Received December 25, 2019, accepted January 5, 2020, date of publication January 8, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964783

Quasi-Affine Transformation Evolutionary Algorithm With Communication Schemes for Application of RSSI in Wireless Sensor Networks

ZHI-GANG DU, JENG-SHYANG PAN, (Senior Member, IEEE), SHU-CHUAN CHU, HAN-JIANG LUO, AND PEI HU

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Shu-Chuan Chu (scchu0803@gmail.com)

ABSTRACT QUasi-Affine TRansformation Evolutionary algorithm (QUATRE) is a new optimization algorithm based on population for complex multiple real parameter optimization problems in real world. In this paper, a novel multi-group multi-choice communication strategy algorithm for QUasi-Affine TRansformation Evolutionary (MM-QUATRE) algorithm is proposed to solve the disadvantage that the original QUATRE is always easily to fall into local optimization in the strategy of updating bad nodes with multiple groups and multiple choices. We compared it with other intelligent algorithms, the most advanced PSO variant, parallel PSO (P-PSO) variant, native QUATRE and parallel QUATRE (P-PSO) under CEC2013 large-scale optimization test suite. Thus, the performance of MM-QUATRE was verified. The conclusion that the MM-QUATRE algorithm is superior to other intelligent algorithms is proved by the experimental results. In addition, the application results of MM-QUATRE algorithm (MM-QUATRE-RSSI) based on RSSI in WSN node localization were analyzed and studied. The results appear that this method has higher localization accuracy than other similar algorithms.

INDEX TERMS PSO, P-PSO, QUATRE, RSSI, WSN, MM-QUATRE, bad point update.

I. INTRODUCTION

In the past decades, the problem of global optimization has attracted the attention of many scholars. These scholars have resarched various bionic intelligent optimization algorithms, and evolutionary computing (EC) techniques [1]–[3]. In general, by using the gradient of the objective function, the maximum or minimum value of the objective function is obtained, but the multiple dimensional function is extremely time-consuming and always trapped into the poor local optimization. The development of evolutionary computing technology makes it possible to solve these complex problems. Moreover, Some functions are nondifferentiable, so evolutionary computing technology comes into being. Optimization problem not only exists in the field of science, but are also found in our daily lives [4]. Different objective function must be used for different optimization problems. Therefore, in the process of optimization designing the appropriate objective function is the primary step. Although in some specific optimization, the objective function usually contains a series of constraints, the unconditional constrained optimization method plays a fundamental role in various optimization applications, which will affect the performance and direction of the constrained optimization method. In this article, the main focus is on unconstrained optimization in the search field. In addition, there are many ways to deal with unconstrained optimization. Computational intelligence (CI) [5]–[11] provides a new thinking for these optimization problems, especially if the problem is related to uncertain or noisy. Evolutionary computing (EC) [12], [13] consists of many simple and efficient algorithms, and proposes an evolutionary optimization method that can be reduced to a subset of CI. It has many other branches, such as bionic intelligent algorithm, neural network, quantum computing, meme computing and so on. For example, by simulating the foraging behavior of birds in a certain area, a simple and effective particle swarm

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

optimization(PSO) algorithm is generated [14]–[16]. In addition, many researchers have studied and improved the optimization ability of native PSO by improving PSO algorithms search strategies. Differential evolution (DE) is a new algorithm inspired by genetic annealing algorithm, it is formed by a mixture of genetic algorithm and simulated annealing algorithm. By simulating the release of pheromones during foraging, the ant optimization algorithm is generated. The artificial bee colony (ABC) algorithm was developed by simulating the communication behavior between bee [17]–[19]. Differential Evolution Algorithm (DE) [20] is an efficient global optimization algorithm. It is also a group-based heuristic search algorithm. Each individual in the group corresponds to a solution vector. The evolutionary process of the differential evolution algorithm is very similar to the genetic algorithm, including mutation, hybridization and selection operations, but the specific definition of these operations is different from the genetic algorithm.

QUATRE [21]–[24] is a novel proposed global optimization algorithm for evolutionary structure. It overcomes the disadvantages of DE algorithm, such as the decrease of population diversity, premature convergence to the local optimal point, or stagnation of the algorithm with the increase of the number of evolutionary evolution iterations. QUATRE also has the abiliy to converge very fast, even in the optimization process of solving high-dimensional Multi-modal problems. For purpose of enhance the global optimization performance of QUATRE and avoid falling into the local optimal position, we propose a new improved strategy of Multi-group and Multi-choice in this paper. It is called MM-QUATRE. The algorithm optimizes the performance of the native QUATRE algorithm by selecting a random communication strategy between subgroups to update the lower state points in each group. Therefore, compared with other algorithms, MM-QUATRE algorithm has obvious advantages in global optimization.

Node location information is very important and plays a key role in WSN [25], [26], [28], [29], navigation, tracking, monitoring and other applications. According to whether the distance between nodes needs to be measured or not, the positions can be divided into those based on distance measurement and those without distance measurement [30]–[32]. According to the occasion of deployment can be divided into outdoor positioning and indoor positioning. In general, the positioning accuracy of range-based positioning algorithm is higher than that of rangeless positioning algorithm. The common methods based on distance measurement include angle orientation, three-face orientation and maximum likelihood estimation [27]. Common ranging methods include RSSI, TOA/TDOA/RTOF, phase difference, near-field electromagnetic ranging (NFER), etc. Because RSSI-based ranging method does not need additional equipment, simple and easy to operate, in recent years, published research results have been widely used. In recent years, many scholars have applied intelligent algorithm to the positioning algorithm of wireless sensor network based on distance measurement [22], [30], [33], [34].

The remain of this paper deals with the following. In Section 2, we briefly introduces QUATRE algorithm and RSSI based wireless sensor network node location. Section 3 the proposed MM-QUATRE algorithm is introduced in detail. Section 4 shows the experimental results of the single objective real parameter optimization under the CEC2013 [35]–[37] test suite and compares them with other EC algorithms and the results of application in RSSI-based wireless sensor network. Finally, the newly proposed MM-QUATRE algorithm is summarized in section 5.

II. RELATED WORKS

A. QUATRE ALGORITHM

In geometry, affine transformation is the transformation process from one affine space to another affine space [21], [24]. The accurate evolutionary formula of QUATRE is shown in eq.1.

$$\mathbf{X} \leftarrow \mathbf{M} \otimes \mathbf{X} + \overline{\mathbf{M}} \otimes \mathbf{B} \tag{1}$$

The operator $\otimes$ represents the multiplication by components, and its meaning is the same as the '.* 'operator in Matlab. The matrix $\mathbf{X}$ represents the position matrix of the particle swarm, then $\mathbf{X} = \left[\vec{x_1}, \vec{x_2}, \ldots, \vec{x_{ps}}\right]^T$. The vector $\vec{x_i}$ represents the position of the ith particle in the population, which can be expressed as $\vec{x_i} = [x_{i1}, x_{i2}, \ldots, x_{iD}]$. ps represents the number of particles in the population. Matrix $\mathbf{B}$ represents the mutation matrix of particles between populations, which can be generated in various ways. c represents the difference matrix coefficient factor. Where, the difference matrix is the result of $\mathbf{X_{r1}} - \mathbf{X_{r2}}$. TABLE 1 lists seven mutation schemes for matrix $\mathbf{B}$ [22]. $\mathbf{X_{r1}}$, $\mathbf{X_{r2}}$, $\mathbf{X_{r3}}$, $\mathbf{X_{r4}}$, $\mathbf{X_{r5}}$ represent the random matrix generated by random permutation of the matrix $\mathbf{X}$. $\vec{x_{gbest,G}}$ denote the position vector of the globally optimal particle at the Gth iteration. $\mathbf{X_{gbest,G}} = \left[\vec{x_{gbest,G}}, \vec{x_{gbest,G}}, \ldots, \vec{x_{gbest,G}}\right]$ global optimal particle position vector form the first iteration to the current iteration [21], [22], [24].

$$\mathbf{X} = \begin{bmatrix} \vec{x_1} \\ \vec{x_2} \\ \cdots \\ \vec{x_{ps}} \end{bmatrix} \qquad \mathbf{X_{gbest,G}} = \begin{bmatrix} \vec{x_{gbest,G}} \\ \vec{x_{gbest,G}} \\ \cdots \\ \vec{x_{gbest,G}} \end{bmatrix} \tag{2}$$

$\mathbf{M}$ is the evolutionary matrix, made up of 0 and 1. $\overline{\mathbf{M}}$ represents a matrix of binary inverse operations about M. Binary inversion means taking the inverse of a matrix. The 0 element inverts the contravariant to 1, 1 element inverts the contravariant to 0 [21], [22], [24]. As shown in eq.3.

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \qquad \overline{\mathbf{M}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

The evolution matrix $\mathbf{M}$ is diverse in QUATRE algorithm. The generation of the evolutionary matrix $\mathbf{M}$ requires first

TABLE 1. Seven schemes of mutation matrix **B** calculation.

No.	QUATRE/x/y	Equation
1	QUATRE/rand/1	$\mathbf{B} = \mathbf{X_{r1,G}} + F \cdot (\mathbf{X_{r2,G}} - \mathbf{X_{r3,G}})$
2	QUATRE/best/1	$\mathbf{B} = \mathbf{X_{gbest,G}} + F \cdot (\mathbf{X_{r1,G}} - \mathbf{X_{r2,G}})$
3	QUATRE/target/1	$\mathbf{B} = \mathbf{X} + F \cdot (\mathbf{X_{r1,G}} - \mathbf{X_{r2,G}})$
4	QUATRE/target-to-best/1	$\mathbf{B} = \mathbf{X} + F \cdot (\mathbf{X_{gbest,G}} - \mathbf{X}) + F \cdot (\mathbf{X_{r1,G}} - \mathbf{X_{r2,G}})$
5	QUATRE/rand/2	$\mathbf{B} = \mathbf{X_{r1,G}} + F \cdot (\mathbf{X_{r2,G}} - \mathbf{X_{r3,G}}) + F \cdot (\mathbf{X_{r4,G}} - \mathbf{X_{r5,G}})$
6	QUATRE/best/2	$\mathbf{B} = \mathbf{X_{gbest,G}} + F \cdot (\mathbf{X_{r1,G}} - \mathbf{X_{r2}}) + F \cdot (\mathbf{X_{r3,G}} - \mathbf{X_{r4,G}})$
7	QUATRE/target/2	$\mathbf{B} = \mathbf{X} + F \cdot (\mathbf{X_{r1,G}} - \mathbf{X_{r2,G}}) + F \cdot (\mathbf{X_{r3,G}} - \mathbf{X_{r4,G}})$

initializing a $\mathbf{M_{tmp}}$ matrix of the lower triangle, and then randomly permutation the row vectors of $\mathbf{M_{tmp}}$ [21], [22], [24]. When $ps = D$, the transformation formula of $\mathbf{M}$ is shown as eq.4.

$$\mathbf{M_{tmp}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

When the total number of particle population ps is much larger than the individual dimension D, the evolution matrix $\mathbf{M}$ needs to expand according to ps. In general, when $ps = i \times D + K$, the first $i \times D$ rows of matrix $\mathbf{M}$ are composed of i lower triangular matrices of $D \times D$, and the last K rows are composed of the first K rows of $D \times D$ lower triangular matrices [21], [22], [24]. For example, when $ps = 2 \times D + 2$, is shown in eq.5.

$$\mathbf{M_{tmp}} = \begin{bmatrix} 1 & & & \dots & \\ 1 & 1 & & \dots & \\ 1 & 1 & 1 & \dots & \\ & & & \dots & \\ 1 & 1 & 1 & \dots & 1 \\ 1 & & & \dots & \\ 1 & 1 & & \dots & \\ 1 & 1 & 1 & \dots & \\ & & & \dots & \\ 1 & 1 & 1 & \dots & 1 \\ 1 & & & \dots & \\ 1 & 1 & & \dots & \end{bmatrix} \mathbf{M} = \begin{bmatrix} 1 & \dots & & 1 & \\ & 1 & & \dots & \\ 1 & 1 & \dots & 1 & 1 \\ & & & \dots & \\ 1 & & & \dots & 1 \\ & 1 & \dots & 1 & \\ 1 & 1 & & \dots & \\ 1 & & 1 & \dots & \\ & & & \dots & \\ & 1 & & \dots & \\ 1 & & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{5}$$

B. NETWORK NODE LOCATION OF RSSI

Suppose there are a total of $K = M + N$ nodes in a two-dimensional region, with M anchor nodes and N unknown nodes. The coordinates of anchor node are $B_i(u_i, v_i)$ $(i = 1, 2, ..., N)$, and the coordinates of the unknown node are $S_t(u_t, v_t)$ $(t = 1, 2..., M)$. RSSI value can be obtained when the node communicates, and the distance d_{ui} between the anchor node and the unknown node can be calculated by eq.6. $\hat{d}_i$ represents the distance from the point located using the DV_hop method to the i-th anchor node. In wireless sensor networks, the main purpose of the positioning problem is to minimize the positioning error. Considering that the distance estimation error also increases with the number of hops. Therefore, the error function can be weighted by the reciprocal of the hop square [22]. The fitness function of MM-QUATRE can be defined as eq.7.

$$d_{ui} = \sqrt{(u - u_i)^2 + (v - v_i)^2} \tag{6}$$

$$f(x, y) = min(\sum_{i=1}^{m} (\frac{1}{hop_{ui}})^2 (d_{ui} - \hat{d}_i)^2) \tag{7}$$

1) CALCULATION OF WEIGHTED HOPS BASED ON RSSI

The model of RSSI is shown in eq.8, which takes into account the complexity of the transmission environment and the influence of noise on the ranging results in the transmission process. In eq.8, $p_r(d_0)$ represents the power received when the distance between two nodes is d_0, while $p_r(d)$ represents the power received when the distance between two nodes is d, and η represents the path loss constant of the signal in the path transmission process (the actual measured value is between 1.5 and 5) [38], [39]. x_σ is an uncertainty factor determined by multipath fading and occlusion.

$$P_r(d) = p_r(d_0) - 10\eta \log(\frac{d}{d_0}) + X_\sigma \tag{8}$$

When the distance between two nodes is so far, the final positioning error is also so larger. Hence, the RSSI error should be given according to this rule. First, $RSSI_j$ needs to be added to the broadcast message, which contains the signal received after the jth broadcast from the ith anchor node B_i to the kth unknown node S_k. If the number of hops corresponding to the message received is larger than the number of hops of the previous message, the anchor node discards the message with a larger number of hops [39]. Then find the path proportionality coefficient λ from one anchor node B_i to all anchor nodes, as shown in eq.9.

$$\lambda_i = \sum_{j=1, j \neq i}^{m} \frac{\hat{D}_{i,j}}{d_{i,j}}, \tag{9}$$

where $\hat{D}_{i,j}$ represents the connection distance from the i-th anchor node to the j-th anchor node through the relay of multiple nodes, and $d_{i,j}$ represents the Euclidean distance from the i-th anchor node to the j-th anchor node.

Finally, the path scale factor λ of the i-th anchor node obtained by eq.9 and the relay distance $\hat{D}_n$ of the unknown node to the anchor node i obtained by eq.8 are used to estimate the true distance from the unknown node to the anchor node i according to eq.10.

$$d = \frac{1}{\lambda} \hat{D}_n, \tag{10}$$

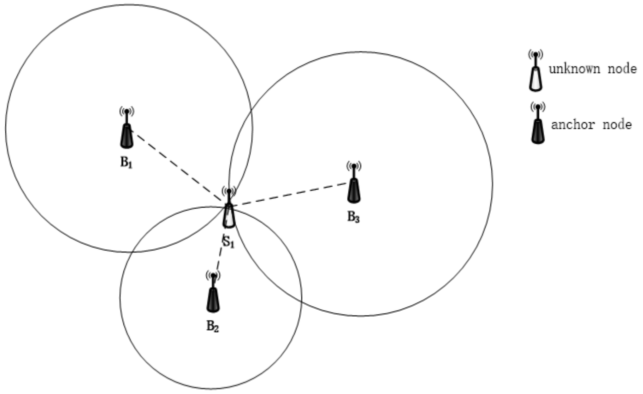


FIGURE 1. Schematic diagram of three side measurement method.

2) LOCATION NODE CALCULATION BASED ON RSSI

If the anchor node consists of more than two, the location is conducted by the method of trilateral measurement or maximum likelihood estimation [40], as shown in FIGURE 1.
where, the coordinates of anchor nodes B_1, B_2 and B_3 are (u_1, v_1), (u_2, v_2) and (u_3, v_3), respectively. S_1 represents the unknown node, and its coordinates are assumed as (u, v). When there are three or more anchor nodes, the coordinates (u, v) of unknown node S_1 can be obtained through the solution eq.11.

$$\begin{cases} (u_1-u)^2+(v_1-v)^2=d_1^2 \\ (u_2-u)^2+(v_2-v)^2=d_2^2 \\ \cdots \\ (u_m-u)^2+(v_m-v)^2=d_m^2 \end{cases} \tag{11}$$

Formula (11) can be firstly expanded to formula (12), and then each equation can be obtained by subtracting the last equation, as shown below.

$$\begin{cases} u_1^2-u_m^2+2(u_1-u_m)u+v_1^2-v_m^2 \\ \qquad -2(v_1-v_m)v=d_1^2-d_m^2 \\ u_2^2-u_m^2+2(u_2-u_m)u+v_2^2-v_m^2 \\ \qquad -2(v_2-v_m)v=d_2^2-d_m^2 \\ \qquad \cdots \\ u_{m-1}^2-u_m^2+2(u_{m-1}-u_m)u+v_{m-1}^2 \\ -v_m^2-2(v_{m-1}-v_m)v=d_{m-1}^2-d_m^2 \end{cases} \tag{12}$$

Then the eq.12 is written as a matrix form of $\mathbf{AX}=\mathbf{B}$. Finally, the unknown anchor node position matrix $\mathbf{X}$ can be expressed as eq.16.

$$\mathbf{X}=\begin{bmatrix} u \\ v \end{bmatrix} \tag{13}$$

$$\mathbf{A}=\begin{bmatrix} 2(u_1-u_m) & 2(v_1-v_m) \\ 2(u_2-u_m) & 2(v_2-v_m) \\ \cdots & \\ 2(u_{m-1}-u_m) & 2(v_{m-1}-v_m) \end{bmatrix} \tag{14}$$

$$\mathbf{B}=\begin{bmatrix} u_1^2+v_1^2-u_m^2-v_m^2+d_m^2-d_1^2 \\ u_2^2+v_2^2-u_m^2-v_m^2+d_m^2-d_2^2 \\ \cdots \\ u_{m-1}^2+v_{m-1}^2-u_m^2-v_m^2+d_m^2-d_{m-1}^2 \end{bmatrix} \tag{15}$$

$$\mathbf{AX}=\mathbf{B} \quad \Rightarrow \quad \mathbf{X}=(\mathbf{A^TA})^{-1}\mathbf{A^TB} \tag{16}$$

III. MM-QUATRE ALGORITHM AND ITS APPLICATION IN WSN LOCALIZATION BASED ON RSSI

A. MM-QUATRE ALGORITHM

The original QUATRE algorithm has certain limitations, that is to say, it is very easy to fall into the local optimal, so it is necessary to avoid all individuals of the population falling into the local optimal solution through the idea of group communication. MM-QUATRE algorithm divides to multiple subgroups and enables them to communicate after a certain number of iterations. By exchanging information among groups and selecting different strategies, the positions of some particles with bad states in each subgroup can be changed, so that they have a stronger optimization ability. The specific idea is to carry out inter-group communication after iterating *iter* times. Each group selects *s* worst points and uses the global optimization of other groups to evolve these bad points. For purpose of further realizing the diversity in the process of evolution and improve the ability of global optimization, one of the strategies was selected randomly for updating evolution. The evolutionary strategy is shown in TABLE 2 (assuming there are four groups).

In TABLE 2, $Xbad_{i,j}^G$ represents the position of the jth bad point in the ith subgroup after iterating G times. $Xbad_m^G$ represents the globally optimal particle position of the mth subgroup after iterating G times. r_1,r_2 and r_3 are uniformly distributed values ranging from 0 to 0.3, 0.3 to 0.7, and 0.7 to 1, respectively.

The concrete implementation of the algorithm consists of the following steps.

Step 1, initialize the parameters required by the algorithm, such as the number of groups *ps*, the number of subgroups *t* (this paper takes four groups as examples), the maximum iteration times *iterMax*, and the step size *GenCS* among groups.

Step 2, initialize the positions of all particles in the four subgroups and form the position matrix $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, and $\mathbf{X_4}$ of particles in all subgroups according to eq.2.

Step 3, generate evolution matrix M and mutation matrix B according to eq.5 and TABLE 1.

Step 4, according to eq.1, each subgroup evolves independently, and intergroup communication is carried out for many of *GenCS* iterations and a communication strategy is randomly selected from TABLE 2 to update the points with poor status of each subgroup.

Step 5, repeat step 4 until the maximum number of iterations is satisfied.

TABLE 2. Three species communication strategies.

No.	Equation
1	$Xbad_{i,j}^{G} = \omega Xbad_{i,j}^{G} + r_1(Xbest_m^G - Xbad_{i,j}^{G})$
2	$Xbad_{i,j}^{G} = \omega Xbad_{i,j}^{G} + r_2((Xbest_m^G + Xbest_n^G)/2 - Xbad_{i,j}^{G})$
3	$Xbad_{i,j}^{G} = \omega Xbad_{i,j}^{G} + r_3((Xbest_m^G + Xbest_n^G + Xbest_t^G)/3 - Xbad_{i,j}^{G})$

The pseudo-code of the algorithm is shown below.

Algorithm 1 Shows the Pseudo Code of MM-QUATRE Algorithm

//**initialization** Initialize the searching space V, dimension D, Set the generation counter $Gen = 1$, the ps individuals are randomly divided into four subgroups $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$ and $\mathbf{X_4}$ on average, and evaluate fitness values of all individuals, and intergroup communication step size $GenCS$.
//**MainLoop** Four subgroups are randomly generated and the position matrix of the particles in the four subgroups is initialized($\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, $\mathbf{X_4}$).
Evaluate fitness values of all individuals.
while (The number of iterations is not satisfied) **do**
    Generate the evolution matrix $\mathbf{M}$, mutation matrix $\mathbf{B}$ and matrix $\overline{\mathbf{M}}$.
    Evolve individuals in each group using Equation 1.
    Evaluate fitness values of all individuals.
    for $(i = 1; i \leq 4; i++)$ **do**
        for $(j = 1; j \leq ps; j++)$ **do**
            if $f(\mathbf{X_{i,j}}) \leq f(\mathbf{Xpbest_{i,j}})$ **then**
                $\mathbf{Xpbest_{i,j}} = \mathbf{X_{i,j}}$
            end if
        end for
    end for
    $\mathbf{X} = \mathbf{X_{pbest,j}}$, $\mathbf{Xgbet} = opt(\mathbf{Xpbest_{i,j}})$.
    while Gen % $GenCS$==0 **do**
        Select a random strategy from TABLE 2 to update the bad point.
    end while
end while

Output: The global optimum $\mathbf{X}_{gbest}$, global best fitness value $f(\mathbf{X}_{gbest})$.

B. OUR PROPOSED ALGORITHM APPLIED IN WSN LOCALIZATION BASED ON RSSI

In this section, the scheme of using MM-QUATRE localization in RSSI-based wireless sensor network node localization [38] is introduced. We know that the hop number of two anchor nodes is obtained by broadcasting information, and the distance between two nodes is estimated by RSSI value. Then the average distance of each hop between anchor nodes is calculated and the position of unknown node is estimated by least square method or maximum likelihood estimation. The estimated distance between the nodes is obtained by multiplying the hop value by the average hop of the anchor node. However, when the number of hops between anchor node and unknown node is greater than 2, this method has the problem of poor distance estimation, which leads to the decrease of positioning accuracy. The main purpose of the positioning problem is to minimize the estimation error and improve the positioning accuracy. To reduce the estimation error, an improved RSSI algorithm based on genetic algorithm is proposed for MM-QUATRE node localization in WSN. The algorithm first calculates the minimum hop number and distance between anchor nodes through the communication between them, and then calculates the average step length of each hop. Then the hops received by all anchor nodes are weighted to calculate the hops of unknown nodes [22]. Finally, the location of unknown nodes is estimated by the proposed MM-QUATRE algorithm.

For each unknown node, we use a step optimized by a completely independent MM-QUATRE algorithm to locate. The specific optimization process includes the following steps.

Step 1, calculate the distance between anchor nodes and jump out through RSSI value.

Step 2, then weighted the path received by all anchor nodes to calculate the location of unknown nodes.

Step 3, initializes the parameters used by MM-QUATRE (For example, population number ps, dimension D, etc.).

Step 4, initialize the position matrix ($\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, $\mathbf{X_4}$), mutation matrix $\mathbf{B}$ and evolution matrix $\mathbf{M}$ for all populations. Step 5, the best individual optimal position is selected to enter the next generation for the next evolution in each iteration. Repeat steps 2 through 4 until the stop condition is met.

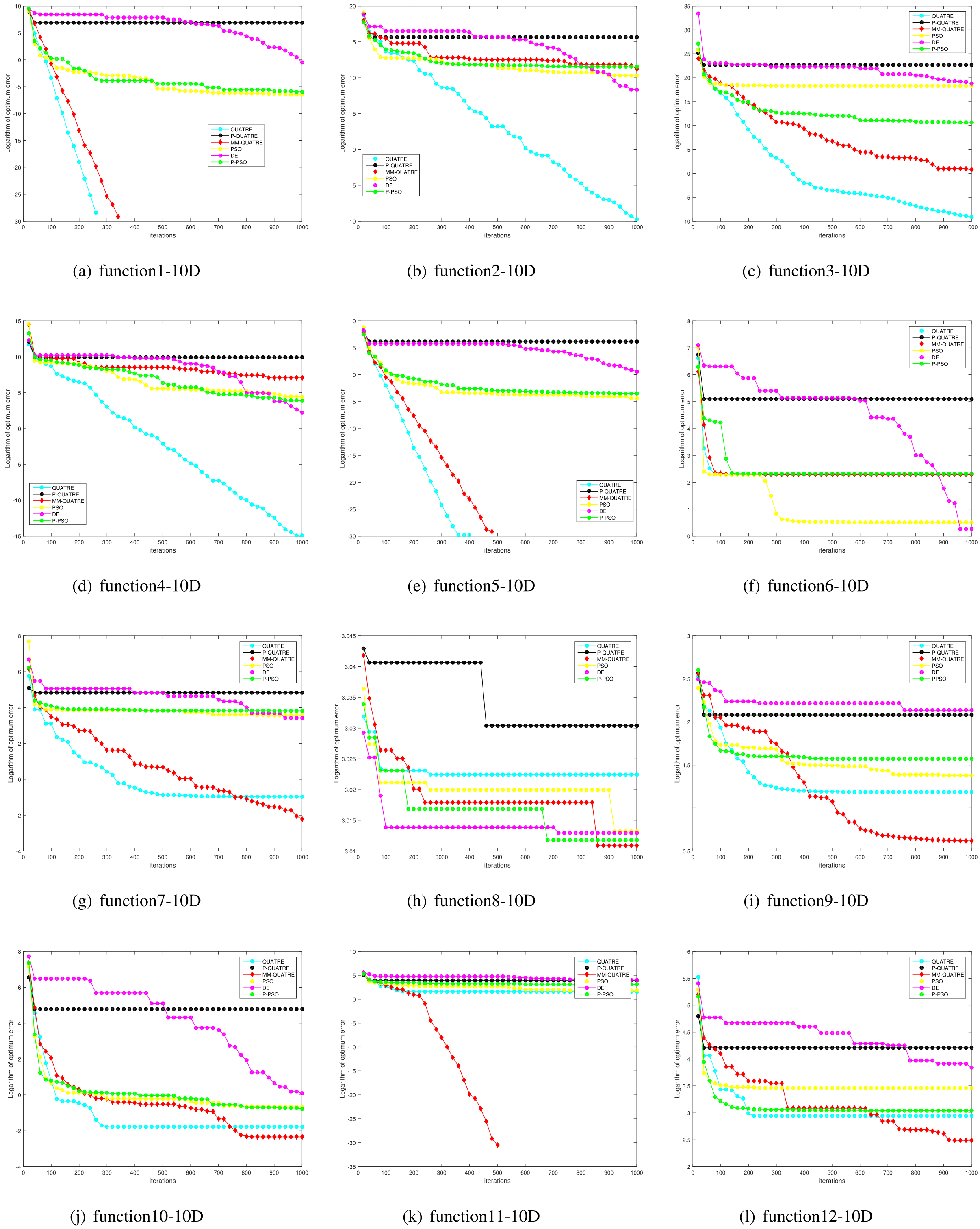
IV. EXPERIMENTAL ANALYSIS

In this section, we verify the performance of the newly proposed MM-QUATRE algorithm and its application in WSN node localization based on RSSI respectively through two groups of experimental data.

A. SIMULATION RESULTS ON A STANDARD BOUNDED CONSTRAINT BENCHMARK

The following experimental results we used the CEC2013 reference function set to verify the performance of our newly proposed MM-QUATRE algorithm. CEC2013 benchmark set [35]–[37] contains 28 test functions, including 5 single-peak functions ($f_1 - f_5$), 15 multi-peak functions ($f_6 - f_{20}$), and 8 conforming functions ($f_{21} - f_{28}$). All of these benchmark functions are moved to the same global minimum.

The proposed MM-QUATRE was compared with QUATRE, PSO [14], P-PSO [41], DE, and P-QUATRE



(a) function1-10D (b) function2-10D (c) function3-10D

(d) function4-10D (e) function5-10D (f) function6-10D

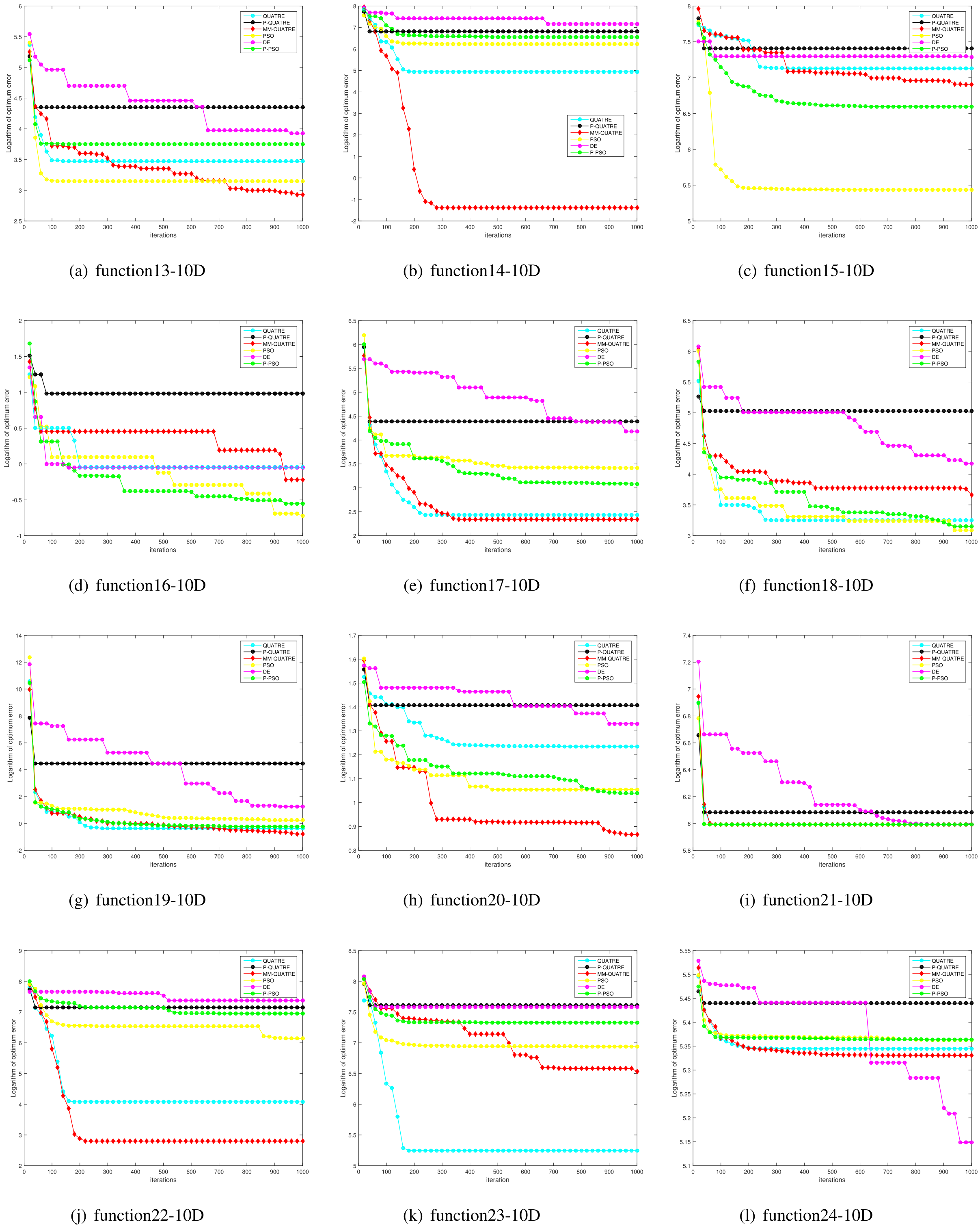
(g) function7-10D (h) function8-10D (i) function9-10D

(j) function10-10D (k) function11-10D (l) function12-10D

FIGURE 2. **Comparison of the best of fitness for functions $f_1 - f_{12}$ with 10D optimization.**

algorithms. These algorithms set the dimension as 10, the search area as 10 dimensions [−100, 100], and the swarm is composed of 200 particles. There are 28 test functions in total, and each test function is run 51 times. The mean and variance of errors are shown in TABLE 3 - 4. The bolded value is the minimum error obtained by the six algorithms.

According to the data shown in TABLE 3 - 4, from the perspective of optimization accuracy, MM-QUATRE is



(a) function13-10D (b) function14-10D (c) function15-10D

(d) function16-10D (e) function17-10D (f) function18-10D

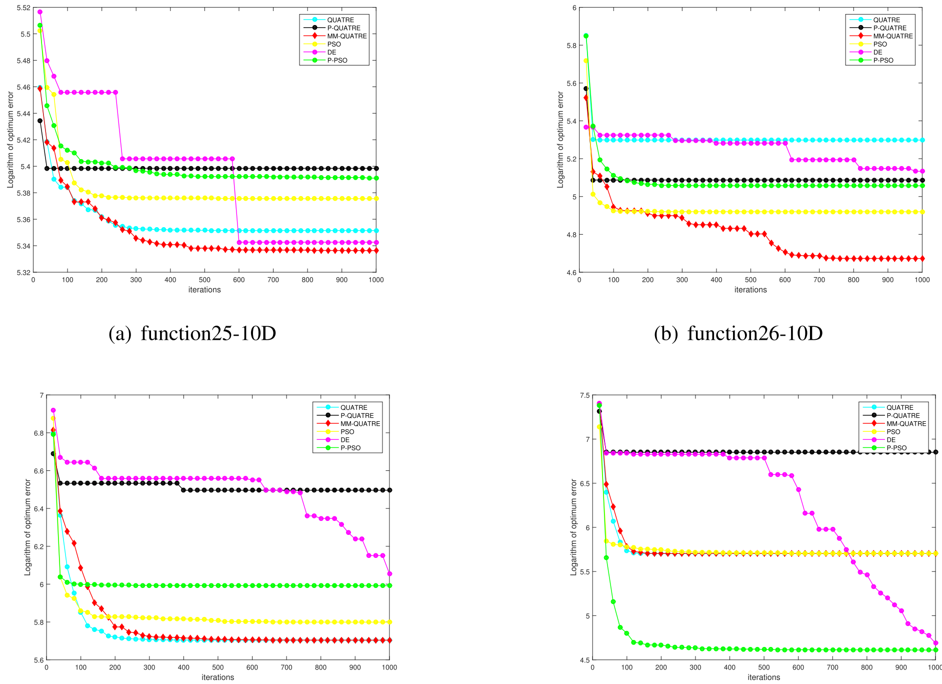
(g) function19-10D (h) function20-10D (i) function21-10D

(j) function22-10D (k) function23-10D (l) function24-10D

FIGURE 3. Comparison of the best of fitness for functions $f_{15} - f_{24}$ with 10D optimization.

obviously better to the other four relative algorithms. The performance of MM-QUATRE algorithm in benchmark these functions (f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{19}, f_{20}, f_{22}, f_{26}, f_{28}) is superior to that of other algorithms. When MM-QUATRE, P-QUATRE and QUATRE algorithm take (f_1, f_5) as the fitness function, the same optimal value can be obtained.

MM-QUATRE algorithm and other Swarm algorithms each fitness function 51 times running results in the



(a) function25-10D

(b) function26-10D

(c) function27-10D

(d) function28-10D

FIGURE 4. **Comparison of the best of fitness for functions $f_{25} - f_{28}$ with 10D optimization.**

TABLE 3. **The mean of error value of 51 runs of six algorithms in 28 benchmark functions of CEC2013.**

10D	MM-QUATRE	P-QUATRE	QUATRE	DE	PSO	P-PSO
f_1	**0**	**0**	**0**	0.739818664	0.001206135	0.005866406
f_2	97179.04063	85426.85137	**0.000557889**	3294.711252	92638.39054	68384.17118
f_3	1167.774295	100.7948122	**3.455801074**	102226375	34310800.78	22485901.69
f_4	1384.56593	2077.745298	**2.30E-07**	5.63E+00	66.4480108	59.56889992
f_5	**0**	**0**	**0**	1.318964608	0.020537998	0.030698153
f_6	4.045268178	2.6863875	4.70439812	**1.780849688**	8.74276643	5.648167836
f_7	0.202892228	**0.189106124**	0.871850669	24.56850794	27.95464676	22.46067368
f_8	20.36293841	20.36123257	20.460242	20.35903748	**20.34202936**	20.36059799
f_9	1.49175654	**0.977205986**	2.158101512	8.975566214	4.717156373	4.74142257
f_{10}	**0.150189455**	0.154159211	0.165848518	1.007661801	0.386370943	0.440301328
f_{11}	**0.858396049**	1.290646039	3.121440179	47.24541785	20.75641012	17.36177512
f_{12}	**9.982011525**	10.32129108	12.3719477	47.24900851	26.75096215	23.35983185
f_{13}	**12.95545372**	13.27536858	20.9775093	46.72612521	35.38855834	34.7133315
f_{14}	**11.20780248**	21.72924653	115.1742355	1555.62719	632.7795179	634.8677794
f_{15}	**661.6747356**	723.0923319	1010.954331	1497.579254	697.9260472	710.5917722
f_{16}	0.980289899	1.026311338	1.178023141	1.14221882	0.92721363	**0.456921219**
f_{17}	9.948205327	**8.694712367**	11.10988824	61.60986312	26.9153558	24.04218847
f_{18}	30.3124868	29.78515043	32.85601671	63.50124614	30.48091278	**24.06704791**
f_{19}	**0.389129739**	0.455030829	0.602447026	3.922956861	1.038193434	0.943207523
f_{20}	**2.490615022**	2.638410348	2.912626037	3.769369258	3.152868287	2.956356822
f_{21}	335.4271638	**313.8433311**	388.4177574	319.5817635	396.2759849	372.8123593
f_{22}	**69.99601412**	75.89132455	160.6812678	1788.963154	915.4123134	912.6888066
f_{23}	727.6704769	**672.3238626**	1052.247758	1825.171155	1161.54102	1085.535174
f_{24}	199.4465603	198.3984723	204.208444	**185.3117496**	216.4016311	211.7053517
f_{25}	202.290113	**194.4194039**	204.0862553	194.9544787	212.8115209	209.0458773
f_{26}	**120.6848264**	122.091166	167.3376274	176.1076494	188.0611368	150.795882
f_{27}	309.5792775	**307.7345055**	356.8828837	414.6347365	389.3369444	373.633645
f_{28}	**249.0196078**	268.627451	292.1568627	292.1568627	532.1037462	404.9964508
Optimal number	13	9	5	2	1	2

Wilcoxon-signed test [42], [43] values are shown in TABLE 5. When the data in the table is less than 0.05, it can be explained that the MM-QUATRE algorithm and the corresponding swarm intelligent algorithm have obvious performance differences under the corresponding test function. Combining TABLE 3 and TABLE 5 can

TABLE 4. The standard deviation of error value of 51 runs of six algorithms in 28 benchmark functions of CEC2013.

10D	MM-QUATRE	P-QUATRE	QUATRE	DE	PSO	P-PSO
f_1	**0**	**0**	**0**	0.553097344	0.030465436	0.064137751
f_2	223.2882947	227.7023242	**0.031256179**	39.53077814	368.4373301	250.768226
f_3	72.15042011	25.66380787	**4.563251311**	7036.001917	8662.037869	6589.996409
f_4	25.40869758	29.61377266	**0.000481097**	1.566941975	5.695039596	5.581055916
f_5	**0**	**0**	**0**	0.647032457	0.084629495	0.098560686
f_6	2.207977092	2.072892415	2.214146767	**1.0868185870**	3.261122791	2.14770591
f_7	0.556510531	**0.493862376**	1.846909388	2.5357872	5.327796701	4.005272813
f_8	0.278339135	**0.253359877**	0.297664348	0.272467633	0.254480119	0.271608927
f_9	1.073322005	0.86829874	1.216754739	**0.730037763**	1.16114321	1.113794239
f_{10}	0.278787558	**0.269126104**	0.311588276	0.323387221	0.409572263	0.40479914
f_{11}	**0.85646253**	1.063096457	1.180347979	2.433841846	3.163871213	2.81657347
f_{12}	2.024263825	**1.931524269**	2.350996393	2.468848614	3.312822166	2.999766956
f_{13}	2.286321234	**2.168703013**	2.916181583	2.89727787	3.652889021	3.189610977
f_{14}	**4.344373313**	5.261460762	9.88606657	10.509522	14.38383149	16.0048212
f_{15}	15.64755761	15.47987471	17.21070425	**13.34707944**	17.05101991	14.39697065
f_{16}	0.529503553	0.493481027	0.532731125	0.467443422	0.485736025	**0.439964197**
f_{17}	**1.311285064**	1.932611472	1.594395429	2.527430738	2.451369361	2.421130794
f_{18}	2.197662913	**2.137044772**	2.912464628	2.518495367	2.255879837	2.365670811
f_{19}	**0.312634365**	0.325597517	0.422676988	0.766463163	0.571669117	0.587568243
f_{20}	**0.519096237**	0.57774824	0.75640529	0.631154329	0.686037473	0.765216064
f_{21}	9.920935688	10.55857928	6.897329322	10.22717336	**5.289606443**	8.801724082
f_{22}	**7.255304549**	7.681168888	9.648994865	11.92278243	16.62282517	16.63312218
f_{23}	16.09907899	15.06915411	18.8291719	**12.59161581**	19.2327294	18.73009748
f_{24}	4.739746654	4.121144956	**3.78336172**	4.329803357	3.997551501	4.619925543
f_{25}	4.865604928	4.855596899	**3.736610336**	4.104771795	4.048598108	4.552686359
f_{26}	5.754676771	5.397209641	7.370737924	**4.592562559**	7.937243473	6.096502176
f_{27}	6.35164002	5.726179784	9.446211786	**4.32465564**	9.548782232	6.026126302
f_{28}	8.067164145	8.570764525	6.261616593	**1.700352932**	13.65905571	13.63953596
Optimal number	9	7	7	7	1	1

TABLE 5. Wilcoxon-signed test MM-QUATRE vs. other swarm intelligence algorithms.

n(51)	P-QUATRE	QUATRE	DE	P-PSO	PSO
f_1	1	1	5.15E-10	5.15E-10	5.15E-10
f_2	0.192604843	5.15E-10	5.15E-10	0.001743677	0.059555935
f_3	0.735781612	4.78E-06	5.15E-10	5.15E-10	5.15E-10
f_4	1.04E-04	5.15E-10	5.15E-10	5.15E-10	5.15E-10
f_5	1	1	5.15E-10	5.15E-10	5.15E-10
f_6	0.226594765	0.128399292	0.063461193	6.68E-04	3.99E-06
f_7	0.843950553	0.055852182	5.15E-10	5.15E-10	5.15E-10
f_8	0.851292663	2.42E-06	0.866014996	0.866014996	0.086282096
f_9	0.005854892	0.014426086	5.15E-10	9.87E-10	9.87E-10
f_{10}	0.076463432	0.3158797	5.15E-10	1.18E-09	1.67E-08
f_{11}	0.033060544	1.46E-08	5.14E-10	5.15E-10	5.15E-10
f_{12}	0.037320855	0.016412659	5.15E-10	1.87E-09	1.57E-09
f_{13}	0.033902494	5.46E-06	5.15E-10	6.53E-10	1.18E-09
f_{14}	0.015190735	9.87E-10	5.15E-10	5.15E-10	5.15E-10
f_{15}	0.030691918	1.18E-07	5.15E-10	0.260667003	0.728727848
f_{16}	0.393667354	0.000131105	0.000121479	1.32E-09	0.3158797
f_{17}	0.97009123	8.87E-06	5.15E-10	5.15E-10	5.15E-10
f_{18}	0.693813168	0.100931022	5.15E-10	3.99E-06	0.721696977
f_{19}	0.001800151	1.18E-07	5.15E-10	5.46E-10	5.15E-10
f_{20}	0.023883042	0.000131105	5.15E-10	2.68E-05	1.20E-08
f_{21}	0.345945826	0.002837545	0.189423869	2.21E-06	5.15E-10
f_{22}	0.015193737	7.78E-06	5.15E-10	5.15E-10	5.15E-10
f_{23}	0.311378182	7.45E-06	5.15E-10	2.42E-06	3.25E-07
f_{24}	0.567470026	0.004781253	2.68E-05	7.13E-06	8.65E-09
f_{25}	0.024472312	0.001688836	0.006943156	3.04E-06	2.53E-07
f_{26}	0.028526112	2.53E-06	4.17E-09	1.01E-05	9.66E-09
f_{27}	0.017716664	0.955149732	1.05E-09	5.23E-09	2.72E-08
f_{28}	0.0309375	0.007385254	8.25E-08	1.34E-08	1.25E-09

confirm the performance of our proposed MM-QUATRE algorithm.

The best one-time optimization process of six algorithms based on 28 benchmark functions of CEC2013 is shown in FIGURE 2 - 4. In overall 28 functions, according to the final optimization results show that the performance of MM-QUATRE algorithm in 14 benchmark functions (f_7, f_8, f_9, f_{10}, $f_{11}, f_{12}, f_{13}, f_{14}, f_{17}, f_{19}, f_{20}, f_{22}, f_{25}, f_{26}$) is superior to other

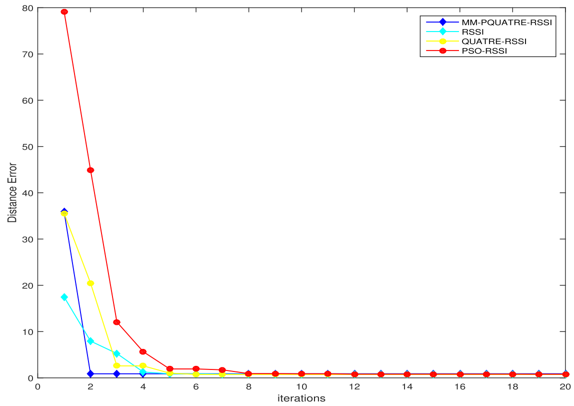


FIGURE 5. Convergence curve of fitness function value of SI-RSSI algorithms.

TABLE 6. The error between the estimated distance of the positioning algorithm and the real node position.

MM-QUATRE-RSSI	QUATRE-RSSI	PSO-RSSI	RSSI
2.2827101	2.571757017	2.41036322	2.849474

algorithms. Among the 4 benchmark functions (f_1, f_5, f_{21}, f_{27}), MM-QUATRE obtains the same optimum error compared with other algorithms, but MM-QUATRE have slower speed that convergence to the optimum error speed compare with other algorithms in these benchmark functions. Only ten functions (f_2, f_3, f_4, f_6, f_{15}, f_{16}, f_{18}, f_{23}, f_{24}) are inferior to other algorithms in the optimization ability of MM-QUATRE. When f_1,f_5, and f_{11} are used as benchmark functions, some algorithms have achieved optimal results, so the error is 0. After the logarithmic operation, it becomes negative infinity, so the interruption phenomenon in the figure appears.

B. SIMULATION RESULTS OF APPLIED MM-QUATRE TO NODE LOCALIZATION IN WSN BASED ON RSSI

This section mainly presents the simulation results of the practical application of the proposed MM-QUATRE algorithm to the localization of RSSI nodes [38], and compares them with RSSI, QUATRE-RSSI, and PSO-RSSI localization algorithms. The simulation results are shown in FIGURE 5 and TABLE 6. Where, FIGURE 5 shows the error variation diagram of positioning of each unknown node, and the result in TABLE 6 shows the error after positioning of all nodes.

The node layout area adopted in this simulation is a two-dimensional plane of 1000m $\times$ 1000m. The total number of nodes is 300, including 60 anchor nodes and 240 unknown nodes. The communication radius of nodes is 200 meters.

As shown in FIGURE 5, the curve is the convergence of global optimization of each swam intelligent algorithm with eq.8 as a fitness function. MM-QUATRE-RSSI algorithm has a faster localization speed and stronger convergence ability than other algorithms. TABLE 6 is the average error of the unknown and actual true positions of the unknown nodes estimated by the intelligent algorithm. According to the data shown in TABLE 6 MM-QUATRE-RSSI algorithm performs better than other algorithms. In conclusion, the newly proposed MM-QUATRE algorithm has significant advantages in the localization of WSN nodes based on RSSI.

V. CONCLUSION

In this paper, a new QUATRE algorithm for Multi-group and Multi-choice bad point updating strategy is proposed. During the implementation of MM-QUATRE algorithm, the group was divided into four subgroups to improve the diversity of optimization ability. Each subgroup completes iterative evolution independently, and carries out inter-group communication for 50 times every iteration, and updates the bad point status of each subgroup according to the communication information, so that the bad point can regain the ability to search for optimization. In addition, the random selection strategy in the process of updating the bad point increases more possibilities for finding the globally optimal position. The CEC2013 test suite was used to confirm the ability of the algorithm to search global optimization. The experimental results show that MM-QUATRE algorithm is superior to other algorithms not only in convergence speed, but also convergence performance. In order to improve the accuracy of RSSI algorithm in node localization, we combined MM-QUATRE algorithm into RSSI algorithm. In this application, we first refine the number of hops of anchor nodes and calculate the distance between anchor nodes with RSSI value of signal transmission between anchor nodes, and then estimate the location of unknown nodes with MM-QUATRE algorithm. Simulation results show that the proposed MM-QUATRE-RSSI algorithm has higher accuracy than RSSI, PSO-RSSI and QUATRE-RSSI algorithms.

In the future work, we will further modify the variation scheme and communication strategy adopted in the progress, so as to improve the performance of evolutionary algorithm and swarm intelligence algorithm. We also need to apply the subsequent improved algorithm to different types of application scenarios, such as clustering methods in WSN problems, hierarchical routing, and deployment and coverage problems in WSN.

REFERENCES

[1] A. E. Eiben and M. Schoenauer, "Evolutionary computing," *Inf. Process. Lett.*, vol. 82, no. 1, pp. 1–6, 2002.
[2] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowl.-Based Syst.*, vol. 168, no. 9, pp. 80–99, Mar. 2019.
[3] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.
[4] C.-H. Chen, F.-J. Hwang, and H.-Y. Kung, "Travel time prediction system based on data clustering for waste collection vehicles," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 7, pp. 1374–1383, Jul. 2019.
[5] J. C. Bezdek, "What is computational intelligence?" Washington, DC, USA, Tech. Rep., 1994.
[6] J. Jang, C. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing—A computational approach to learning and machine intelligence [book review]," *IEEE Trans. Autom. Control*, vol. 42, no. 10, pp. 1482–1484, Oct. 1997.

[7] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, Guilin, China, Aug. 2006, pp. 854–858.
[8] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, and J. Xie, "Novel systolization of subquadratic space complexity multipliers based on Toeplitz matrix-vector product approach," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1614–1622, Jul. 2019.
[9] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "A compact bat algorithm for unequal clustering in wireless sensor networks," *Appl. Sci.*, vol. 9, no. 10, p. 1973, May 2019.
[10] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 203–216, Apr. 2019.
[11] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.
[12] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.
[13] X. Xue, J. Chen, and X. Yao, "Efficient user involvement in semiautomatic ontology matching," *IEEE Trans. Emerg. Topics Comput. Intell.*, to be published, doi: 10.1109/TETCI.2018.2883109.
[14] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[15] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 459–472, Jun. 2019.
[16] S. C. Chu, J.-F. Chang, J. F. Roddick, and J.-S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *J. Inf. Sci. Eng.*, vol. 21, no. 4, pp. 809–818, 2005.
[17] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 652–657, Jan. 2011.
[18] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-S. Pan, "Multi-strategy ensemble artificial bee colony algorithm," *Inf. Sci.*, vol. 279, pp. 587–603, Sep. 2014.
[19] P.-W. Tsai, M. K. Khan, J.-S. Pan, and B.-Y. Liao, "Interactive artificial bee colony supported passive continuous authentication system," *IEEE Syst. J.*, vol. 8, no. 2, pp. 395–405, Jun. 2014.
[20] K. V. Price, "Differential evolution," in *Handbook of Optimization*. Berlin, Germany: Springer, 2013, pp. 187–214.
[21] Z. Meng, J.-S. Pan, and H. Xu, "QUasi-affine transformation evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization," *Knowl.-Based Syst.*, vol. 109, pp. 104–121, Oct. 2016.
[22] N. Liu, J.-S. Pan, J. Wang, and T.-T. Nguyen, "An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks," *Sensors*, vol. 19, no. 19, p. 4112, Sep. 2019.
[23] Z. Meng and J.-S. Pan, "A competitive quasi-affine transformation evolutionary (C-QUATRE) algorithm for global optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Budapest, Hungary, Oct. 2016, pp. 1644–1649.
[24] J.-S. Pan, Z. Meng, H. Xu, and X. Li, "Quasi-affine transformation evolution (QUATRE) algorithm: A new simple and accurate structure for global optimization," in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.*, Morioka, Japan, Aug. 2016, pp. 657–667.
[25] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, no. 9, Dec. 2018, Art. no. 9472075.
[26] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 3, Mar. 2019, Art. no. 155014771983958.
[27] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, Jun. 2019, doi: 10.3390/s19112579.
[28] K. Jia and Z. Wei, "Water conservancy monitoring based on visual sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 6, Jun. 2018, Art. no. 155014771877957.
[29] F. Zhang, G. Ding, L. Xu, B. Chen, and Z. Li, "An effective method for the abnormal monitoring of stage performance based on visual sensor network," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 4, Apr. 2018, Art. no. 155014771876957.
[30] J. Wang, Y. Gao, W. Liu, W. Wu, and S.-J. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Comput., Mater. Continua*, vol. 58, no. 3, pp. 711–725, 2019.
[31] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, and V. Snášel, "α-fraction first strategy for hierarchical model in wireless sensor networks," *J. Internet Technol.*, vol. 19, no. 6, pp. 1717–1726, 2018.
[32] J. Wang, X. J. Gu, W. Liu, A. K. Sangaiah, and H. J. Kim, "An empower hamilton loop based data collection algorithm with mobile agent for WSNs," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 18, 2019, doi: 10.1186/s13673-019-0179-4.
[33] J. Li and D. Wang, "The security DV-hop algorithm against multiple-wormhole-node-link in WSN," *Proc. TIIS*, vol. 13, no. 4, pp. 2223–2242, 2019.
[34] J. Li, D. Wang, and Y. Wang, "Security DV-hop localisation algorithm against wormhole attack in wireless sensor network," *IET Wireless Sensor Syst.*, vol. 8, no. 2, pp. 68–75, Apr. 2018.
[35] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2337–2344.
[36] R. Tanabe and A. Fukunaga, "Evaluating the performance of shade on CEC 2013 benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1952–1959.
[37] J. Tvrdík and R. Poláková, "Competitive differential evolution applied to CEC 2013 problems," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1651–1657.
[38] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "A novel approach to indoor RSSI localization by automatic calibration of the wireless propagation model," in *Proc. VTC Spring-IEEE 69th Veh. Technol. Conf.*, Barcelona, Spain, Apr. 2009 pp. 1–5.
[39] A. Awad, T. Frunzke, and F. Dressler, "Adaptive distance estimation and localization in WSN using RSSI measures," in *Proc. 10th Euromicro Conf. Digit. Syst. Design Archit., Methods Tools (DSD)*, Lübeck, Germany, Aug. 2007, pp. 471–478.
[40] F. Caballero, L. Merino, P. Gil, I. Maza, and A. Ollero, "A probabilistic framework for entire WSN localization using a mobile robot," *Robot. Auto. Syst.*, vol. 56, no. 10, pp. 798–806, Oct. 2008.
[41] A. W. McNabb, C. K. Monson, and K. D. Seppi, "Parallel PSO using MapReduce," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 7–14.
[42] R. F. Woolson, "Wilcoxon signed-rank test," in *Wiley Encyclopedia of Clinical Trials*. Hoboken, NJ, USA: Wiley, 2007, pp. 1–3.
[43] D. Rey and M. Neuhäuser, "Wilcoxon-signed-rank test," *Int. Encyclopedia Stat. Sci.*, pp. 1658–1659, 2011.

ZHI-GANG DU received the B.S. degree from the Qindao College, Qingdao Technological University, Qingdao, China, in 2017. He is currently pursuing the master's degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests include swarm intelligence and artificial neural networks.

JENG-SHYANG PAN received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Director of the Fujian Provincial Key Laboratory of Big Data Mining and Applications and an Assistant President with the Fujian University of Technology. He is also a Professor with the Harbin Institute of Technology. He is also a Professor and a Doctoral Supervisor with the College of Computer Engineering, Shandong University of Science and Technology. He is an IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section. He was offered Thousand Talent Program in China, in 2010.

SHU-CHUAN CHU received the Ph.D. degree from the School of Computer Science, Engineering and Mathematics, Flinders University, Australia, in 2004. She joined Flinders University, in December 2009, after nine years at Cheng Shiu University, Taiwan. She has been a Research Fellow with the College of Science and Engineering, Flinders University, since December 2009. She has been a Research Fellow, with PhD advisor, at the College of Computer Science and Engineering, Shandong University of Science and Technology, since September 2019. Her research interests mainly include swarm intelligence, intelligent computing, and data mining.

HAN-JIANG LUO received the Ph.D. degree from the Department of Computer Science, Ocean University of China. He was a Visiting Scholar with HKUST, in 2014. He is currently with the School of Computer Science and Engineering, Shandong University of Science and Technology. He has published research articles in some prestigious journals and conferences, such as *ACM Computing Surveys*, the IEEE Communications Surveys and Tutorials, IEEE TPDS, IEEE TVT, and IEEE ICDCS. His research interests include wireless sensor networks, underwater wireless sensor networks, the Internet of Things, smart port, data science, and artificial intelligence.

PEI HU received the M.Eng. degree in computer application technology from Central South University, Changsha, China, in 2009. He is currently pursuing the Ph.D. degree with the Shandong University of Science and Technology, Qingdao, China, under the supervision of Dr. J.-S. Pan, where he is a Lecturer. He has published several SCI articles on swarm intelligence. His research interests include evolutionary computation and neural networks.