IODPSO-G, and IODPSO-L [110]).

The statistical results reveal that the CGQBA can compete with many optimization methods; only the PI-CBA, HAAA and AGWO perform better than the CGQBA. **Fig. 5** depicts the convergence characteristics of the PSO, BA, QBA, GQBA and CGQBA.

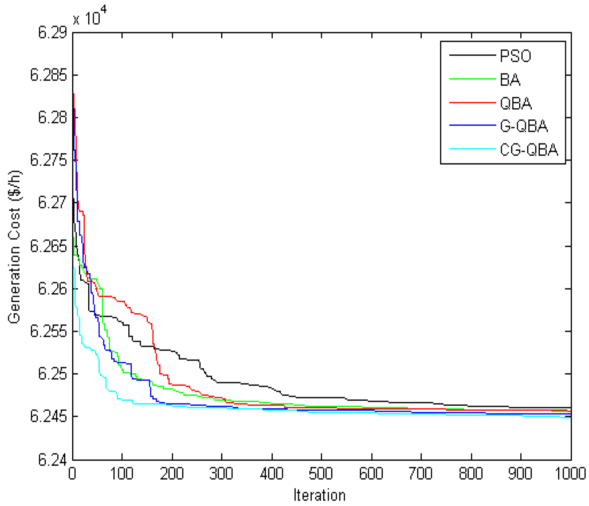

FIGURE 4. **Convergence characteristic of the PSO, BA, QBA, GQBA and CGQBA for the test system 3.**

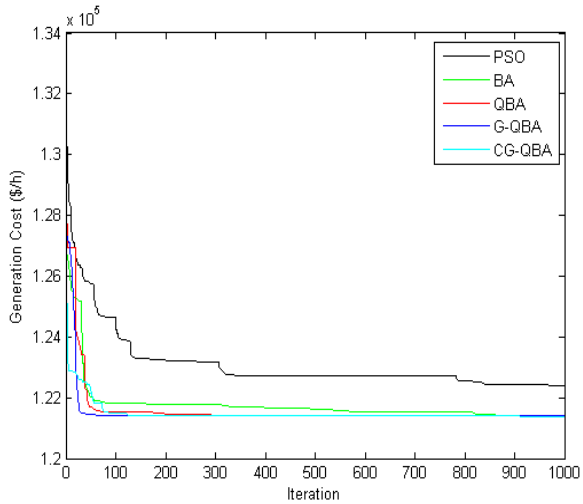

FIGURE 5. **Convergence characteristics of the PSO, BA, QBA, GQBA and CGQBA for the test system 4.**

From **Fig. 5,** the CGQBA performs better than the GQBA in early iterations in terms of convergence, but the GQBA becomes superior as the number of iterations increases. However, the optimal solution is finally achieved by the CGBA.

F. TEST SYSTEM 5

A system with 40 generating units meeting a load demand of 10500 MW is considered. This system incorporates the valve-point loading effects, and the transmission loss is considered. The data are given in [146] and [152].

Table 12 presents the results obtained by the proposed QBA, GQBA and CGQBA. The best fuel cost and the corresponding transmission loss achieved by the CGQBA are **$136109.16/hr** and **862.55671 MW,** respectively. In comparison with the other reported methods available in the literature, the GWO, QOTLBO, TLBO, SDE [32], CSA,

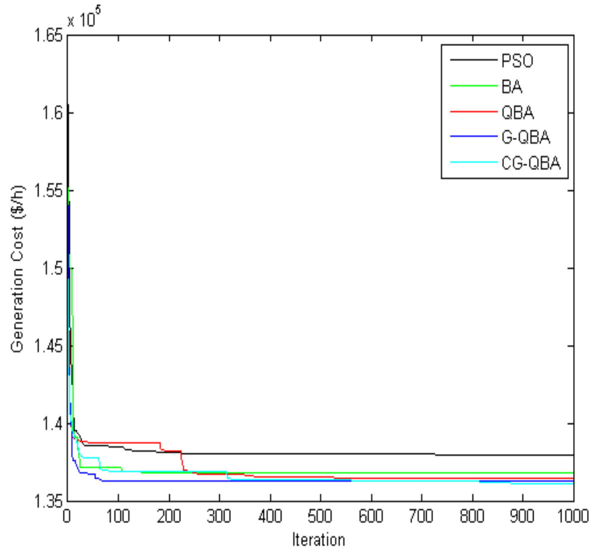

FIGURE 6. **Convergence characteristic of the PSO, BA, QBA, GQBA and CGQBA for the test system 5.**

MCSA [50], OGWO [69], ACS [92], AAA, HAAA [99], OIWO [104], BBO, DE/BBO, ORCCRO [111], OKHA [112] and KHA-IV [113], the CGQBA provides a better performance, as shown in **Table 13**.

Fig. 6 shows the convergence behavior of the PSO, BA, QBA, GQBA and CGQBA for Test System 5. As depicted in **Fig. 6**, the CBGQBA exhibits strong convergence in the beginning when compared to the other algorithms. The GQBA becomes better in later iterations, but finally, the optimal solution is achieved by the CGQBA.

G. TEST SYSTEM 6

This test system consists of 110 generating units with quadratic cost behavior. The load demand is 15000 MW, and valve-point loading effects are taken into consideration. The system data are taken from [104]. **Table 14** provides the results achieved by the QBA, GQBA and CGQBA. The best fuel cost obtained by the CGQBA is **$197853.82/hr**.

The compared algorithms for Test System 6 include the CSA, MCSA, SA, SAB, SAF [50], OIWO [104], CSO, OLCSO [105], AGWO [107], BBO, DE/BBO, ORCCRO [111] and CFA [114], as indicated in **Table 15.**

From the results obtained in **Table 15,** it is shown that the CGQBA performs better than the algorithms in recently cited works. The convergence characteristics of the PSO, QBA, GQBA and CGQBA are illustrated in **Fig. 7.** It is revealed that the CGQBA avoids being trapped into local optima and achieves the optimal solution at the end.

H. TEST SYSTEM 7

This test system comprises 160 generating units meeting a load demand of 43200 MW, and it is obtained by duplicating the 10-unit system 16 times.

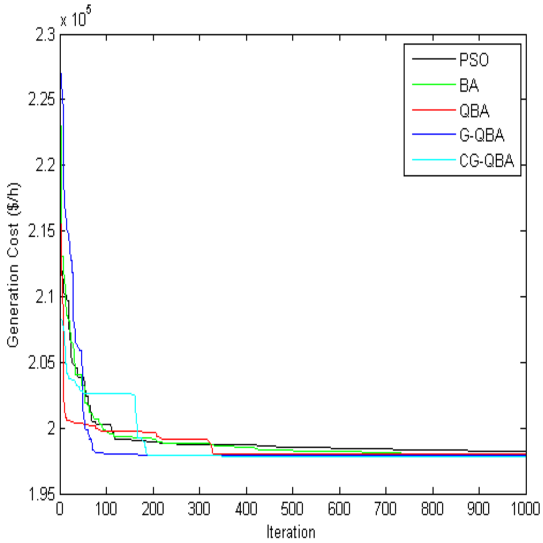

FIGURE 7. Convergence characteristic of the PSO, BA, QBA, GQBA and CGQBA for the test system 6.

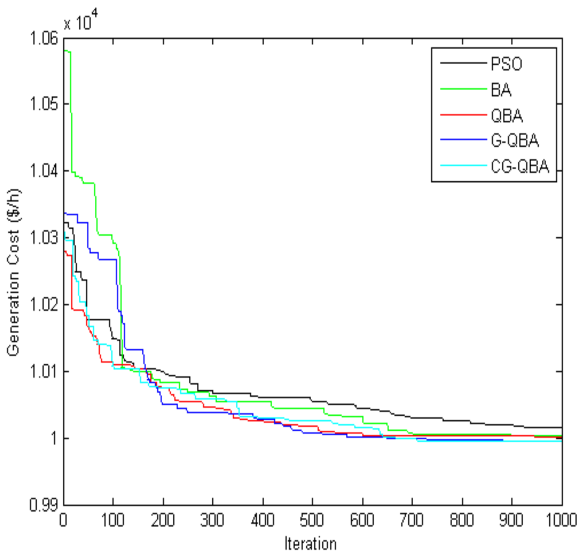

FIGURE 8. Convergence characteristic of the PSO, BA, QBA, GQBA and CGQBA for the test system 7.

The system contains multiple fuel options and incorporates valve-point loading effects. The data are adopted from [115]. The presentation of the optimal values and costs achieved by the QBA, GQBA and CGQBA are provided in **Table 16.** The best fuel cost obtained by the CGQBA is **$9994.3235/hr.Table 17** shows the comparison of statistical results of the CGQBA and the other recently reported algorithms (the ACSS [49], DPADE [52], CBA, CGA_MU, IGA_MU [67], ADE-MMS, SADE, MBDE, IMSaDE [98], PI-CBA [106], BBO, DE/BBO, ORCCRO [111], RCCRO [116] and CSA [117]).

According to the results, the CGQBA provides satisfactory results even if there are some algorithms that perform better than it. The most eminent algorithms that outperform the CGQBA are the ACSS, DPADE, and ADE-MMS.

Fig. 8 depicts the relevant convergence characteristics of the generation cost versus the number of iterations for the best solutions found by the PSO, BA, QBA, GQBA and CGQBA for Test System 7. It is shown in **Fig. 8** that there is an alternating pattern in terms of which algorithm is best based on convergence between the QBA, GQBA and CGQBA. In early iterations, the QBA and CGQBA are better than the GQBA. In the following iterations, the GQBA becomes better than the QBA and CGQBA. The superiority of the CGQBA over the QBA and GQBA is proven in late iterations as it converges to the optimal solution.

V. CONCLUSION

In this paper, a proposed Cauchy-Gaussian quantum-behaved bat algorithm is successfully applied for solving the ELD problem. Quantum mechanics theories and the Gaussian and Cauchy operators are integrated into the classical bat algorithm to improve its performance. First, the bat algorithm guarantees quantum behavior by incorporating quantum mechanics theories. Second, the Gaussian and Cauchy probability distributions are applied to the QBA in place of a uniform distribution to avoid the premature convergence that persists in the QBA and to balance exploitation and exploration.

To demonstrate the feasibility of the proposed method, we compare the GQBA, QBA and the other optimization methods reported in the literature based on the different test systems possessing 3, 6, 20, 40, 110 and 160 units, as illustrated in the **Tables**. According to the results, it can be seen that the CGQBA outperforms or can compete with many methods recently reported in the literature. Moreover, the CGQBA is proven to tackle small-, medium- and large-scale problems.

For future research, we will try using the combination of two or more operators and varying them in the pursuit of the most effective optimization algorithm. Moreover, we will study some of the more complex problems: dynamic economic/emission dispatch (DEED), combined heat and power (CHP), combined heat and power economic dispatch (CHPED), combined heat and power economic emission dispatch (CHPEED) and combined cooling, heating and power (CCHP) in the presence of renewable energy (photovoltaic and wind energy).

REFERENCES

[1] M. Basu, "Kinetic gas molecule optimization for nonconvex economic dispatch problem," *Int. J. Electr. Power Energy Syst.*, vol. 80, pp. 325–332, Sep. 2016.

[2] K. Zare, M. T. Haque, and E. Davoodi, "Solving non-convex economic dispatch problem with valve point effects using modified group search optimizer method," *Electr. Power Syst. Res.*, vol. 84, no. 1, pp. 83–89, Mar. 2012.

[3] M. Basu and A. Chowdhury, "Cuckoo search algorithm for economic dispatch," *Energy*, vol. 60, pp. 99–108, Oct. 2013.

[4] M. Modiri-Delshad, S. H. A. Kaboli, E. Taslimi-Renani, and N. A. Rahim, "Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options," *Energy*, vol. 116, pp. 637–649, Dec. 2016.

[5] R. A. Jabr, A. H. Coonick, and B. J. Cory, "A homogeneous linear programming algorithm for the security constrained economic dispatch problem," *IEEE Trans. Power Syst.*, vol. 15, no. 3, pp. 930–936, 2000.
[6] A. Sasson, "Nonlinear programming solutions for load-flow, minimum-loss, and economic dispatching problems," *IEEE Trans. Power App. Syst.*, vol. PAS-88, no. 4, pp. 399–409, Apr. 1969.
[7] R. M. S. Dhanraj and F. Gajendran, "Quadratic programming solution to emission and economic dispatch problem," *J. Inst. Eng., India*, vol. 86, pp. 129–132, Sep. 2005.
[8] Z.-X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses," *IEEE Trans. Power Syst.*, vol. 7, no. 2, pp. 544–550, May 1992.
[9] J. A. Momoh, R. F. Austin, R. Adapa, and E. C. Ogbuobiri, "Application of interior point method to economic dispatch," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 1992, pp. 1096–1101.
[10] Z. L. Wu, J. Y. Ding, Q. H. Wu, Z. X. Jing, and X. X. Zhou, "Two-phase mixed integer programming for non-convex economic dispatch problem with spinning reserve constraints," *Electr. Power Syst. Res.*, vol. 140, pp. 653–662, Nov. 2016.
[11] L. Zhigang, W. Wenchuan, Z. Boming, S. Hongbin, and G. Qinglai, "Dynamic economic dispatch using Lagrangian relaxation with multiplier updates based on a quasi-Newton method," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4516–4527, Nov. 2013.
[12] M. V. F. Pereira and L. Pinto, "A decomposition approach to the economic dispatch of hydrothermal systems," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 10, pp. 3851–3860, Oct. 1982.
[13] T. Ding, R. Bo, F. Li, and H. Sun, "A bi-level branch and bound method for economic dispatch with disjoint prohibited zones considering network losses," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 2841–2855, Nov. 2015.
[14] C. E. Lin, S. T. Chen, and C. L. Huang, "A direct Newton-Raphson economic dispatch," *IEEE Trans. Power Syst.*, vol. 7, no. 3, pp. 1149–1154, Aug. 1992.
[15] P. Aravindhababu and K. R. Nayar, "Economic dispatch based on optimal lambda using radial basis function network," *Int. J. Electr. Power Energy Syst.*, vol. 24, no. 7, pp. 551–556, Oct. 2002.
[16] D. M. D. Narvaez, G. C. Velez, and D. F. D. Narvaez, "Application of the gradient method in the economic dispatch," *Contemprary Eng. Sci.*, vol. 11, no. 96, pp. 4761–4768, 2018.
[17] M. Ghasemi, "A chaotic modified algorithm for economic dispatch problems with generator constraints," *Amer. J. Electr. Comput. Eng.*, vol. 1, no. 2, pp. 61–71, 2017.
[18] B. Bentouati, S. Chettih, and R. A.-A. El-Sehiemy, "A chaotic krill herd algorithm for optimal solution of the economic dispatch problem," *Int. J. Eng. Res. Afr.*, vol. 31, pp. 155–168, Jul. 2017.
[19] M. Basu, "Improved differential evolution for economic dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 63, pp. 855–861, Dec. 2014.
[20] Z.-L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187–1195, Aug. 2003.
[21] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for solving the non-convex economic load dispatch problem," *Neurocomputing*, vol. 171, pp. 955–965, Jan. 2016.
[22] K. Dasgupta, S. Banerjee, and C. K. Chanda, "Economic load dispatch with prohibited zone and ramp-rate limit constraints—A comparative study," in *Proc. IEEE 1st Int. Conf. Control, Meas. Instrum. (CMI)*, Jan. 2016, pp. 26–30.
[23] M. K. Modi, A. Swarnkar, N. Gupta, K. R. Niazi, and R. C. Bansal, "Stochastic economic load dispatch with multiple fuels using improved particle swarm optimization," *IFAC-PapersOnLine*, vol. 48, no. 30, pp. 490–494, 2015.
[24] M. Mohammadian, A. Lorestani, and M. M. Ardehali, "Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm," *Energy*, vol. 161, pp. 710–724, Oct. 2018.
[25] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and A. L. Bolaji, "Tournament-based harmony search algorithm for non-convex economic load dispatch problem," *Appl. Soft Comput.*, vol. 47, pp. 449–459, Oct. 2016.
[26] G. Platt, X. S. Yang, and A. S. Neto, *Computational Intelligence, Optimization and Inverse Problems With Applications in Engineering*, 1st ed. Cham, Switzerland: Springer, Jan. 2019.
[27] X.-S. Yang and X. He, "Nature-inspired optimization algorithms in engineering: Overview and applications," in *Nature-Inspired Computation in Engineering* (Studies in Computational Intelligence). Cham, Switzerland: Springer, 2016, pp. 1–20.
[28] A. Nakib and E.-G. Talbi, *Metaheuristics for Medicine and Biology* (Studies in Computational Intelligence), 1st ed. Berlin, Germany: Springer, 2017.
[29] S. P. Hey, "Heuristics and meta-heuristics in scientific judgement," *Brit. J. Philosophy Sci.*, vol. 67, no. 2, pp. 471–495, Jun. 2016.
[30] A. Soler-Dominguez, A. A. Juan, and R. Kizys, "A survey on financial applications of metaheuristics," *ACM Comput. Surveys*, vol. 50, no. 1, pp. 1–23, Apr. 2017.
[31] S. Pothiya, I. Ngamroo, and W. Kongprawechnon, "Ant colony optimisation for economic dispatch problem with non-smooth cost functions," *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 5, pp. 478–487, Jun. 2010.
[32] M. Pradhan, P. K. Roy, and T. Pal, "Grey wolf optimization applied to economic load dispatch problems," *Int. J. Elect. Power Energy Syst.*, vol. 83, pp. 325–334, Dec. 2016.
[33] A. Y. Abdelaziz, E. S. Ali, and S. M. A. Elazim, "Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems," *Energy*, vol. 101, pp. 506–518, Apr. 2016.
[34] X.-S. Yang, S. S. Sadat Hosseini, and A. H. Gandomi, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect," *Appl. Soft Comput.*, vol. 12, no. 3, pp. 1180–1186, Mar. 2012.
[35] Y. Sonmez, H. T. Kahraman, M. K. Dosoglu, U. Guvenc, and S. Duman, "Symbiotic organisms search algorithm for dynamic economic dispatch with valve-point effects," *J. Experim. Theor. Artif. Intell.*, vol. 29, no. 3, pp. 495–515, May 2017.
[36] L. A. Wulandhari, S. Komsiyah, and W. Wicaksono, "Bat algorithm implementation on economic dispatch optimization problem," *Procedia Comput. Sci.*, vol. 135, pp. 275–282, Jan. 2018.
[37] B. Mohammadi-ivatloo, A. Rabiee, A. Soroudi, and M. Ehsan, "Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch," *Energy*, vol. 44, no. 1, pp. 228–240, Aug. 2012.
[38] V. K. Kamboj, A. Bhadoria, and S. K. Bath, "Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer," *Neural Comput. Appl.*, vol. 28, no. 8, pp. 2181–2192, Aug. 2017.
[39] S. Duman, N. C. Sahu, and P. K. Hota, "Gravitational search algorithm for economic dispatch with valve-point effects," *Int. Rev. Electr. Eng.*, vol. 5, no. 6, pp. 2890–2895, 2010.
[40] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loadings," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1325–1331, Aug. 1993.
[41] S. Banerjee, D. Maity, and C. K. Chanda, "Teaching learning based optimization for economic load dispatch problem considering valve point loading effect," *Int. J. Electr. Power Energy Syst.*, vol. 73, pp. 456–464, Dec. 2015.
[42] S. Hemamalini and S. P. Simon, "Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions," *Electr. Power Compon. Syst.*, vol. 38, no. 7, pp. 786–793, 2010.
[43] N. Noman and H. Iba, "Differential evolution for economic load dispatch problems," *Electr. Power Syst. Res.*, vol. 78, no. 8, pp. 1322–1331, Aug. 2008.
[44] T. Sen and H. D. Mathur, "A new approach to solve economic dispatch problem using a hybrid ACO-ABC-HS optimization algorithm," *Elect. Power Energy Syst.*, vol. 78, pp. 735–744, Dec. 2016.
[45] X. Li, H. Zhang, and Z. Lu, "A differential evolution algorithm based on multi-population for economic dispatch problems with valve-point effects," *IEEE Access*, vol. 7, pp. 95585–95609, 2019.
[46] W. T. Elsayed, Y. G. Hegazy, F. M. Bendary, and M. S. El-bages, "Modified social spider algorithm for solving the economic dispatch problem," *Eng. Sci. Technol., Int. J.*, vol. 19, no. 4, pp. 1672–1681, Dec. 2016.
[47] M. Hamdi, L. Idomghar, M. Chaoui, and A. Kachouri, "An improved adaptive differential evolution optimizer for non-convex economic dispatch problems," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105868.
[48] A. Dutta, S. Das, B. Tudu, and K. K. Mandal, "A novel improved algorithm using cuckoo search for economic load dispatch," in *Proc. IEEE 1st Int. Conf. Power Electron., Intell. Control Energy Syst. (ICPEICES)*, Jul. 2016, pp. 1–5.

[49] P. Zakian and A. Kaveh, "Economic dispatch of power systems using an adaptive charged system search algorithm," *Appl. Soft Comput.*, vol. 73, pp. 607–622, Dec. 2018.

[50] F. Mohammadi and H. Abdi, "A modified crow search algorithm (MCSA) for solving economic load dispatch problem," *Appl. Soft Comput.*, vol. 71, pp. 51–65, Oct. 2018.

[51] M. Gholamghasemi, E. Akbari, M. B. Asadpoor, and M. Ghasemi, "A new solution to the non-convex economic load dispatch problems using phasor particle swarm optimization," *Appl. Soft Comput.*, vol. 79, pp. 111–124, Jun. 2019.

[52] X. Chen, "Novel dual-population adaptive differential evolution algorithm for large-scale multi-fuel economic dispatch with valve-point effects," *Energy*, vol. 203, Jul. 2020, Art. no. 117874.

[53] V. Kansal and J. S. Dhillon, "Emended salp swarm algorithm for multi-objective electric power dispatch problem," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106172.

[54] J. X. V. Neto, G. Reynoso-Meza, T. H. Ruppel, V. C. Mariani, and L. D. S. Coelho, "Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution," *Int. J. Electr. Power Energy Syst.*, vol. 84, pp. 13–24, Jan. 2017.

[55] T. Jayabarathi, T. Raghunathan, B. R. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy*, vol. 111, pp. 630–641, Sep. 2016.

[56] G. Yuan and W. Yang, "Study on optimization of economic dispatching of electric power system based on hybrid intelligent algorithms (PSO and AFSA)," *Energy*, vol. 183, pp. 926–935, Sep. 2019.

[57] H. Barati and M. Sadeghi, "An efficient hybrid MPSO-GA algorithm for solving non-smooth/non-convex economic dispatch problem with practical constraints," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 1279–1287, Dec. 2018.

[58] D. Santra, A. Mukherjee, K. Sarker, and S. Mondal, "Dynamic economic dispatch using hybrid metaheuristics," *J. Electr. Syst. Inf. Technol.*, vol. 7, no. 1, p. 3, Dec. 2020.

[59] M. J. Morshed and A. Asgharpour, "Hybrid imperialist competitive-sequential quadratic programming (HIC-SQP) algorithm for solving economic load dispatch with incorporating stochastic wind power: A comparative study on heuristic optimization techniques," *Energy Convers. Manage.*, vol. 84, pp. 30–40, Aug. 2014.

[60] X. Han, L. Quan, and X. Xiong, "A modified gravitational search algorithm based on sequential quadratic programming and chaotic map for ELD optimization," *Knowl. Inf. Syst.*, vol. 42, no. 3, pp. 689–708, Mar. 2015.

[61] C. Takeang and A. Aurasopon, "Multiple of hybrid lambda iteration and simulated annealing algorithm to solve economic dispatch problem with ramp rate limit and prohibited operating zones," *J. Electr. Eng. Technol.*, vol. 14, no. 1, pp. 111–120, Jan. 2019.

[62] A. R. Al-Roomi and M. E. El-Hawary, "Economic load dispatch using hybrid MpBBO-SQP algorithm," in *Nature-Inspired Computation in Engineering* (Studies in Computational Intelligence), vol. 637, X.-S. Yang, Ed. Cham, Switzerland: Springer, 2016, pp. 217–250.

[63] G. Pradhan and P. D. Dewangan, "Solving optimal load dispatch problem using enhanced BAT optimization algorithm," in *Proc. Innov. Power Adv. Comput. Technol.*, Apr. 2017, pp. 1–6.

[64] A. Kavousi-Fard and A. Khosravi, "An intelligent θ-modified bat algorithm to solve the non-convex economic dispatch problem considering practical constraints," *Int. J. Electr. Power Energy Syst.*, vol. 82, pp. 189–196, Nov. 2016.

[65] G. Y. G. Tchapda, Z. Wang, and Y. Sun, "Application of improved particle swarm optimization in economic dispatch of power system," in *Proc. 10th Int. Symp. Comput. Intell. Des. (ISCID)*, Dec. 2017, pp. 500–503.

[66] H. Liang, Y. Liu, Y. Shen, and F. Li, "A multiobjective chaotic bat algorithm for economic and emission dispatch," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 4684–4689.

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

[68] A.J. Wood and B.F. Wollenberg, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2012.

[69] M. Pradhan, P. K. Roy, and T. Pal, "Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 2015–2025, Dec. 2018.

[70] C. Gan, W. Cao, M. Wu, and X. Chen, "A new bat algorithm based on iterative local search and stochastic inertia weight," *Expert Syst. Appl.*, vol. 104, pp. 202–212, Apr. 2018.

[71] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperat. Strategies Optim.*, vol. 284, pp. 65–74, 2010.

[72] X.-S. He, W.-J. Ding, and X.-S. Yang, "Bat algorithm based on simulated annealing and Gaussian perturbations," in *Neural Computing Application*. London, U.K.: Springer-Verlag, 2013.

[73] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, vol. 1, Dec. 2004, pp. 111–116.

[74] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, "Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection," *Evol. Comput.*, vol. 20, no. 3, pp. 349–393, Sep. 2012.

[75] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. Congr. Evol. Comput.*, vol. 1, Jun. 2004, pp. 325–331.

[76] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Syst. Appl.*, vol. 42, nos. 17–18, pp. 6350–6364, Oct. 2015.

[77] B. Zhu, W. Zhu, Z. Liu, Q. Duan, and L. Cao, "A novel quantum-behaved bat algorithm with mean best position directed for numerical optimization," *Comput. Intell. Neurosci.*, vol. 2016, Dec. 2016, Art. no. 6097484.

[78] X. Huang, C. Li, Y. Pu, and B. He, "Gaussian quantum bat algorithm with direction of mean best position for numerical function optimization," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–18, Nov. 2019.

[79] L. dos Santos Coelho and B. A. de Meirelles Herrera, "Quantum Gaussian particle swarm optimization approach for PID controller design in AVR system," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2008, pp. 3708–3713.

[80] H. Chen, Y. Xu, M. Wang, and X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," *Appl. Math. Model.*, vol. 71, pp. 45–59, Jul. 2019.

[81] Y. Xu, H. Chen, A. A. Heidari, J. Luo, Q. Zhang, X. Zhao, and C. Li, "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Syst. Appl.*, vol. 129, pp. 135–155, Sep. 2019.

[82] M. A. Buchari, S. Mardiyanto, and B. Hendradjaya, "Implementation of chaotic Gaussian particle swarm optimization for optimize Learning-to-Rank software defect prediction model construction," *J. Physics: Conf. Ser.*, vol. 978, Mar. 2018, Art. no. 012079.

[83] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[84] Q. Liu, J. Li, L. Wu, F. Wang, and W. Xiao, "A novel bat algorithm with double mutation operators and its application to low-velocity impact localization problem," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103505.

[85] A. Haghrah, M. Nazari-Heris, and B. Mohammadi-ivatloo, "Solving combined heat and power economic dispatch problem using real coded genetic algorithm with improved Mühlenbein mutation," *Appl. Thermal Eng.*, vol. 99, pp. 465–475, Apr. 2016.

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

[87] M. S. P. Subathra, S. E. Selvan, T. A. A. Victoire, A. H. Christinal, and U. Amato, "A hybrid with cross-entropy method and sequential quadratic programming to solve economic load dispatch problem," *IEEE Syst. J.*, vol. 9, no. 3, pp. 1031–1044, Sep. 2015.

[88] S. S. S. Hosseini, X.-S. Yang, A. H. Gandomi, and A. Nemati, "Solutions of non-smooth economic dispatch problems by swarm intelligence," in *Adaptation and Hybridization in Computational Intelligence*. Cham, Switzerland: Springer, 2015, pp. 129–146.

[89] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, A. L. Bolaji, and A. Almomani, "Economic load dispatch problems with valve-point loading using natural updated harmony search," *Neural Comput. Appl.*, vol. 29, no. 10, pp. 767–781, May 2018.

[90] M. M. Delshad, "Backtracking search algorithm for optimal power dispatch in power system," Ph.D. dissertation, Dept. Eng., Univ. Malaya, Kuala Lumpur, Malaysia, 2016.

[91] S. Pan, J. Jian, H. Chen, and L. Yang, "A full mixed-integer linear programming formulation for economic dispatch with valve-point effects, transmission loss and prohibited operating zones," *Electr. Power Syst. Res.*, vol. 180, Mar. 2020, Art. no. 106061.

[92] S. H. A. Kaboli and A. K. Alqallaf, "Solving non-convex economic load dispatch problem via artificial cooperative search algorithm," *Expert Syst. Appl.*, vol. 128, pp. 14–27, Aug. 2019.

[93] C.-T. Su and C.-T. Lin, "New approach with a hopfield modeling framework to economic dispatch," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 541–545, May 2000.

[94] M. Modiri-Delshad and N. A. Rahim, "Solving non-convex economic dispatch problem via backtracking search algorithm," *Energy*, vol. 77, no. 1, pp. 372–381, Dec. 2014.

[95] M. Moradi-Dalvand, B. Mohammadi-Ivatloo, A. Najafi, and A. Rabiee, "Continuous quick group search optimizer for solving non-convex economic dispatch problems," *Electr. Power Syst. Res.*, vol. 93, pp. 93–105, Dec. 2012.

[96] X. Chen, B. Xu, and W. Du, "An improved particle swarm optimization with biogeography-based learning strategy for economic dispatch problems," *Complexity*, vol. 2018, pp. 1–15, Jul. 2018.

[97] K. Hussain, W. Zhu, M. N. M. Salleh, H. Ali, N. Talpur, R. Naseem, A. Ahmad, and A. Ullah, "Enhanced bat algorithm for solving nonconvex economic dispatch problem," in *Recent Advances on Soft Computing and Data Mining* (Advances in Intelligent Systems and Computing), vol. 978, N. N. Ghazali, M. Deris, and J. Abawajy, Eds. Cham, Switzerland: Springer, 2020, pp. 419–428.

[98] Q. Zhang, D. Zou, N. Duan, and X. Shen, "An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem," *Appl. Soft Comput.*, vol. 78, pp. 641–669, May 2019.

[99] M. Kumar and J. S. Dhillon, "Hybrid artificial algae algorithm for economic load dispatch," *Appl. Soft Comput.*, vol. 71, pp. 89–109, Oct. 2018.

[100] J. Xu, Y. 1. Fu, K. Yun, L. Su, F. Li, and J. Guan, "Noninferior solution grey wolf optimizer with an independent local search mechanism for solving economic load dispatch problems," *Energies*, vol. 12, p. 2274, Jan. 2019.

[101] Y.-K. Wang and X.-B. Chen, "Improved multi-area search and asymptotic convergence PSO algorithm with independent local search mechanism," *Kongzhi Yu Juece/Control. Decis.*, vol. 33, pp. 1382–1389, Aug. 2018.

[102] D. C. Secui, "A new modified artificial bee colony algorithm for the economic dispatch problem," *Energy Convers. Manage.*, vol. 89, pp. 43–62, Jan. 2015.

[103] V. Hosseinnezhad, M. Rafiee, M. Ahmadian, and M. T. Ameli, "Species-based quantum particle swarm optimization for economic load dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 63, pp. 311–322, Dec. 2014.

[104] A. K. Barisal and R. C. Prusty, "Large scale economic dispatch of power systems using oppositional invasive weed optimization," *Appl. Soft Comput.*, vol. 29, pp. 122–137, Apr. 2015.

[105] G. Xiong and D. Shi, "Orthogonal learning competitive swarm optimizer for economic dispatch problems," *Appl. Soft Comput.*, vol. 66, pp. 134–148, May 2018.

[106] A. Shukla and S. N. Singh, "Pseudo-inspired CBA for ED of units with valve-point loading effects and multi-fuel options," *IET Gener., Transmiss. Distrib.*, vol. 11, no. 4, pp. 1039–1045, Mar. 2017.

[107] D. Singh and J. S. Dhillon, "Ameliorated grey wolf optimization for economic load dispatch problem," *Energy*, vol. 169, pp. 398–419, Feb. 2019.

[108] K. O. Alawode, A. M. Jubril, L. O. Kehinde, and P. O. Ogunbona, "Semidefinite programming solution of economic dispatch problem with non-smooth, non-convex cost functions," *Electr. Power Syst. Res.*, vol. 164, pp. 178–187, Nov. 2018.

[109] W. T. El-Sayed, E. F. El-Saadany, H. H. Zeineldin, and A. S. Al-Sumaiti, "Fast initialization methods for the nonconvex economic dispatch problem," *Energy*, vol. 201, Jun. 2020, Art. no. 117635.

[110] Q. Qin, S. Cheng, X. Chu, X. Lei, and Y. Shi, "Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization," *Appl. Soft Comput.*, vol. 59, pp. 229–242, Oct. 2017.

[111] K. Bhattacharjee, A. Bhattacharya, and S. H. Dey, "Oppositional real coded reaction optimization for different economic dispatch problems," *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 378–391, Dec. 2014.

[112] B. Mandal, P. K. Roy, and S. Mandal, "Economic load dispatch using krill herd algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 57, pp. 1–10, May 2014.

[113] S. M. A. Bulbul, M. Pradhan, P. K. Roy, and T. Pal, "Opposition based krill herd algorithm applied to economic load dispatch problem," *Ain Shams Eng. J.*, vol. 9, no. 3, pp. 423–440, Sep. 2016.

[114] M. Ghasemi, S. Ghavidel, J. Aghaei, E. Akbari, and L. Li, "CFA optimizer: A new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems," *Int. Trans. Electr. Energy Syst.*, vol. 28, no. 5, p. e2536, May 2018.

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

[116] K. Bhattacharjee, A. Bhattacharya, and S. Halder nee Dey, "Chemical reaction optimisation for different economic dispatch problems," *IET Gener., Transmiss. Distrib.*, pp. 530–541, Mar. 2014.

[117] D. N. Vo, P. Schegner, and W. Ongsakul, "Cuckoo search algorithm for non-convex economic dispatch," *IET Gener., Transmiss. Distrib.*, vol. 7, pp. 645–654, Oct. 2013.

FRANÇOIS XAVIER RUGEMA received the B.S. degree in electronics and telecommunication engineering from the Kigali Institute of Science and Technology, Kigali, Rwanda, in 2005, and the M.S. degree in telecommunication engineering from the National University of Rwanda, Huye, Rwanda, in 2010. He is currently pursuing the Ph.D. degree in electrical engineering with Northeast Electric Power University, Jilin, China.

From 2009 to 2015, he was an Assistant Lecturer with the National University of Rwanda. His current research interests include stability analysis and control of grid-connected renewable generation, economic load dispatch, reactive power dispatch, and PV parameter estimation.

GANGUI YAN (Member, IEEE) received the Ph.D. degree from Tsinghua University, in 2003. He is currently a Professor and a Ph.D. Supervisor with Northeast Electric Power University, Jilin, China. His current research interests include renewable energy systems, energy storage system and power electronic technique, stability analysis, and control of grid-connected renewable generations.

SYLVERE MUGEMANYI received the B.S. degree in electronics and telecommunication engineering from the Kigali Institute of Science and Technology, Kigali, Rwanda, in 2005, and the M.S. degree in telecommunication engineering from the National University of Rwanda, Huye, Rwanda, in 2010. He is currently pursuing the Ph.D. degree in electrical engineering with Northeast Electric Power University, Jilin, China.

From 2009 to 2015, he was an Assistant Lecturer with the IPRC-North/College of Tumba, Rwanda. His research interests include intelligent information processing, power cyber physical systems, smart grid communications, and smart grid cyber security.

QI JIA (Member, IEEE) received the B.S. and M.S. degrees from Northeast Electric Power University, Jilin, China, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering. His current research interests include renewable energy generation, stability analysis, and control of grid-connected renewable generation.

SHANFENG ZHANG is currently pursuing the M.S. degree with the School of Electrical Engineering, Northeast Electric Power University, Jilin, China. His current research interests include renewable energy generation and new energy generation actively participates in grid frequency regulation.

CHRISTOPHE BANANEZA received the B.S. degree in electrical power engineering from the National University of Rwanda, Huye, Rwanda, in 2011, and the M.S. degree in electrical engineering from the College of Energy and Electrical Engineering, Hohai University, Nanjing, China, in 2019.

He is currently an Assistant Lecturer with the IPRC-North/College of Tumba, Rwanda. His research interest includes power systems.