Enhanced Differential Crossover and Quantum Particle Swarm Optimization for IoT Applications

SHEETAL N. GHORPADE[1], **MARCO ZENNARO**[2], (Senior Member, IEEE),
BHARAT S. CHAUDHARI[3], (Senior Member, IEEE),
RASHID A. SAEED[4], (Senior Member, IEEE), **HESHAM ALHUMYANI**[4],
AND S. ABDEL-KHALEK[4]

[1]RMD Sinhgad School of Engineering, Savitribai Phule Pune University, Pune 411058, India
[2]Science, Technology and Innovation Unit, The Abdus Salam International Centre for Theoretical Physics, 34151 Trieste, Italy
[3]School of Electronics and Communication Engineering, MIT World Peace University, Pune 411037, India
[4]Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

Corresponding author: Sheetal N. Ghorpade (sn_ghorpade@yahoo.com)

ABSTRACT An optimized design with real-time and multiple realistic constraints in complex engineering systems is a crucial challenge for designers. In the non-uniform Internet of Things (IoT) node deployments, the approximation accuracy is directly affected by the parameters like node density and coverage. We propose a novel enhanced differential crossover quantum particle swarm optimization algorithm for solving nonlinear numerical problems. The algorithm is based on hybrid optimization using quantum PSO. Differential evolution operator is used to circumvent group moves in small ranges and falling into the local optima and improves global searchability. The cross operator is employed to promote information interchange among individuals in a group, and exceptional genes can be continued moderately, accompanying the evolutionary process's continuance and adding proactive and reactive features. The proposed algorithm's performance is verified as well as compared with the other algorithms through 30 classic benchmark functions in IEEE CEC2017, with a basic PSO algorithm and improved versions. The results show the smaller values of fitness function and computational efficiency for the benchmark functions of IEEE CEC2019. The proposed algorithm outperforms the existing optimization algorithms and different PSO versions, and has a high precision and faster convergence speed. The average location error is substantially reduced for the smart parking IoT application.

INDEX TERMS Convergence, crossover operator, differential evolution operation, Internet of Things, optimization, particle swarm optimization, quantum computing.

I. INTRODUCTION

Optimization problem frequently occurs in real-time scenarios and one need to have efficient technique to attain the optimal solution with high convergence while dealing with a specific problem. The traditional gradient-based optimization method has limitations, and it fails to address complex optimization problems [1]. Metaheuristic algorithms are extensively utilized in solving the real life optimization problems. They are iterative and based on social behaviors or natural phenomena [2], [3]. The fundamental idea behind

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

natural evolutionary and swarm intelligence algorithms is to use mathematical models for simulating biological and physical structures in nature. The metaheuristic algorithms are comparatively efficient than the gradient based on the optimization [4]–[8]. The capability of parallel execution and disseminated features of swarm intelligence algorithms facilitates the probability of solving complex non-linear problems with innovative abilities such as flexibility, robustness, and searching capacity. However, the metaheuristic algorithm still needs to be upgraded because the convergence rate towards an optimum solution is comparatively slower. Hence, there is a need to alter and enhance exploration and exploitation abilities of the algorithms. [9]–[14].

Classical particle swarm optimization (PSO) [15], ant colony optimization (ACO) [16], grey wolf optimization (GWO) [17], Dragonfly Algorithm (DA) [18], Improved Whale Optimization (IWO) [19], Bat optimization algorithm (BOA) [20], Grass Hoffer Optimization Algorithm (GHO) [21], An enhanced bacterial foraging optimization (EBFO) [22], Gray Wolf Optimization (GWO) hybridized with Grasshopper Optimization Algorithm (GOA) and developed GWO-GOA optimization algorithm [23], and others are the few examples of swarm intelligence algorithm. These algorithms determine the optimal solution with heuristic information and can be applied to dynamic, multiobjective, and NP-hard problems.

With exponential growth in the deployment of the Internet of Things (IoT) and the advancements in supporting technologies such as cloud computing, mobile applications, and interfaces, swarm intelligence-based optimization exhibits considerable importance in dealing with the challenges faced for performance optimization by these networks. Generally, IoT deployment comprises large number of low cost and low power sensor nodes connected to the cloud servers and applications through the access points or gateways devices [24]. The important characteristics and requirements for IoT are traffic patterns and data rates, capacity and densification, coverage, energy efficient operations, localization, lower hardware complexity and cost effectiveness, and others. The IoT has a several applications such as smart cities, smart environment, utility metering, smart grid and energy, security and emergencies, retail, automotive and logistics, industrial automation and manufacturing, agriculture and farming, smart home/buildings, and real estate, health, life sciences, and wearables. Connectivity of a large number of devices in heterogeneous networks, energy consumption, node localization, routing of data packets, and security are the crucial challenges in IoT.

The IoT systems are modelled as a set of simple devices, and swarm intelligence algorithms can be used to optimize the performance. A huge amount of data is collected from IoT nodes. The analysis of such data is performed using different mechanisms employing edge computing, fog computing, and cloud computing, where swarm intelligence can be applied as a multiobjective optimization problem. This approach greatly helps in improving the performance of the networks and reducing the complexity and cost. A variety of algorithms based on swarm intelligence has been developed for wireless sensor network (WSN) routing protocols. A global positioning system (GPS) is commonly used for node localization problems. However, it is not economical and feasible due to high energy consumption. IoT node localization can be resolved as an error optimization problem using a swarm intelligence algorithm. Likewise, swarm-based optimization can be used in various ways to improve the performance of IoT networks. One of the such challenges is non-uniform deployment of IoT nodes due to mobility and because of application requirements. The mobile IoT nodes significantly improve data sensing capabilities with enhanced coverage

and lower energy consumption. However, such scenarios and topologies pose the additional challenge of maintaining the node density and coverage to satisfy the application requirements. The node density and coverage directly affect the approximation accuracy. Many of the existing IoT node localization approaches are designed on a basic disk coverage model, which is unrealistic for implementing in actual application environments. In these approaches, spatial relationships of the supervised physical characteristics, sensor node association, and network fault tolerance are ignored, and hence it fails to attain the global optimization requirements. Furthermore, these approaches did not discuss and address the optimal solutions for node density and coverage in the IoT networks. To tackle the challenges of optimizing the node density and coverage, we propose a novel enhanced differential crossover quantum particle swarm optimization (EDCQPSO) algorithm. We have used hybrid optimization using quantum PSO, differential evolution operator, and crossover operator to have proactive and reactive operations. The developed algorithm has smaller fitness values and faster convergence, and it can be used for optimization in a wide variety of IoT applications. To demonstrate the usability of algorithm in IoT, we considered car parking IoT application. Our algorithm gives lower localization error and improved precision for the higher node densities as compare to the other existing algorithms. The paper's remaining structure is organized as: Section II presents the literature study about PSO enhancements. Section III describes a quantum particle swarm optimization (QPSO). Section IV presents the development of enhanced differential crossover quantum particle swarm optimization (EDCQPSO) algorithm. Section V discusses results and performance evaluation. Section VI presents the study on EDCQPSO for IoT application, and the paper is concluded in Section VII.

II. RELATED WORK

In the recent past, several swarm intelligence approaches, and modifications have been proposed. The relevant approaches to the research undertaken are discussed here.

Tam *et al.* [25] proposed a hybrid approach using fuzzy clustering and PSO to reduce network interruption. This hybrid approach is executed repetitively until the construction of optimal sensor topology. Energy consumption is reduced by this method and improves connectivity from cluster head to base station and other nodes to cluster head. Optimized minimal spanning tree topology control using PSO is proposed in [26] to overcome low coverage drawbacks in traditional approaches. It converges to the condensed topology uniformly with lesser energy consumption. Swarm-based modified bat optimization algorithm [27] is utilized for calculating the precision of node localization problems. It improves localization and attains fast convergence. Discrete PSO and minimal spanning tree-based topology scheme with multiobjective constraints [28] consider

the distance among the nodes, coverage of each edge, and their residual energies. Ghorpade *et al.* [29] developed a binary grey wolf optimization topology control technique which works on active-inactive schedules of sensor nodes and presents a fitness function to minimize number of active nodes for achieving extended lifetime. This algorithm achieves maximum coverage and connectivity. Ant colony optimization (ACO) is combined with local search for node deployment in WSN by considering cost reliability as a constraint [30]. Simulations results have proven that the proposed approach generates improved quality than the greedy algorithm.

Although the PSO-based node localization approach [31] is computationally effective, there is not much improvement in the localization error. Bat algorithm-based localization [32] replicates bats' behavior using echolocation for the prey hunting during the darkness. In this approach, bat calculations are concurred along with a growing of chemotactic bacterial sponging control for improving the constraint accuracy in the lesser time. A multiobjective GWO technique for accurate localization of IoT nodes [33] is developed for achieving the higher efficiency with smaller number of the anchors. The objective functions have included the distance and topological constraints. Kumar *et al.* [34] have proposed a combined hybrid particle swarm optimization (HPSO) technique with the biogeography based optimization (BBO), which is also a two-step location estimation for minimizing location errors. RSSI is used as an input parameter, and the output weight is used for weighted centroid localization. These methods are inclined towards lower accuracy in case of unevenness between the identified nodes. A novel multiobjective optimization agent using particle swarm GWO and inverse fuzzy ranking is proposed in [35]. The developed enhanced PSGWO model is utilized for population and multi criteria based soft computing algorithms. This bio-inspired optimization technique is used to calculate low energy optimum path for IoT networks.

An IoT-based range-based localization for smart city applications is proposed for accurate and low-cost localization [36]. The extreme learning machine (ELM), fuzzy system, and modified swarm intelligence is used to develop hybrid optimized fuzzy threshold ELM (HOFTELM) algorithm for the localization of elderly persons in smart cities. The algorithm outperforms existing techniques with average location error ratio (ALER) and computationally efficient. Although Van [37] has demonstrated that PSO is not an algorithm for global optimization; however, for the improvement in the performance of PSO, Sun *et al.* [38] have proposed quantum PSO (QPSO) by combining quantum theory with PSO. QPSO algorithm guarantees the global optimal solution for the infinite number of search iterations. However, it is impractical since any algorithm permits only finite for the best solution in real-time applications. Moreover, QPSO falls into the local optima resulting the slower convergence. Various approaches have been proposed for the improvement in the convergence speed and global optima. Liang *et al.* [39]

has developed comprehensive learning quantum PSO using the learning approach. The information from other particles is utilized for updating particle velocity. This approach allows the swarm's diversity to be well-maintained for discouraging convergence occurring at an early stage. Parallel diversity-controlled quantum particle swarm optimization (PDQPSO) [40] is proposed to enhance efficiency and get rid of early convergence. This approach aims to use the parallel technique to increase the population's diversity and reduce the algorithm run time. It achieves promising performance and reduced computational time for most of the test functions. LDS Coelho [41] incorporated a chaotic mutation operator with Quantum PSO. Simulations are carried out for solving optimization problems and it demonstrates improved performance. Shanshan Tu *et al.* [42] proposed updating of crossover parameter to improve the quantum PSO performance and global search abilities. An approach proposed in [43] combines QPSO with Cauchy mutation operator (QPSO-CD) which adds extended capabilities for global hunt.

Quantum based PSO with opposition based learning and generalized opposition based learning (CSQPSO) [44] improves the exploitation and also supports exploration. However, parallel improvement in global exploration ability and convergence speed is a challenging task. While avoiding local optima, the convergence speed of an algorithm may get reduced.

Accordingly, the QPSO algorithm is requires precise design for the real-world optimization problem. For the swarm intelligence algorithms, balancing the global and local search capabilities is a crucial problem. In PSO, when we think of exploration, the fast convergence features lead to early convergence. If the focus is on gain, then the single exploration approach of particle swarm has unsatisfactory convergence accuracy. For multiobjective PSO, the regular updates in global solutions also increase exploration and progress.

For improving QPSO, sufficient data about each particle its own and optimal global position should be utilized by choosing an appropriate technique. Our research has incorporated a differential evolution into QPSO for improving the population diversity and avoid local optima. It uses competition and cooperation among individuals to solve optimization problems. Additionally, we have introduced a crossover operator with QPSO. The cross operations will promote the information interchange among individuals in a group, and those exceptional genes can be continued moderately, accompanying the continuance of the evolutionary process. The value of crossover probability plays a vital role in an algorithm's searchability and convergence speed. Ultimately groups can progress in the desired route. Enhanced differential crossover QPSO algorithm aims to improve control of exploring and exploiting hunts by considering adjacent relationships between the particles by a linear increase in the connectivity of the swarm's topology and carrying out regulating mechanisms.

III. QUANTUM PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) [15], is based on the concept of swarm's social behavior that results in a group of nodes spread in a search space. It starts with initial population of swarm, called as nodes which explores the arbitrary position p_{lm} and velocity v_{lm} in m – dimensional hyperspace for node l. Every node is determined by using an objective function $f(p_1, p_2, p_3, \ldots, p_m)$ where $f : R^m \to R$, represents the number of sensors. The PSO tries for wide coverage for a given connectivity value. Then, PSO guides each node for the position updates in the search space by considering the obtained global solution and best fitness values. The position update process is continued until the desirable globally best solution is attained or performed the given target of iterations.

To determine the next position of a node in each iteration, velocity is updated by using (1), and position is updated by using (2)

$$V_{lm}^{t+1} = V_{lm}^t + a_1 b_1 \left(Pbest_{lm}^t - P_{lm}^t \right)$$
$$+ a_2 b_2 \left(Pgbest_{lm}^t - P_{lm}^t \right) \qquad (1)$$
$$P_{lm}^{t+1} = P_{lm}^t + V_{lm}^{t+1} \qquad (2)$$

$l \cdot m = 1, 2, 3, \ldots, M + N \cdot l \cdot m$ represents index of the sensor P_{lm}^t and V_{lm}^t are the mth position component and velocity of lth sensor in t^{th} iteration. b_1 and b_2 are the random numbers such that $0 \leq b_1, b_2 \leq 1$. $Pbest_{lm}^t$ and $Pgbest_{lm}^t$ are the best and global best positions of lth sensor and the swarm. a_1 and a_2 are confidence nodes as in perception and community behavior. In the process of estimation, the sensor will take the weighted average position, which is determined using

$$W_{lm}^t = \frac{a_1 (b_1)_{lm}^t Pbest_{lm}^t + a_2 (b_2)_{lm}^t Pgbest_{lm}^t}{a_1 (b_1)_{lm}^t + a_2 (b_2)_{lm}^t},$$
$$1 \leq m \leq M \qquad (3)$$

PSO inclines to get stuck into local optima while tackling the composite problems. For improvement of PSO Sun *et al.* [38] have proposed quantum PSO (QPSO). The quantum particle swarm optimization algorithm assumes that the node swarm system satisfies quantum mechanics' elementary proposition. Node l moves in the δ probable well centered at the point 'W' in m^{th} dimension with basic quantum actions characteristic and its state can be described by

$$\psi \left(P_{lm}^{t+1} \right) = \frac{1}{\sqrt{C_{lm}^t}} * \exp \left(\frac{-\left| P_{lm}^{t+1} - W_{lm}^t \right|}{C_{lm}^t} \right) \qquad (4)$$

where C is the characteristic length of probable well δ and is associated with speed of the convergence and searchability. The probability density function of node l is as given in

$$Q \left(P_{lm}^{t+1} \right) = \frac{1}{\sqrt{C_{lm}^t}} * \exp \left(\frac{-2\left| P_{lm}^{t+1} - W_{lm}^t \right|}{C_{lm}^t} \right) \qquad (5)$$

To obtain the node's position, it is collapsed into a classical state from the quantum state. The position of the node is determined by using

$$P_{lm}^{t+1} = W_{lm}^t \pm \frac{C_{lm}^t}{2} \ln \frac{1}{r_{lm}^t} \qquad (6)$$

where W is the node motion center and is called the attractor of the node. r is lies between 0 to 1 with a uniform distribution function. Parameter C is determined by using

$$C_{lm}^t = 2\gamma \left\| L_m^t - P_{lm}^t \right\| \qquad (7)$$
$$L_m^t = \frac{\sum_{l=1}^{N} Pbest_{lm}^t}{N} \qquad (8)$$

γ is the contraction and expansion factor, which has to be decreased while running the algorithm. $L^t = \{L_1^t, L_2^t, \ldots, L_m^t\}$ is mean optimal position, representing mean optimal position of all nodes.

IV. ENHANCED DIFFERENTIAL CROSSOVER QUANTUM PARTICLE SWARM OPTIMIZATION

In QPSO, every node holds the weighted mean position obtained by considering earlier individual and group optimal positions as a desirability point. Such a method has the advantage of simple calculations, but this holding weighted mean position has two drawbacks; in addition to own learning experience, the position of every node is subject to the group's historical optimal position. In addition to this, the possible dispersal space of each node's attraction point progressively declines during an algorithm's development process. It leads to a swift decay of diversity reducing capability while handling the multiobjective and composite optimization problems. It ultimately reduces ability to jump out of local optimization in the later stage.

Since the algorithm gets in to local optima in finishing stage, indicating that individual and global positions of the particles are almost adjacent to each other or maybe coincident. Hence, for improving the QPSO algorithm's performance, adequate information about the nodes' individual and global optimal positions can be used by choosing a suitable technique. To overcome this drawback, a differential evolution operator can be incorporated into QPSO. A differential evolutionary algorithm [45] is proposed on the population differences. It is based on the use of competition and cooperation among individuals for solving optimization problems. The differential evolution operator improves the population diversity as well as jumping out of local optima. Position update in QPSO is performed by using

$$U_{lm}^t = \chi Pbest_{lm}^t + (1 - \chi) gbest_m^t \qquad (9)$$
$$AVbest_m = \frac{1}{N} \sum_{l=1}^{N} Pbest_{lm}^t \qquad (10)$$
$$P_{lm}^{t+1} = W_{lm}^t \pm \gamma \left| Avbest_m - P_{lm}^t \right| \ln \left(\frac{1}{r_{lm}^t} \right) \qquad (11)$$

χ is lies in between 0 and 1. W_{lm}^t is arbitrary position amid *Pbest* and *gbest*. By combining (3) and (5), the position

evolution equation changes to (12) as given below,

$$P_{lm}^{t+1} = \chi \left(Pbest_{lm}^t - gbest_m^t \right) + gbest_m^t$$
$$\pm \gamma \left| AVbest_m - P_{lm}^t \right| \ln \left(\frac{1}{r_{lm}^t} \right) \quad (12)$$

Let a and b be the nodes in the existing swarm distinct from l then the position difference between them is,

$$\emptyset = P_b - P_a \quad (13)$$

Substitute $\emptyset$ to replace the difference $Pbest_{lm}^t - gbest_m^t$ of (12) and randomness can be increased by adding a random number $(1 - \chi)$ to the second term $gbest_m^t$ of (12). The new evolution equation is

$$P_{lm}^{t+1} = \chi \phi_m + (1 - \chi) gbest_m^t$$
$$\pm \gamma \left| AVbest_m - P_{lm}^t \right| \ln \left(\frac{1}{r_{lm}^t} \right) \quad (14)$$

Differential evolution operator introduced in (14) helps avoid group moves in small range, hence falls in to the local optima, as favorable for enhancing the global searchability.

In the next phase, we have introduced a crossover operator with QPSO. These cross operations will promote the information interchange among individuals in a group, and those exceptional genes can be continued moderately, accompanying the continuance of the evolutionary process. Ultimately groups can progress in the desired route. The position estimate P_l^{t+1} of node l is generated by using (3), (7), (8), and (14). Later, the estimated position P_l^{t+1} and individual optimal position $Pbest_l^t$ are separated for the generation of the test position $Y_l^m = \{y_{l1}^t, y_{l2}^t, \ldots, y_{lm}^t\}$ the cross equation is,

$$Y_{lm}^{t+1} = \begin{cases} P_{lm}^{t+1}, & (rand)_m < c, m = m_{rand} \\ Pbest_{lm}^t, & otherwise \end{cases} \quad (15)$$

where $(rand)_m$ is random number with uniform distribution such that $(rand)_m \in [0, 1]$ and c is the crossover probability. Whereas m_{rand} is randomly and uniformly generated integer on $[1, M]$.

Lastly, updated optimal position is given by

$$Pbest_{lm}^{t+1} = \begin{cases} Y_{lm}^{t+1}, & f\left(Y_{lm}^{t+1}\right) < f\left(Pbest_{lm}^t\right) \\ Pbest_{lm}^t, & otherwise \end{cases} \quad (16)$$

$f(*)$ is a compatible cost function. The value of the crossover probability plays a vital role in an algorithm's searchability and convergence speed. Smaller values of probability enable individuals to hold further information and preserve higher diversity of the group, helps during the global exploration. On the contrary, the larger value of the probability impulses individuals to acquire additional experimental information in the group, consequently accelerating an algorithm's convergence speed.

By considering the crucial role of crossover probability c, it is directly encoded into each node for achieving adaptive control. Node l in given population is defined in

$$P_l^t = \{p_{l1}^t, p_{l2}^t, \ldots, p_{lm}^t, c_l^t\} \quad (17)$$

Crossover probability for every node in the population is updated by using

$$c_l^{t+1} = \begin{cases} rand_m(0, 1), & rand_m(0, 1) < \propto \\ c_l^t, & otherwise \end{cases} \quad (18)$$

α is the updated probability of parameter c. For ease of operations, we have introduced an additional binary vector B_l^{t+1} for every node l.

$$B_l^{t+1} = \left\{ b_{l1}^{t+1}, b_{l2}^{t+1}, \ldots, b_{lm}^{t+1} \right\} \quad (19)$$

$$b_{lm}^{t+1} = \begin{cases} 1, & rand_m(0, 1) < c_l^{t+1}, m = m_{rand} \\ 0, & otherwise \end{cases} \quad (20)$$

$$Z_l^{t+1} = \frac{1}{M} \sum_{l=1}^{M} b_{lm}^{t+1} \quad (21)$$

By ignoring the influence of m_{rand}, Z_l^{t+1} follows binomial distribution with M parameters and probability c_l^{t+1}. The probability c_l^{t+1} is calculated by using

$$c_l^{t+1} = \begin{cases} B_l^t Z_l^{t+1} + \left(1 - B_l^t\right) c_l^t, & f\left(Z_l^{t+1}\right) < f\left(c_l^t\right) \\ c_l^t, & otherwise \end{cases} \quad (22)$$

Random number B_l^t lies between $0.9 \leq B_l^t \leq 1$. Additionally, extension coefficient λ is designed so that with the increase in the number of iterations, it decreases linearly.

$$\lambda = \lambda_{max} - \frac{t}{T} * (\lambda_{max} - \lambda_{min}) \quad (23)$$

where T represents the maximum iterations to be attained. Enhanced DCQPSO algorithms process flow is shown in the Fig. 1

The steps of the algorithm are as given below:

1. Set $t = 0$, initialize current position P_l^0 of every node in the swarm, and assemble $c_l^0 = P_l^0$. Also, set other relevant parameters.
2. Determine the mean optimal position of the node swarm by using (10).
3. For every node $l, (1 \leq l \leq N)$ in the group, perform Step 4 to Step 7.
4. Use (4) to introduce differential evolution operator for updating node position.
5. Establish the crossover operator and estimate the position by considering an updated position in the previous step and initiate the test position by using (15).
6. At the test position, determine the adaptive value of every node's dimension and use (22) to update the crossover probability.
7. Update the individual optimal position of the nodes by using (16)

V. EXPERIMENTAL SETUP AND PERFORMANCE ANALYSIS

We initially present the comparison of proposed algorithm, EDCQPSO, with others through 30 classic benchmark functions in IEEE CEC2017 [46], as shown in Table 1. The performance of our algorithm on benchmark functions was verified.
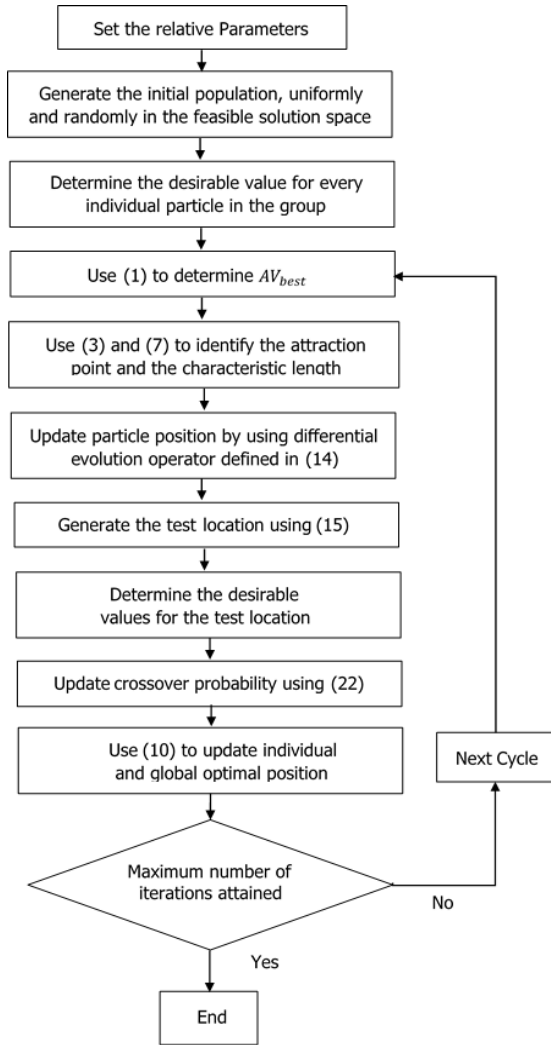
FIGURE 1. Process flow of EDCQPSO algorithm.

TABLE 1. CEC2017 test functions [46].

Function	Function name	Optimum
CEC2017 unimodal functions (UF)		
CE01	Shifted and rotated bent cigar function	100
CEE02	Shifted and rotated sum of different power function	200
C03	Shifted and rotated Zakharov function	300
CEC2017 multimodal functions (MF)		
CE04	Shifted and rotated Rosenbrock's function	400
CE05	Shifted and rotated Rastrigin's function	500
CE06	Shifted and rotated expanded Scaffer's F6 function	600
CE07	Shifted and rotated Lunacek bi-Rastrigin function	700
CE08	Shifted and rotated noncontinuous Rastrigin's function	800
CE09	Shifted and rotated Levy function	900
CE10	Shifted and rotated Schwefel's function	1000
CEC2017 hybrid functions (HF)		
CE11	Hybrid Function1 ($N = 3$)	1100
CE12	Hybrid Function2 ($N = 3$)	1200
CE13	Hybrid Function3 ($N = 3$)	1300
CE14	Hybrid Function4 ($N = 4$)	1400
CE15	Hybrid Function5 ($N = 4$)	1500
CE16	Hybrid Function6 ($N = 4$)	1600
CE17	Hybrid Function6 ($N = 5$)	1700
CE18	Hybrid Function6 ($N = 5$)	1800
CE19	Hybrid Function6 ($N = 5$)	1900
CE20	Hybrid Function16 ($N = 6$)	2000
CEC2017 composition functions (COMPOSITE FUNCTION)		
CE21	Composite Function 1 ($N = 3$)	2100
CE22	Composite Function 2 ($N = 3$)	2200
CE23	Composite Function 3 ($N = 4$)	2300
CE24	Composite Function 4 ($N = 4$)	2400
CE25	Composite Function 5 ($N = 5$)	2500
CE26	Composite Function 6 ($N = 5$)	2600
CE27	Composite Function 7 ($N = 6$)	2700
CE28	Composite Function 8 ($N = 6$)	2800
CE29	Composite Function 9 ($N = 3$)	2900
CE30	Composite Function 10 ($N = 3$)	3000

EDCQPSO is also compared with different PSO versions using ten benchmark functions in IEEE CEC2019, as shown in Table 2. We used the Friedman test [47] and Wilcoxon symbolic rank test [48] for optimal results on the benchmarks and statistical analysis. To analyze the proposed algorithm's performance, we have used classic benchmark functions from IEEE CEC2017 [49] and IEEE CEC2019 [50]. IEEE CEC2017 is composed of three unimodal (C01-C03), seven multimodal (C04-C10), ten hybrid (C11-C20), and ten composite (C21-C30) functions. IEEE CEC2019 is composed of 10 functions (C31-C40). The benchmark functions of IEEE CEC2017 is as given in Table 1 and used for comparing our algorithm with other swarm intelligence algorithms.

Simulations are carried out in MATLAB with identical parameter settings for comparison of the results. For performance analysis, the Friedman test [47] is used to thoroughly evaluate all algorithms' optimal results on the benchmark functions. To classify the chosen algorithms' mean performance, the average sort value (ASV) is attained through statistical comparisons.

Additionally, we have implemented the paired Wilcoxon symbolic rank test [48] for statistical assessment to identify variance among two samples with 5% level of significance. The statistical results are shown in Table 5 and Table 8. In these tables, symbol '+' specifies that with 95% inevitability the null hypothesis is rejected (Avg. value < 0.05), the symbol '-' indicates that the null hypothesis is rejected (Avg. value < 0.05) and symbol '=' represents that there is no statistical variance among the pairwise algorithms (Avg. value ≥ 0.05).

A. COMPARISONS OF THE EDCQPSO WITH OTHER SWARM ALGORITHMS

We have compared the performance of EDCQPSO with six recently developed swarm intelligence algorithms. These

TABLE 2. CEC2019 test functions [46].

Function	Function name	$F_i^* = F_i(y^*)$	D	Search Range
CE31	Storns Chebyshev polynomial fitting problem	1	9	[−8192, 8192]
CE32	Inverse Hilbert matrix problem	1	16	[−16834, 16834]
CE33	Lennard-Jones minimum energy cluster	1	18	[−4, 4]
CE34	Rastrigin's function	1	10	[−100, 100]
CE35	Griewank's function	1	10	[−100, 100]
CE36	Weierstrass function	1	10	[−100, 100]
CE37	Modified Schwefel's function	1	10	[−100, 100]
CE38	Expanded Schaffer's F6 function	1	10	[−100, 100]
CE39	Happy Cat function	1	10	[−100, 100]
CE40	Ackley function	1	10	[−100, 100]

TABLE 3. Simulation parameters (1).

Algorithm	Other Parameters
EDCQPSO	$B_i^t \in [0, 1]$. $\chi \in (0, 1)$,
DA [18]	$w \in [0.9\ 0.2], s = 0.1, a = 0.1,$ $c = 0.7, f = 1, e = 1$
GWO [17]	$a \in [2, 0]$
IWO [19]	$a_1 \in [2, 0], a_2 \in [−2, −1], b = 1$
GHO [21]	$l = 1.5, f = 0.5, c_{max} = 1; c_{min} = 0.00004$
EBFO [22]	$\Delta \in [−1, 1]$
GWO-GOA [23]	$N = 30, C_f = 2500, L_f = 0.94$

algorithms are; GWO [17], DA [18], IWO [19], GHO [21], EBFO [22], and GWO-GOA [23]. All the algorithms are simulated in the same environment on the benchmark functions of CEC2017 by setting parameters required parameters for each algorithm. Details of parameters chosen for every algorithm are presented in Table 3. Max. number of iterations to be attained are 2000 with population's size of 40 for each algorithm.

The comparison of mean values and standard deviation after thirty iterations on thirty benchmark functions are listed. Table 4 shows that EDCQPSO ranks first, followed sequentially by GWO-GOA, GHO, GWO, IWO, EBFO, and DA, based on overall rank for CE01-CE30 functions of CEC2017 [46]. On three unimodal test functions (CE01-CE03), EDCQPSO performs better than other algorithms. The multimodal test functions (CE04–CE10) EDCQPSO are highly comparable for CE04, CE06, and CE09. However, GWO-GOA outperforms all the other algorithms on CE10.

It can also be observed that results obtained by GWO-GOA are competing closely to multimodal EDCQPSO, but the trend changes for hybrid and composite functions. On the ten hybrid test functions (CE11–CE20), excluding CE14, EDCQPSO attains the optimal results. For the hybrid functions CE11, CE12, CE13, CE15, CE17, and CE19, EDCQPSO performs outstandingly compared to other algorithms. Lastly, for the ten composition functions (CE21–CE30),

EDCQPSO outperforms the remaining algorithms. except for CE24. It gives the best optimal value for CE30. The performance improvement is due to the proposed differential evolution operator which escapes group changes in smaller range and falling in to local optima, promoting global searchability.

Proposed algorithm shows an average improvement of 87.65%, 81.29%, 76.98%, 70.79%, 69.68% and 66.38% in comparison with DA, EBFO, IWO, GWO, GHO and GWO-GOA respectively. The convergence progression of all the above comparative algorithms for sample functions from CEC2017 is shown in Fig.2. The logarithmic scale of optimal objective function value on standard test functions is evaluated by considering a population size of 40 with 2000 iterations.

The proposed algorithm shows appropriate behavior until maximum iterations on most tested functions throughout the evolution process, whereas others methods get stuck into local minima.

Approximately after 600 iterations, EDCQPSO converges rapidly towards the global optimum because the cross operations used in the proposed algorithm encourage information interchange among individuals in a group. Those exceptional genes get continued moderately, accompanying the continuance of the evolutionary process. The convergence rate of GWO-GOA for unimodal is also comparable. However, in the case of hybrid and composite function, it converges fast for initial iterations, and for higher iterations, it moves around local optima.

On an average for unimodal, multimodal, hybrid and composite function EDCQPSO performs 34.94%, 34.39%, 31.01%, 23.18%, 19.37% and 16.27% faster than DA, EBFO, IWO, GWO, GHO and GWO-GOA respectively.

The results shows that EDCQPSO performs better as compared to other five algorithms for most CEC2017 test functions.

B. COMPARISONS OF THE EDCQPSO WITH OTHER VERSIONS OF PSO

We have also compared performance of EDCQPSO with PSO and its versions. These algorithms are; PSO [15], PDQPSO [40], QPSO – CD [42], CLQPSO [43], and CSQPSO [44]. All these algorithms are simulated in the same environment on the IEEE CEC2019 benchmark functions by setting parameters the same as that of the original paper. Details of parameters chosen for every algorithm are presented in Table 6.

The maximum number of iterations to be attained are 2000 for population size of 40 for each algorithm. For all the PSO algorithm variants, convergence rate, as shown in Fig. 3, is analyzed in a logarithmic scale of best objective function value on test functions. EDCQPSO reaches the optimal solution with high precision and faster convergence speed.

All the results and statistical analysis shows that the proposed algorithm improves the solution quality and

TABLE 4. Mean and standard deviation of different algorithms for IEEE CEC2017.

|		EDCQPSO	DA	GWO	IWO	GWO-GOA	GHO	EBFO
CE01	Average	4.97 E03	2.52 E09	2.63 E09	1.18 E06	1.69 E05	1.25 E06	9.34 E09
	Std.Dev.	4.01 E03	2.06 E09	2.02 E09	3.26 E05	5.63 E05	1.61 E06	7.94 E09
CE02	Average	2.98 E17	5.96 E37	1.26 E31	4.63 E18	6.41 E25	1.22 E27	1.84 E43
	Std.Dev.	1.01 E18	1.97 E38	3.23 E31	1.45 E18	1.81 E26	6.70 E27	1.01 E44
CE03	Average	5.71 E03	1.04 E05	4.46 E04	1.87 E04	6.83 E03	6.88 E03	1.25 E05
	Std.Dev.	2.97 E03	2.52 E04	9.63 E03	1.38 E04	1.66 E03	3.38 E03	5.44 E04
CE04	Average	1.21 E02	1.22 E03	6.15 E02	5.07 E02	3.76 E02	5.13 E02	1.20 E03
	Std.Dev.	0.97 E01	6.47 E02	9.11 E01	2.54 E01	2.13 E01	2.36 E01	8.37 E02
CE05	Average	6.07 E02	8.74 E02	6.09 E02	7.72 E02	6.34 E02	6.45 E02	7.04 E02
	Std.Dev.	2.36 E01	8.02 E01	2.42 E01	6.30 E01	2.85 E01	3.27 E01	4.60 E01
CE06	Average	5.92 E02	6.76 E02	6.10 E02	6.63 E02	6.04 E02	6.46 E02	6.40 E02
	Std.Dev.	1.01 E00	1.24 E01	4.94 E00	1.25 E01	1.76 E01	1.89 E01	1.05 E01
CE07	Average	7.61 E02	1.07 E03	8.75 E02	1.12 E03	8.14 E02	8.83 E02	1.14 E03
	Std.Dev.	3.31 E01	7.23 E01	3.69 E01	7.80 E01	3.60 E01	5.38 E01	2.33 E02
CE08	Average	7.98 E02	1.10 E03	8.92 E02	9.93 E02	8.69 E02	9.38 E02	1.01 E03
	Std.Dev.	2.07 E01	5.14 E01	2.28 E01	3.54 E01	2.64 E01	3.76 E01	3.90 E01
CE09	Average	1.41 E03	1.32 E04	2.00 E03	7.95 E03	2.90 E03	5.53 E03	7.53 E03
	Std.Dev.	5.16 E02	4.72 E03	6.29 E02	3.01 E03	8.88 E02	4.24 E03	2.63 E03
CE10	Average	5.04 E03	7.09 E03	4.19 E03	5.70 E03	4.45 E03	5.26 E03	5.54 E03
	Std.Dev.	3.97 E02	7.87 E02	6.35 E02	7.76 E02	2.78 E02	8.68 E02	6.87 E02
CE11	Average	1.02 E03	2.73 E03	2.25 E03	1.28 E03	1.46 E03	1.39 E03	5.06 E03
	Std.Dev.	4.31 E01	9.50 E02	1.09 E03	6.76 E01	8.52 E01	9.03 E01	4.25 E03
CE12	Average	5.71 E05	5.70 E08	6.01 E07	6.08 E06	1.94 E06	1.99 E07	5.43 E08
	Std.Dev.	2.14 E05	5.21 E08	7.49 E07	3.76 E06	5.10 E06	2.25 E07	9.58 E08
CE13	Average	2.18 E04	7.80 E07	2.20 E07	1.88 E05	4.45 E04	1.65 E05	1.29 E07
	Std.Dev.	1.32 E04	1.97 E08	8.58 E07	1.12 E05	4.30 E04	1.35 E05	2.68 E07
CE14	Average	6.93 E03	1.15 E06	4.15 E05	3.83 E04	3.59 E03	3.31 E03	4.87 E05
	Std.Dev.	2.79 E03	2.03 E06	4.51 E05	3.15 E04	3.47 E03	3.19 E03	1.34 E06
CE15	Average	5.94 E03	2.27 E05	2.07 E06	7.44 E04	1.31 E04	7.36 E04	6.07 E04
	Std.Dev.	5.19 E03	3.24 E05	7.77 E06	3.98 E04	1.81 E04	4.72 E04	4.87 E04
CE16	Average	2.11 E03	4.00 E03	2.56 E03	3.13 E03	2.65 E03	2.88 E03	3.05 E03
	Std.Dev.	2.01 E02	6.16 E02	2.95 E02	3.45 E02	2.05 E02	3.64 E02	4.39 E02
CE17	Average	1.71 E03	2.89 E03	1.99 E03	2.45 E03	2.11 E03	2.25 E03	2.50 E03
	Std.Dev.	7.51 E01	3.52 E02	1.44 E02	2.86 E02	8.43 E01	2.01 E02	2.56 E02
CE18	Average	1.85 E05	6.98 E06	1.10 E06	7.07 E05	6.26 E05	7.01 E05	2.11 E06
	Std.Dev.	1.37 E05	8.23 E06	1.25 E06	5.97 E05	1.27 E05	1.05 E06	3.06 E06
CE19	Average	4.17 E03	4.87 E07	9.45 E05	4.31 E05	7.96 E04	3.65 E06	1.59 E07
	Std.Dev.	2.24 E03	6.38 E07	1.26 E06	2.90 E05	7.00 E04	3.12 E06	3.80 E07
CE20	Average	2.16 E03	2.83 E03	2.43 E03	2.78 E03	2.37 E03	2.60 E03	2.72 E03
	Std.Dev.	1.01 E02	1.90 E02	1.37 E02	2.29 E02	1.54 E02	1.77 E02	2.24 E02
CE21	Average	2.34 E03	2.66 E03	2.39 E03	2.56 E03	2.45 E03	2.43 E03	2.50 E03
	Std.Dev.	4.14 E01	7.42 E01	2.34 E01	7.41 E01	3.17 E01	3.20 E01	5.32 E01
CE22	Average	2.21 E03	7.65 E03	4.94 E03	6.16 E03	3.84 E03	5.75 E03	6.86 E03
	Std.Dev.	1.11 E00	2.10 E03	1.80 E03	1.64 E03	1.88 E02	1.86 E03	8.83 E02
CE23	Average	2.41 E03	3.28 E03	2.76 E03	3.00 E03	2.65 E03	2.80 E03	2.84 E03
	Std.Dev.	3.11 E01	1.75 E02	3.61 E01	7.99 E01	3.53 E01	3.99 E01	4.03 E01
CE24	Average	2.41 E03	3.45 E03	2.94 E03	3.20 E03	2.92 E03	2.16 E03	2.99 E03
	Std.Dev.	3.01 E01	1.62 E02	5.35 E01	1.17 E02	3.19 E01	2.98 E01	3.20 E01
CE25	Average	2.62 E03	3.17 E03	2.98 E03	2.91 E03	2.82 E03	2.93 E03	3.31 E03
	Std.Dev.	7.68 E00	1.82 E02	3.83 E01	2.02 E01	1.08 E01	2.87 E01	5.67 E02
CE26	Average	3.97 E03	8.66 E03	4.84 E03	6.62 E03	4.71 E03	5.29 E03	5.81 E03
	Std.Dev.	3.19 E02	1.60 E03	3.82 E02	1.41 E03	6.61 E02	9.69 E02	4.31 E02
CE27	Average	3.11 E03	3.56 E03	3.26 E03	3.28 E03	3.14 E03	3.24 E03	3.26 E03
	Std.Dev.	1.94 E-04	1.72 E02	2.09 E01	4.33 E01	9.38 E00	1.99 E01	3.22 E01
CE28	Average	3.13 E03	3.80 E03	3.42 E03	3.23 E03	3.26 E03	3.27 E03	4.20 E03
	Std.Dev.	4.97 E01	2.01 E02	7.98 E01	2.62 E01	2.68 E01	2.84 E01	9.42 E02
CE29	Average	3.51 E03	5.45 E03	3.75 E03	4.33 E03	3.88 E03	4.14 E03	4.19 E03
	Std.Dev.	1.08 E02	6.82 E02	1.49 E02	3.27 E02	1.31 E02	1.90 E02	3.11 E02
CE30	Average	1.14 E04	3.89 E07	7.73 E06	1.46 E06	8.61 E05	7.67 E06	1.17 E06
	Std.Dev.	2.99 E03	2.96 E07	7.45 E06	8.15 E05	5.60 E05	5.76 E06	2.04 E06

convergence behaviour. On an average for the test functions in IEEE CEC 2019 EDCQPSO performs 65.05%, 53.77%, 53.72%, 48.19%, and 26.58% faster than PSO, PDQPSO, CLQPSO, QPSO-CD and CSQPSO, respectively. The mean, standard deviation, and rank of the algorithm after ten

iterations on ten benchmark functions of IEEE CEC2019 are compared and are shown in Table 7.

The outcomes of Table 8 prove that based on overall rank on the CE31-CE40 functions of CEC2019, EDCQPSO ranks first and then followed sequentially by CSQPSO, QPSO-CD,
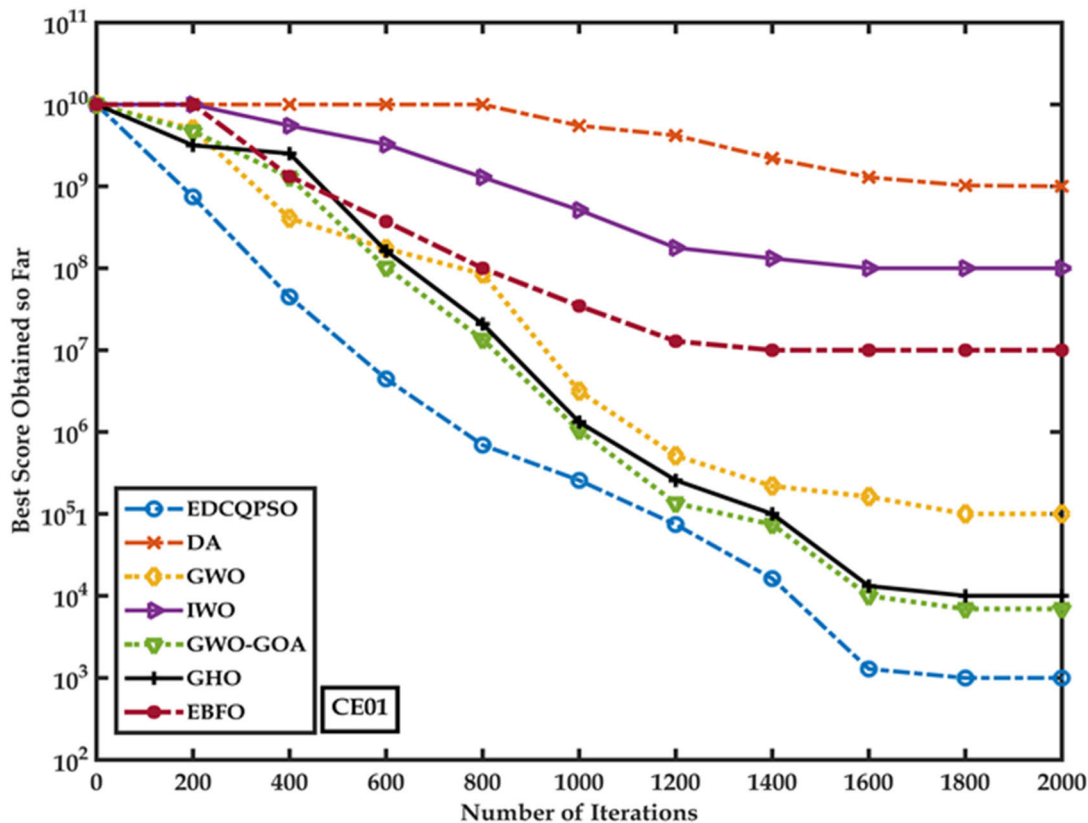
CE01
(a)
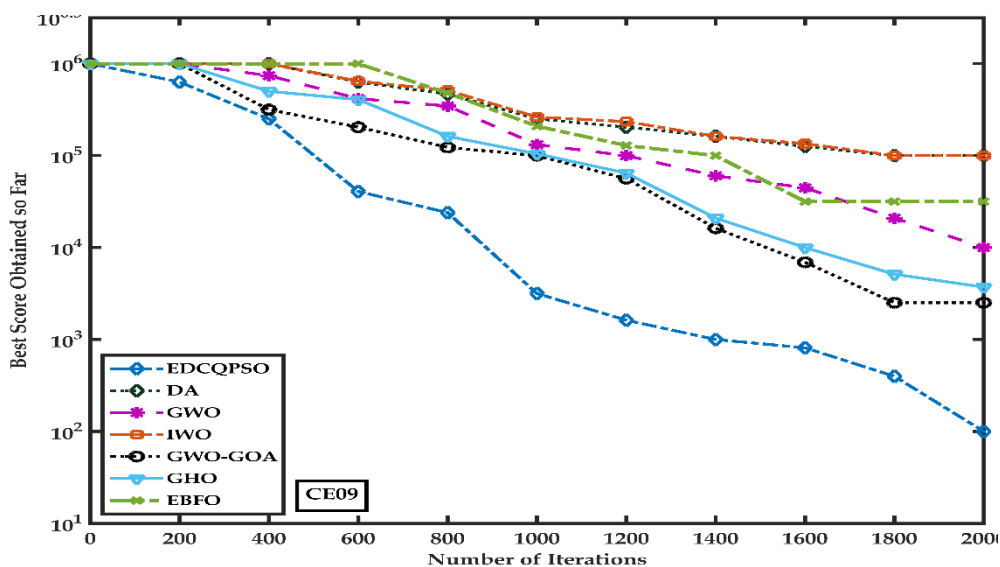
(b)

FIGURE 2. (a) Convergence progression for unimodal, multimodal, hybrid, and composite function for CEC2017. (b) CE09. (c) CE12. (d) CE26.
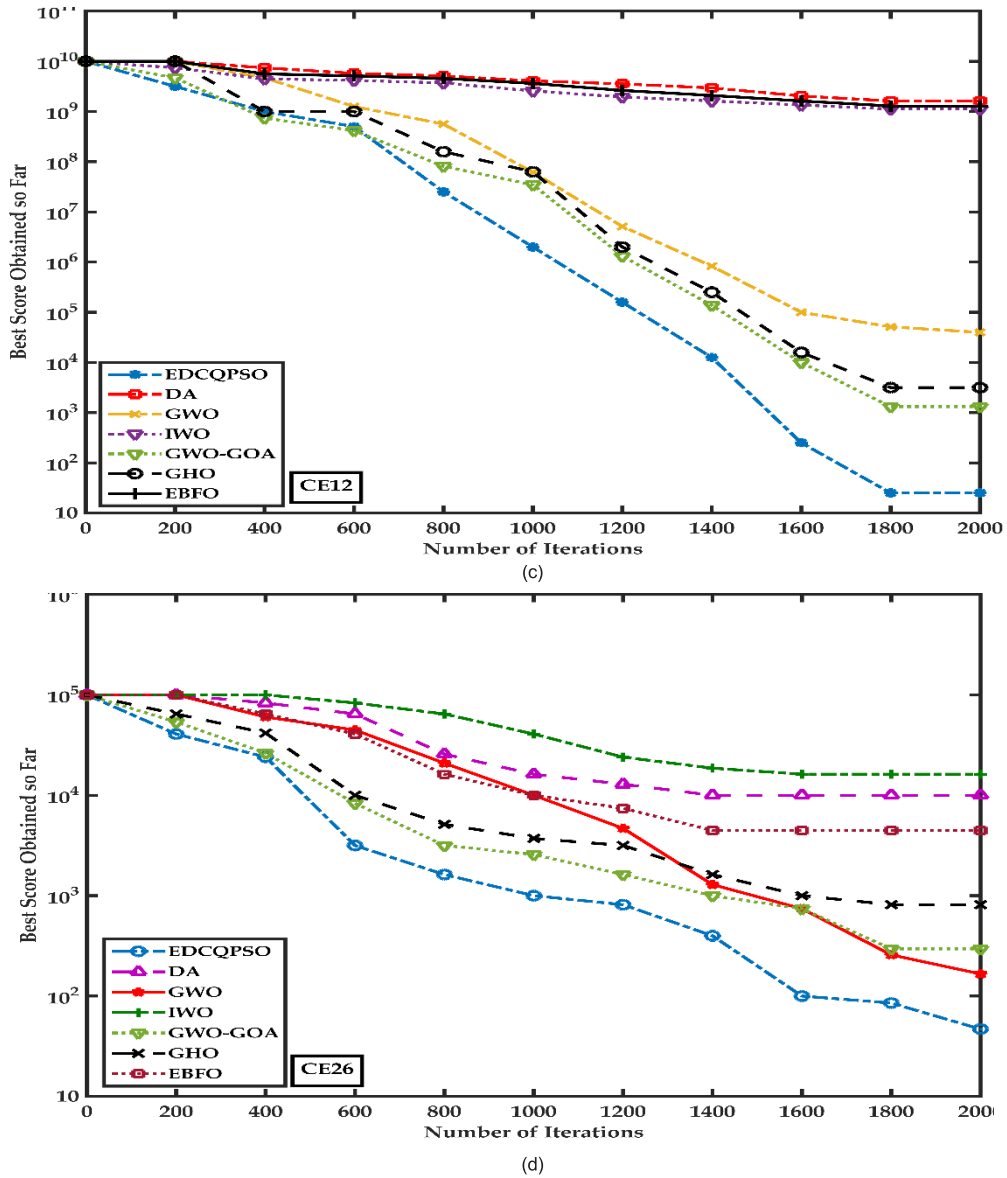
FIGURE 2. *(Continued.)* (a) Convergence progression for unimodal, multimodal, hybrid, and composite function for CEC2017. (b) CE09. (c) CE12. (d) CE26.

TABLE 5. Statistical analysis of different algorithms for IEEE CEC2017.

Algorithm	Rank of Algorithm	ASV	+ : = : -
EDCQPSO	1	1.375556	26 : 4 : 0
DA	7	11.14333	27 : 0 : 3
GWO	4	4.71	28 : 1 : 1
IWO	5	5.976667	30 : 0 : 0
GWO-GOA	2	4.092317	29 : 1 : 0
GHO	3	4.537778	24 : 3 : 3
EBFO	6	7.354444	30 : 0 : 0

TABLE 6. Simulation parameters (2).

Algorithm	Other Parameters
PSO [15]	$w = 1, c_1 = 2, c_2 = 2.$
CLQPSO [42]	$w \in [0.9, 0.2], m = 5, c = 1.496$
PDQPSO [40]	$w \in [0.9, 0.4], c_1 \in [2.5, 0.5], c_1 \in [0.5, 2.5]$
QPSO-CD [43]	$\alpha \in [1.0, 0.5], S = 2, c_1 = c_2 = 2.$
CSQPSO [44]	$\Delta \in [-1, 1], c_1 = 1.2, c_2 = 0.5.$
EDCQPSO	$B_l^t \in [0, 1]. \; \chi \in (0, 1)$

CLQPSO, PDQPSO, and PSO. Proposed algorithm shows an average improvement of 76.52%, 65.38%, 54.72%, 47.90%, and 43.75% in comparison with PSO, PDQPSO, CLQPSO, QPSO-CD and CSQPSO respectively.

The proposed approach has enhanced its global searching capability compared to the other optimal methods on all the test functions.
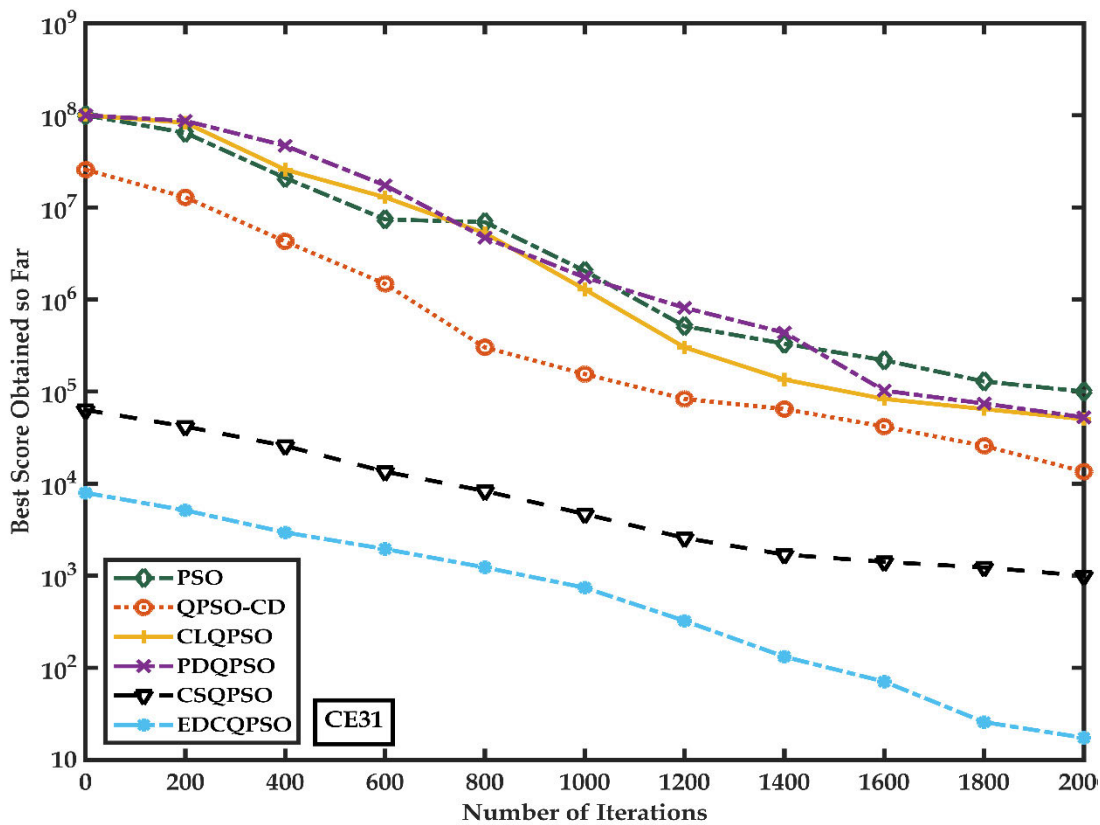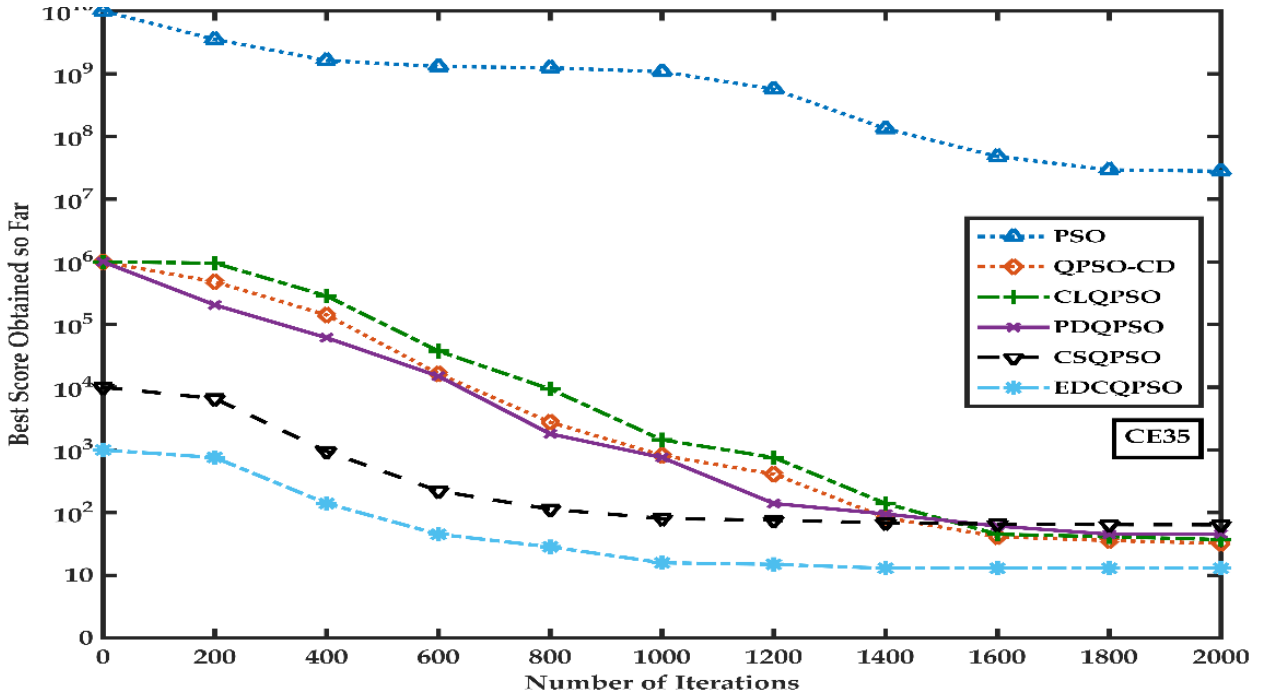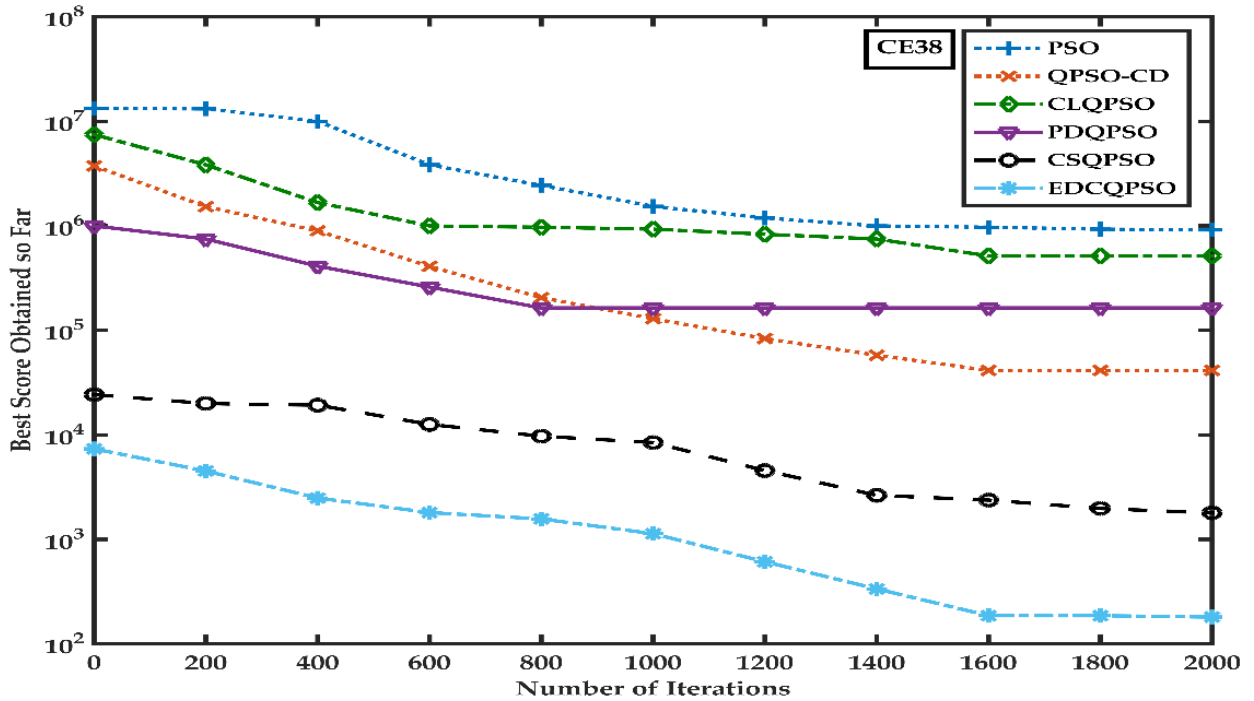
CE31

(a)
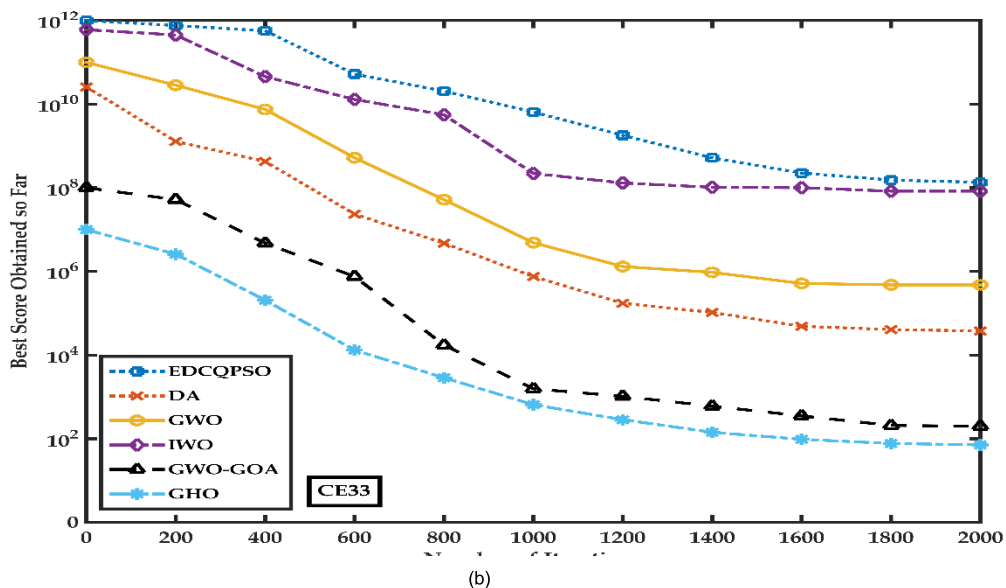
(b)

FIGURE 3. (a) Convergence progression for unimodal, multimodal, hybrid, and composite function for CEC2019. (b) CE33. (c) CE35. (d) CE38.

FIGURE 3. *(Continued.)* **(a) Convergence progression for unimodal, multimodal, hybrid, and composite function for CEC2019. (b) CE33. (c) CE35. (d) CE38.**

VI. EDCQPSO FOR IoT APPLICATIONS

IoT has large number of applications in different areas such as localization, target tracking, automation, environmental monitoring, utility meters, agriculture, health and many more. These applications in wide area are feasible because of large numbers of sensor nodes are deployed and periodically sensing of given parameters. Accurate localization of sensor nodes is one of the most crucial requirements for many applications. Localization is the process of estimating current locations of sensor nodes without the knowledge of

TABLE 7. Mean and standard deviation and rank of different algorithms for IEEE CEC2019.

		PSO	QPSO-CD	CLQPSO	PDQPSO	CSQPSO	EDCQPSO
CE31	Average	8896219	$3.61E+07$	$2.52E+08$	$1.49E+08$	459.1043	342.0721
	Std.Dev.	9085310	$1.41E+07$	$1.38E+08$	$1.09E+08$	1037.417	9719.327
CE32	Average	3418.755	21312.991	25725.15	9846.839	359.3299	321.2178
	Std.Dev.	1247.997	5618.317	6418.767	3100.053	482.5013	459.5207
CE33	Average	9.810943	5.181039	6.687039	8.342442	3.694955	2.814955
	Std.Dev.	1.817491	1.571842	1.664284	1.53575	2.341597	1.982033
CE34	Average	57.29179	39.39786	46.66576	25.81332	14.61567	10.11620
	Std.Dev.	22.17147	8.832141	9.121141	8.125042	7.36528	4.360001
CE35	Average	9.03234	1.410394	1.570454	1.661167	1.806667	1.200007
	Std.Dev.	8.57788	0.196681	0.213347	0.116622	0.069533	0.039211
CE36	Average	9.817933	6.231456	8.175133	3.799174	3.533781	2.421985
	Std.Dev.	1.126773	1.253678	1.284578	1.236747	1.823304	1.910023
CE37	Average	1396.01	10969.84	11159.353	681.257	680.2251	667.2373
	Std.Dev.	235.5401	327.8141	387.7451	243.6093	261.1453	252.1712
CE38	Average	4.967392	4.31508	4.512828	4.107098	3.926769	3.124242
	Std.Dev.	0.395858	0.243752	0.282175	0.436445	0.499053	0.421053
CE39	Average	1.621969	1.314002	1.528999	1.264191	1.184663	1.114663
	Std.Dev.	0.401281	0.232501	0.240001	0.102787	0.103211	0.086121
CE40	Average	21.92294	21.41423	21.47964	21.14255	23.34871	20.30004
	Std.Dev.	0.097032	0.086102	0.087129	0.084619	0.092195	0.078175

TABLE 8. Statistical analysis of different algorithms for IEEE CEC2019.

Algorithm	Rank of Algorithm	ASV	+ : = : -
EDCQPSO	1	1.5	6: 4: 0
PSO	6	6.3887	4: 0: 6
QPSO - CD	3	2.8791	8: 1: 1
CLQPSO	4	3.3125	9: 0: 1
PDQPSO	5	4.3333	10: 0: 0
CSQPSO	2	2.6667	9: 1: 0

their initial locations. Localization algorithm should have a capability to accurately locate the sensor node quickly with minimal energy consumption. To achieve the performance improvement, recently, swarm intelligence based algorithms are being developed for localizing the sensor nodes. Such challenge can be treated as optimization problem in a multi-dimensional space.

Here, using the EDCQPSO algorithm, we aim to localize the deployed IoT nodes and reduce the computational complexity, enhancing these resource-constrained node's lifetimes. To demonstrate localization, we consider IoT based smart car/vehicle parking application. We consider M number of anchor nodes and N number of normal sensor nodes ($M < N$) deployment in a two dimensional space. The model has an objective function $f(p_1, p_2, p_3, \ldots, p_m)$ which defines

coordinates of sensor nodes based on the information about anchor nodes location, using (16) and (22).

The constraints make the evaluated coordinates closer to real positions and helps in generating an accurate topology. In this case, objective function follows two steps. In first step, the normal sensor node will determine its own position based on the received signal strength indicator (RSSI) and time of arrival (ToA) of incoming signal from the anchor node. In the second step, it computes the location of the normal sensor node. For performance analysis, the results of EDCQPSO are compared with PDQPSO [40], CLQPSO [42], QPSO-CD [43], and CSQPSO [44]. With random deployment of sensor nodes in localization area, average localization error (ALE) is calculated as a standard statistical metric and given by

$$ALE = \frac{\sum_{i=1}^{N} \sqrt{\left(u_{i\_pred} - u_{i\_actual}\right)^2 + \left(v_{i\_pred} - v_{i\_actual}\right)^2}}{N}$$

$$(24)$$

where $\left(u_{i\_actual}, v_{i\_actual}\right)$ is the real-time position of the node, and $\left(u_{i\_pred}, v_{i\_pred}\right)$ is the node's estimated position.

The simulations were carried out for 200 m $\times$ 200 m with 200 nodes with random distribution so that M anchor nodes can be found. By assuming the Gaussian distributed RSSI ranging error and node transmission range of 10m to 40m and anchor nodes changing from 20 to 60. Other parameters are same as given in Table 5. The results of anchor node versus ALE for all four algorithms is as shown in Fig. 4. The proposed approach reduces the ALE by a minimum of 47.5%,
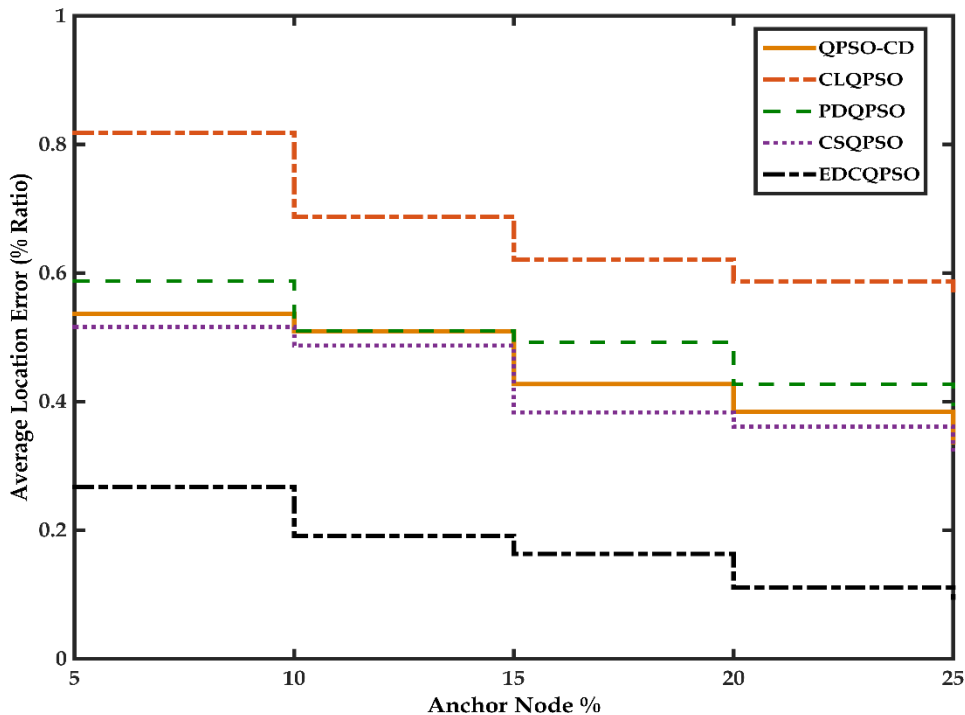
FIGURE 4. Average location error.

31.5%, 26.37% and 25%, compared to CLQPSO, PDQPSO, QPSO-CD and CSQPSO, respectively. It is also observed that the position approximation precision for all the approaches is high for the higher node densities.

VII. CONCLUSION

A novel hybrid enhanced differential crossover quantum PSO algorithm is proposed for IoT applications where real-time processing is required in the presence of multiple realistic constraints. Our algorithm uses quantum PSO, differential evolution operator, and crossover operator. Performance and the proposed algorithm results are validated with thirty benchmark functions of IEEE CEC2017 and on ten test functions of IEEE CEC2019. The algorithm performance is also compared with other existing optimization algorithms and the PSO variants. Results of the proposed algorithm have smaller fitness values, high precision, and faster convergence. The algorithm is used to localize the IoT nodes in smart parking application, and the average location error is reduced up to 25% compared to the existing algorithms.

REFERENCES

[1] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 238–255, Jan. 2021.

[2] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, and D. Liu, "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowl.-Based Syst.*, vol. 96, pp. 61–75, Mar. 2016.

[3] A. G. Hussien, M. Amin, M. Wang, G. Liang, A. Alsanad, A. Gumaei, and H. Chen, "Crow search algorithm: Theory, recent advances, and applications," *IEEE Access*, vol. 8, pp. 173548–173565, 2020.

[4] M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai, H. Huang, and C. Tong, "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses," *Neurocomputing*, vol. 267, pp. 69–84, Dec. 2017.

[5] H. Zhang *et al.*, "A multi-strategy enhanced salp swarm algorithm for global optimization," *Eng. Comput.*, 2020, doi: 10.1007/s00366-020-01099-4.

[6] X. Liang *et al.*, "Chaotic oppositional sine–cosine method for solving global optimization problems," *Eng. Comput.*, 2020, doi: 10.1007/s00366-020-01083-y.

[7] A. F. Ba *et al.*, "Levy-based antlion-inspired optimizers with orthogonal learning scheme," *Eng. Comput.*, 2020, doi: 10.1007/s00366-020-01042-7.

[8] Y. Zhang *et al.*, "Boosted binary Harris hawks optimizer and feature selection," *Eng. Comput.*, 2020, doi: 10.1007/s00366-020-01028-5.

[9] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, p. 78, 2010, doi: 10.1504/IJBIC.2010.032124.

[10] X. Xu and H.-L. Chen, "Adaptive computational chemotaxis based on field in bacterial foraging optimization," *Soft Comput.*, vol. 18, no. 4, pp. 797–807, Apr. 2014.

[11] Z. Cai, J. Gu, J. Luo, Q. Zhang, H. Chen, Z. Pan, Y. Li, and C. Li, "Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy," *Expert Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112814.

[12] A. A. Heidari, R. A. Abbaspour, and H. Chen, "Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105521.

[13] X. Zhao, X. Zhang, Z. Cai, X. Tian, X. Wang, Y. Huang, H. Chen, and L. Hu, "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Comput. Biol. Chem.*, vol. 78, pp. 481–490, Feb. 2019.

[14] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov. 1995, pp. 1942–1948.

[16] Z. Zhang, N. Zhang, and Z. Feng, "Multi-satellite control resource scheduling based on ant colony optimization," *Expert Syst. Appl.*, vol. 41, no. 6, pp. 2816–2823, May 2014.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[18] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016, doi: 10.1007/s00521-015-1920-1.

[19] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in Arabic sentiment analysis," *Int. J. Speech Technol.*, vol. 49, no. 5, pp. 1688–1707, May 2019.

[20] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (Studies in Computational Intelligence). Berlin, Germany: Springer, 2010, pp. 65–74.

[21] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[22] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105884.

[23] R. Purushothaman, S. P. Rajagopalan, and G. Dhandapani, "Hybridizing gray wolf optimization (GWO) with grasshopper optimization algorithm (GOA) for text feature selection and clustering," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106651.

[24] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[25] N. T. Tam, D. T. Hai, L. H. Son, and L. T. Vinh, "Improving lifetime and network connections of 3D wireless sensor networks based on fuzzy clustering and particle swarm optimization," *Wireless Netw.*, vol. 24, no. 5, pp. 1477–1490, Jul. 2018.

[26] W. Guo, B. Zhang, G. Chen, X. Wang, and N. Xiong, "A PSO-optimized minimum spanning tree-based topology control scheme for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 4, Apr. 2013, Art. no. 985410.

[27] S. Goyal and M. S. Patterh, "Modified bat algorithm for localization of wireless sensor network," *Wireless Pers. Commun.*, vol. 86, no. 2, pp. 657–670, Jan. 2016.

[28] B. You, G. Chen, and W. Guo, "Topology control in wireless sensor networks based on discrete particle swarm optimization," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, Nov. 2009, pp. 269–273.

[29] S. N. Ghorpade, M. Zennaro, and B. S. Chaudhari, "Binary grey wolf optimisation-based topology control for WSNs," *IET Wireless Sensor Syst.*, vol. 9, no. 6, pp. 333–339, Dec. 2019.

[30] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10744–10756, 2017.

[31] M. Zemzami, N. Hami, M. Itmi, and N. Hmina, "A comparative study of three new parallel models based on the PSO algorithm," *Int. J. Simul. Multidisciplinary Des. Optim.*, vol. 11, p. 5, Jan. 2020.

[32] P. S. Mann and S. Singh, "Energy-efficient hierarchical routing for wireless sensor networks: A swarm intelligence approach," *Wireless Pers. Commun.*, vol. 92, no. 2, pp. 785–805, Jan. 2017.

[33] S. N. Ghorpade, M. Zennaro, and B. S. Chaudhari, "GWO model for optimal localization of IoT-enabled sensor nodes in smart parking systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1217–1224, Feb. 2021.

[34] V. Kumar, A. Kumar, and S. Soni, "A combined Mamdani–Sugeno fuzzy approach for localization in wireless sensor networks," in *Proc. Int. Conf. Workshop Emerg. Trends Technol.*, 2011, pp. 798–803.

[35] S. Ghorpade, M. Zennaro, and B. S. Chaudhari, "Towards green computing: Intelligent bio-inspired agent for IoT-enabled wireless sensor networks," *Indersci. Int. J. Sensor Netw.*, vol. 35, no. 2, pp. 121–131, 2021.

[36] S. Phoemphon, C. So-In, and D. T. Niyato, "A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm optimization for wireless sensor network localization," *Appl. Soft Comput.*, vol. 65, pp. 101–120, Apr. 2018, doi: 10.1016/j.asoc.2018.01.004.

[37] F. Van den Bergh, "An analysis of particle swarm optimizers," M.S. thesis, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, Nov. 2001.

[38] J. Sun, C.-H. Lai, W. B. Xu, Y. Ding, and Z. Chai, "A modified quantum-behaved particle swarm optimization," in *Proc. 7th Int. Conf. Comput. Sci. (ICCS)*, 2007, pp. 294–301.

[39] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[40] H. Long, S. Wu, and H. Fu, "Parallel diversity-controlled quantum-behaved particle swarm optimization algorithm," in *Proc. 10th Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2014, pp. 74–79.

[41] L. D. S. Coelho, "A quantum particle swarm optimizer with chaotic mutation operator," *Chaos, Solitons Fractals*, vol. 37, no. 5, pp. 1409–1418, Sep. 2008.

[42] S. Tu, O. U. Rehman, S. Rehman, S. U. Khan, M. Waqas, and R. Zhu, "A novel quantum inspired particle swarm optimization algorithm for electromagnetic applications," *IEEE Access*, vol. 8, pp. 21909–21916, 2020.

[43] A. S. Bhatia, M. K. Saggi, and S. Zheng, "QPSO-CD: Quantum-behaved particle swarm optimization algorithm with Cauchy distribution," *Quantum Inf. Process.*, vol. 19, no. 10, p. 345, Oct. 2020.

[44] Y. Zhang and Z. Jin, "Quantum-behaved particle swarm optimization with generalized space transformation search," *Soft Comput.*, vol. 24, no. 19, pp. 14981–14997, Oct. 2020.

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

[46] C. Chen, X. Wang, H. Yu, N. Zhao, M. Wang, and H. Chen, "An enhanced comprehensive learning particle swarm optimizer with the elite-based dominance scheme," *Complexity*, vol. 2020, pp. 1–24, Oct. 2020, doi: 10.1155/2020/4968063.

[47] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[48] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.

[49] A. LaTorre and J.-M. Pena, "A comparison of three large-scale global optimizers on the CEC 2017 single objective real parameter numerical optimization benchmark," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2017, pp. 1063–1070.

[50] A. Epstein, M. Ergezer, I. Marshall, and W. Shue, "GADE with fitness-based opposition and tidal mutation for solving IEEE CEC2019 100-digit challenge," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 395–402.

SHEETAL N. GHORPADE received the M.Sc. degree in mathematics from the University of Pune, Pune, India, and the Ph.D. degree in mathematics from Gyanvihar University, Jaipur, India. With more than 20 years of teaching experience, she is currently an Associate Professor in applied mathematics with the RMD Sinhgad School of Engineering, Pune. She has undertaken research visits at the International Centre for Theoretical Physics (ICTP), Italy, as a Simon's Visitor, in 2018 and 2019. Her research interests include graph theory, optimization techniques, wireless sensor networks, and the Internet of Things.

MARCO ZENNARO (Senior Member, IEEE) received the M.Sc. degree in electronic engineering from the University of Trieste, Trieste, Italy, and the Ph.D. degree from the KTH-Royal Institute of Technology, Stockholm, Sweden. He is a Research Scientist with The Abdus Salam International Centre for Theoretical Physics, Trieste, where he is the Unit Coordinator with the Department of Science, Technology, and Innovation. He is a Visiting Professor with the KIC-Kobe Institute of Computing, Japan. He has given lectures on wireless technologies in more than 30 countries. His research interests include ICT4D and the use of ICT for development, and in particular he investigates the use of the IoT in developing countries.

BHARAT S. CHAUDHARI (Senior Member, IEEE) received the M.E. degree in telecom engineering and the Ph.D. degree from Jadavpur University, Kolkata, India, in 1993 and 2000, respectively. After being a Full Professor in electronics and telecommunication engineering with the Pune Institute of Computer Technology and the Dean of the International Institute of Information Technology, Pune. He joined MIT World Peace University, (then MIT Pune), Pune, India, as a Professor, in 2014. He has authored more than 70 research articles in the field of wireless, telecom, and optical networks. His research interests include wireless sensors networks, low-power wide area networks, the Internet of Things, and optical networks. He has been a Simon's Associate of the International Centre for Theoretical Physics (ICTP), Italy, since 2015. He received the Young Scientist Research Grant from the Department of Science and Technology, Government of India. He is the Founder Chair of IEEE Pune Section and a fellow of IETE and IE (I).

HESHAM ALHUMYANI received the Ph.D. degree from the University of Connecticut, Storrs, USA. He is currently working with the Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia, where he was appointed as the Faculty Dean, in 2019. He has published many research articles in distinctive journals and conferences. His research interests include wireless sensor networks, underwater sensing, the Internet of Things (IoT), and cloud computing.

RASHID A. SAEED (Senior Member, IEEE) received the Ph.D. degree in communications and network engineering, Universiti Putra Malaysia (UPM). He is currently a Professor with the Computer Engineering Department, Taif University. He is also working with the Electronics Department, Sudan University of Science and Technology (SUST). He was a Senior Researcher in Telekom Malaysia Research and Development (TMRND) and MIMOS. He has published more than 150 research articles, books, and book chapters on wireless communications and networking in peer-reviewed academic journals and conferences. His areas of research interests include computer networks, cognitive computing, computer engineering, wireless broadband, and WiMAX femtocell. He is successfully awarded with three U.S. patents in these areas. He supervised more than 50 M.Sc./Ph.D. students. He is a member of IEM (I.E.M), SigmaXi, and SEC.

S. ABDEL-KHALEK received the Ph.D. degree in computer science from Azhar University, in 2016. He is a Full Professor in applied mathematics with the Mathematics Department, Faculty of Science, Sohag University, Egypt. He is also an Associate Professor in applied mathematics with the Department of Mathematics, Faculty of Science, Taif University, Taif, Saudi Arabia. He is the author of several articles published in different international scientific journals. His research interests include different directions in quantum information and computer sciences. He is a member of different working groups.

• • •