Liu *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:175
https://doi.org/10.1186/s13638-019-1481-6

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A bi-population QUasi-Affine TRansformation Evolution algorithm for global optimization and its application to dynamic deployment in wireless sensor networks

Nengxian Liu[1], Jeng-Shyang Pan[1,2,3*] and Trong-The Nguyen[2,4]

Abstract

In this paper, we propose a new Bi-Population QUasi-Affine TRansformation Evolution (BP-QUATRE) algorithm for global optimization. The proposed BP-QUATRE algorithm divides the population into two subpopulations with sort strategy, and each subpopulation adopts a different mutation strategy to keep the balance between the fast convergence and population diversity. What is more, the proposed BP-QUATRE algorithm dynamically adjusts scale factor with a linear decrease strategy to make a good balance between exploration and exploitation capability. We compare the proposed algorithm with two QUATRE variants, PSO-IW, and DE algorithms on the CEC2013 test suite. The experimental results demonstrate that the proposed BP-QUATRE algorithm outperforms the competing algorithms. We also apply the proposed algorithm to dynamic deployment in wireless sensor networks. The simulation results show that the proposed BP-QUATRE algorithm has better coverage rate than the other competing algorithms.

Keywords: Differential evolution, Particle swarm optimization, Bi-population, QUATRE algorithm, Global optimization, Dynamic deployment, Wireless sensor networks

1 Introduction

In the last few decades, there have been many optimization demands arising not only from the scientific community but also from various real-world applications. Generally, the approach to solving these optimization problems often begins with designing the objective function which can model the objectives of optimization problems [1]. Many optimization approaches have been proposed to meet these demands aiming at finding optimal solutions. Some Swarm-based intelligence optimization algorithms, such as particle swarm optimization (PSO) [2], ant colony optimization (ACO) [3], differential evolution (DE) [1], artificial bee colony (ABC) optimization [4], and QUasi-Affine TRansformation Evolution (QUATRE) algorithm [5], and so on, have been developed to tackle these complex optimization problems.

QUATRE algorithm was first presented by Meng et al. in [5] that discussed the relationship between QUATRE algorithm and the other two swarm-based intelligence algorithms PSO and DE. In 1995, Kennedy and Eberhart firstly introduced the PSO algorithm [2]. As PSO is simple, powerful, and straightforward to implement, many researchers have studied this technique and developed various improved variants [6–8]. DE was introduced by Storn and Price [1] in 1995, which was arguably one of the most powerful optimization algorithms. As well, many DE variants have been proposed to enhance the performance of DE algorithm [9–11], and QUATRE algorithm is one of them proposed to conquer representational or positional bias of DE

* Correspondence: jspan@cc.kuas.edu.tw; lylnx@fzu.edu.cn
[1]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[2]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China
Full list of author information is available at the end of the article

algorithm [12]. QUATRE's conventional notation is "QUATRE/x/y" which denotes types of QUATRE variants. It is worth noting that the notation of QUATRE is more general than the DE's notation "DE/x/y/z" [13].

The canonical QUATRE algorithm and its variants can be found in literatures [12–17]. The QUATRE has the advantages of simplicity, few control parameters to set, and convenient to be used, but it has some weaknesses as the DE algorithm such as it will be premature convergence, and will search stagnation and may be easily trapped into local optima. Population diversity plays important role in alleviating these weaknesses. Therefore, it is important to keep the balance between diversity and convergence. In [16], S-QUATRE has been proposed which uses sort strategy to improve the performance of QUATRE algorithm. And S-QUATRE divides the population into the better and the worse groups and evolves the individuals in the worse group. The other algorithms which partition population into two groups or several subpopulations to maintain population diversity and to enhance the performance of algorithms, such as CMA-ES, PSO, DE and CSO, can be found in previous literature [18–22]. On the other hand, both mutation strategies and control parameter scale factor *F* have significant effects on the performance of QUATRE variants. Different mutation strategies in QUATRE algorithm have different performance over various optimization problems [13] because different mutation strategy has different search ability and convergence rate. Usually, similar to the DE algorithm, adopting larger *F* value in QUATRE algorithm means the algorithm is more focused on exploration, while a smaller *F* value means more exploitation [23]. Therefore, in this paper, in order to improve the performance of QUATRE algorithm, we propose a novel Bi-Population QUATRE algorithm with a sort strategy and a linear decrease scale factor *F* (BP-QUATRE), and each subpopulation has a different mutation strategy.

The remainder of the paper is arranged as follows. In Section 2, we briefly review the QUATRE algorithm. Our proposed Bi-Population QUasi-Affine TRansformation Evolution (BP-QUATRE) algorithm is given in Section 3. In Section 4, we apply the proposed algorithm to dynamic deployment in wireless sensor networks. What is more, the experimental analysis of our proposed algorithm under CEC2013 test suite and simulation results in wireless sensor networks are presented in Section 4. Finally, Section 6 gives the conclusion.

2 Canonical QUATRE algorithm

Meng et al. have proposed the QUATRE algorithm for solving optimization problems [5]. QUATRE is an abbreviation of QUasi-Affine TRansformation Evolution, and the reason the authors naming the algorithm QUATRE is that individuals in QUATRE algorithm evolve by using an affine transformation-like equation. The detailed evolution equation of QUATRE is as follows.

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

where $\mathbf{M}$ is an evolution matrix and $\overline{\mathbf{M}}$ is a binary inverted matrix of M. The elements of them are either 0 or 1. The binary invert operation means to invert the values of the matrix. The reverse values of zero elements are ones, while the reverse values of one elements are zeros. Equation 2 shows an example of binary inverse operation.

$$\mathrm{M} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \dots & \\ 1 & 1 & \dots & 1 \end{bmatrix}, \overline{\mathrm{M}} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ & & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{2}$$

$\mathbf{M}$ is transformed from an initial matrix $\mathbf{M}_{\mathrm{ini}}$ which is initialized by a lower triangular matrix with the elements equaling to ones. Transforming from $\mathbf{M}_{\mathrm{ini}}$ to $\mathbf{M}$ contains two consecutive steps. In the first step, every element in each row vector of $\mathrm{M}_{\mathrm{ini}}$ is randomly permuted. In the second step, the sequence of the row vectors is randomly permuted with all elements of each row vector unchanged. An example of the transformation with ps = D is given in Eq. 3. Usually, the population size ps is larger than the dimension, while the matrix $\mathbf{M}_{\mathrm{ini}}$ needs to be extended according to population size ps. Equation 4 shows an example of ps = 2D + 2. Generally, when ps % D = k, the first k rows of the D × D lower triangular matrix are included in $\mathbf{M}_{\mathrm{ini}}$ and adaptively change $\mathbf{M}$ according to $\mathbf{M}_{\mathrm{ini}}$ [12].

$$M_{ini} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \dots & \\ 1 & 1 & \dots & 1 \end{bmatrix} \sim \begin{bmatrix} & 1 & & \\ & \dots & & \\ 1 & 1 & \dots & 1 \\ & 1 & 1 & \end{bmatrix} \\ = M \tag{3}$$

$$M_{ini} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \dots & \\ 1 & 1 & \dots & 1 \\ 1 & & & \\ 1 & 1 & & \\ & & \dots & \\ 1 & 1 & \dots & 1 \\ & & \vdots & \\ 1 & & & \\ 1 & 1 & & \end{bmatrix} \sim \begin{bmatrix} 1 & & \dots & 1 \\ & & \dots & 1 \\ & \dots & & \\ 1 & 1 & & \\ & 1 & & \\ 1 & 1 & \dots & 1 \\ & & 1 & \\ 1 & & & 1 \\ & & \vdots & \\ 1 & 1 & \dots & 1 \\ & 1 & & \end{bmatrix} \\ = M \tag{4}$$

$\mathbf{X} = [\mathrm{X}_{1,\ \mathrm{G}}, \mathrm{X}_{2,\ \mathrm{G}}, \dots, \mathrm{X}_{\mathrm{i},\ \mathrm{G}}, \dots, \mathrm{X}_{\mathrm{ps},\ \mathrm{G}}]^{\mathrm{T}}$is the population matrix with ps individuals. $\mathrm{X}_{\mathrm{i},\ \mathrm{G}} = [\mathrm{x}_{\mathrm{i}1}, \mathrm{x}_{\mathrm{i}2}, \dots, \mathrm{x}_{\mathrm{iD}}]$ is the *i*th row vector of the matrix $\mathbf{X}$, which denotes the location of *i*th individual of the *G*th generation, and each individual $\mathrm{X}_{\mathrm{i},\ \mathrm{G}}$ is a candidate solution for an optimization problem, and D denotes the dimension number of objective function, where $i \in \{1, 2, \ldots, ps\}$. The operation "⊗" stands for component-wise multiplication of the elements in each matrix, which is the same as ".*" operation in Matlab software. $\mathbf{B} = [B_{1,\ G}, B_{2,\ G}, \ldots, B_{i,\ G}, \ldots, B_{ps,\ G}]^T$ is the donor matrix, and it has several different calculation schemes (mutation strategies) which are listed in Eqs. (5)–(8) [7].

$$\text{QUATRE/best/1} : B = X_{gbest,G} + F \cdot \left(X_{r1,G}\text{-}X_{r2,G}\right) \tag{5}$$

$$\text{QUATRE/ rand/1} : B = X_{r0,G} + F \cdot \left(X_{r1,G}\text{-}X_{r2,G}\right) \tag{6}$$

$$\text{QUATRE/target/1} : B = \mathbf{X} + F \cdot \left(X_{r1,G}\text{-}X_{r2,G}\right) \tag{7}$$

$$\begin{aligned}\text{QUATRE/target-to-best/1B} \\ = \mathbf{X} + F \cdot \left(X_{gbest,G}\text{-}\mathbf{X}\right) + F \cdot \left(X_{r1,G}\text{-}X_{r2,G}\right)\end{aligned} \tag{8}$$

$\mathbf{X}_{gbest,\ G} = [X_{gbest,\ G}, X_{gbest,\ G}, \ldots, X_{gbest,\ G}]^T \mathbf{X}_{gbest,\ G} = [X_{gbest,\ G}, X_{gbest,\ G}, \ldots, X_{gbest,\ G}]^T$ denotes a row vector-duplicated matrix with each row vector equaling to the global best individual $X_{gbest,\ G}$ of the *G*th generation. F can be considered as amplification factor, whose value region is (0, 1] for most optimization problems. $\mathbf{X}_{r1,\ G}$, $\mathbf{X}_{r2,\ G}$ and $\mathbf{X}_{r3,\ G}$ are a set of random matrices which are generated by randomly permutating the sequence of row vectors in the matrix $\mathbf{X}$ of the *G*th generation.

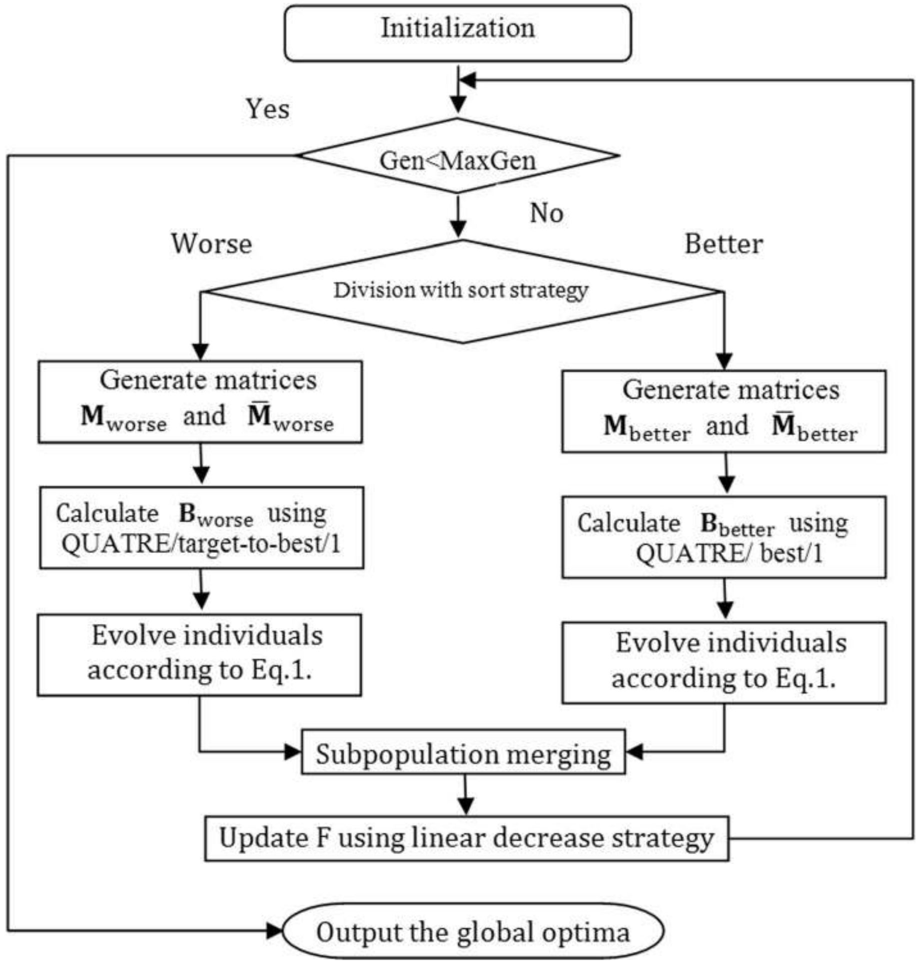
Initialization
Yes
Gen<MaxGen
No
Worse
Better
Division with sort strategy
Generate matrices $\mathbf{M}_{worse}$ and $\bar{\mathbf{M}}_{worse}$
Generate matrices $\mathbf{M}_{better}$ and $\bar{\mathbf{M}}_{better}$
Calculate $\mathbf{B}_{worse}$ using QUATRE/target-to-best/1
Calculate $\mathbf{B}_{better}$ using QUATRE/ best/1
Evolve individuals according to Eq.1.
Evolve individuals according to Eq.1.
Subpopulation merging
Update F using linear decrease strategy
Output the global optima

Fig. 1 The main framework of BP-QUATRE. The flowchart of BPQUATRE algorithm consists of population initialization, population division, subpopulation evolution, subpopulation merging and approach of updating parameter scale factor F. Gen is the current generation number and MaxGen is the maximum generation number. Better means the subpopulation with better fitness values (i.e., with the lower fitness values for a minimization problem). Worse means the subpopulation with worse fitness values. The better subpopulation evolves by adopting mutation strategy "QUATRE/best/1" to make good exploitation around the individuals with better fitness values and to have good convergence rate. The better subpopulation evolves by adopting mutation strategy "QUATRE/best/1" to make good exploitation around the individuals with better fitness values and to have good convergence rate. The worse subpopulation evolves by using mutation strategy "QUATRE/target-to-best/1" to make a good exploration around the individuals with worse fitness values and to preserve population diversity

3 Bi-population QUATRE algorithm with sort strategy and linear decrease scale factor F (BP-QUATRE)

In this section, we describe the main idea of our proposed algorithm BP-QUATRE. As mentioned above, because easily trapping into local optima and premature convergence is the weakness of QUATRE algorithm. In order to alleviate the above weaknesses, BP-QUATRE consisting of population initialization, population division, subpopulation evolution, subpopulation merging, and an approach to update the parameter scale factor *F*, is proposed in this paper. The main framework of BP-QUATRE is shown in Fig. 1.

3.1 Bi-population division and mutation strategies

Usually, as conventional evolutionary algorithms, the QUATRE algorithm suffers from the problem of premature convergence, i.e., the population is too early to lose diversity and fall into local optima. Multi-population approach helps to increase population diversity and alleviate premature convergence [20]. Inspired by this, we use a Bi-population approach to enhance the diversity of the population. In our proposed algorithm, the individuals in the population are firstly sorted after initialization according to the fitness values, and then the entire population is equally divided into two subpopulations based on the sorted sequence, say pop_{better} and pop_{worse}, respectively. As we know, different mutation strategy of QUATRE algorithm has different search abilities. Mutation strategy "QUATRE/best/1" uses the best individual to guide the population and has a fast convergence rate and good local search ability around the best individual. Therefore, the subpopulation pop_{better} evolves by adopting mutation strategy "QUATRE/best/1" to make good exploitation around the individuals with better fitness values and to have good convergence rate. On the other hand, mutation strategy "QUATRE/target-to-best/1" is a strong exploration-biased strategy, because this strategy generates donor individual using the best individual and two random selected individuals. Thus, the subpopulation pop_{worse} evolves by using mutation strategy "QUATRE/target-to-best/1" to make a good exploration around the individuals with worse fitness values and to preserve population diversity. Therefore, this bi-population division and different subpopulation having different mutation strategy approach can make a trade-off between the population diversity and convergence rate.

3.2 Linear decrease scale factor

Scale factor plays an essential role in balancing exploration and exploitation ability of QUATRE algorithm during the search phases. In [5], the authors illustrate the effect of different scaling factor values on the performance of the QUATRE algorithm. And there is no fixed parameter setting which can achieve the best performance for all kinds of problems. It is significant to find a good method to dynamically adjust the scaling factor value. According to [6, 24] for most population-based optimization algorithm, it is a good idea for the algorithm to have more exploration ability in the early stages of the search in order to sample diverse zones of the search space. In the later stages of the search, the algorithm should possess more exploitation ability to search the relatively small space where the potential global optimum lies in. Namely, at the beginning of the search, the scale factor of the algorithm should be larger. While with the increment of generations, the scale factor of algorithm should be decreased to increase the exploitation ability. Hence, we use the linear decrease strategy proposed in [6] to dynamically adjust the value of scale factor which can be described as fellow.

$$F = F_{\max} - (F_{\max} - F_{\min}) \cdot Gen/\text{MaxGen} \tag{9}$$

where $F_{\max}$ and $F_{\min}$ are the predetermined maximum and minimum values of scale factor F. *Gen* is the current generation number, and MaxGen is the maximum generation number.

The pseudo code of BP-QUATRE algorithm is shown in Algorithm 1.

Algorithm 1 BP-QUATRE Algorithm

```
Initialization:
    Initialize the searching space V, dimension D, Gen=1, the benchmark function f(X), randomly Initialize the population X with ps individuals, and evaluate the fitness values of all individuals.
Iteration:
while $Gen$ < MaxGen || !stopCriterion do
    Sort individuals based on fitness values and then divide them into $pop_{\text{better}}$ and $pop_{\text{worse}}$ according to sort order
    Generate matrices $\mathbf{M}_{\text{better}}$ and $\mathbf{M}_{\text{worse}}$ using Eq. 4. Generate matrices $\bar{\mathbf{M}}_{\text{better}}$ and $\bar{\mathbf{M}}_{\text{worse}}$ using Eq. 4.
    Calculate donor matrix $\mathbf{B}_{\text{better}}$ using Eq. 5. Calculate donor matrix $\mathbf{B}_{\text{worse}}$ using Eq. 8.
    Evolve individuals in each sub-population using Eq. 1.
    Evaluate fitness values of all individuals.
    for $i = 1$ to ps do
        if $f(X_i) \leq f(X_{\text{pbest},i})$ then
            $X_{\text{pbest},i} = X_i$
        end if
    end for
    $\mathbf{X} = \mathbf{X}_{\text{pbest}}$ , $X_{\text{gbest}} = \text{opt}\{\mathbf{X}_{\text{pbest}}\}$.
    Update scale factor F according to Eq.9.
    $Gen = Gen + 1$
end while
Output:
    The global optimum $X_{\text{gbest}}$, global best fitness value $f(X_{\text{pbest}})$.
```

4 Apply the proposed BP-QUATRE algorithm to dynamic deployment in wireless sensor networks

In this section, we apply the proposed BP-QUATRE algorithm to dynamic deployment in wireless sensor networks (WSN). The WSN becomes a popular research field [25–29] due to its great value in real-world applications such as environment monitoring, healthcare applications, and forest fire detection. The WSN is composed of a large number of battery-powered, multifunctional, and resources-constrained sensor nodes. The performance of the whole WSN depends on the position of the sensors which affect the coverage, connectivity, energy efficiency, and network lifetime. In some applications, the locations of the sensors are predetermined by static ways. However, in some cases such as battlefield, underwater, and disaster-affected regions where is difficult to predetermine the locations by static ways, only dynamic deployment strategies can be adopted. In dynamic deployment, the sensors are first randomly placed within the area of interest and then sensors can relocate their locations by using information from other sensor nodes. But random initial deployment may not ensure effective coverage. In order to enhance the coverage rate of the whole WSN, a number of algorithms have been developed for dynamic node deployment, including virtual force [30], Voronoi diagram [31], and swarm intelligence algorithms [33–36]. Many swarm intelligence algorithms are employed in sensor deployment, such as the particle swarm optimization (PSO) [32, 33], artificial bee colony algorithm (ABC) [34], differential evolution (DE) [35], and so forth. In this study, the proposed QUATRE algorithm is first applied to dynamic deployment in WSN with the aim of improving the coverage rate. The proposed algorithm is compared with PSO-based and DE-based dynamic deployment algorithm.

4.1 Sensor detection model

Without losing generality, this paper assumes that each sensor node can move and has the same sensor radius and communication range. There are two sensor detection models in wireless sensor networks: binary detection model and probability detection model [36]. In the binary detection model, the detected possibility of the event concerned is 1 within the sensing radius. Otherwise, the probability is 0. This model can be expressed by the Eq. 9 [37].

$$C_{xy}(P, s_i) = \begin{cases} 1, & d(P, s_i) < r \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where r represents sensor radius and $d(P, s_i)$ denotes the Euclidean distance between point P and the sensor node s_i. Although the binary sensor model is relatively simple, the uncertainties in measurement are not taken into account. Generally, sensor detections are imprecise in practical, so the detection probability $C_{xy}(P, s_i)$ needs to be presented in probabilistic terms. Therefore, we use the probabilistic detection model in the paper, which can be expressed by the Eq. 10 [38].

$$C_{x,y}(P, s_i) = \begin{cases} 1, & d(P, s_i) \le r - r_e \\ e^{\left(-\alpha_1 \lambda_1{}^{\beta_1} / \lambda_2{}^{\beta_2} + \alpha_2\right)}, & r - r_e < d(P, s_i) \le r + r_e \\ 0, & d(P, s_i) > r + r_e \end{cases} \tag{11}$$

where $r_e (0 < r_e < r)$ is the measure of uncertainty. α_1, α_2, β_1, and β_2 are detection parameters related to the characteristics of sensors. $\lambda_1 = r_e - r + d(P, s_i)$ and $\lambda_2 = r_e + r - d(P, s_i)$ are the input parameters. In general, the detection probability covered by sensor node may be less than 1. This means that it is necessary to overlap the sensor detection area to compensate for the potential low detection probability [39]. And we assume that sensors observe independently. Considering a point P (x, y) in the overlap region of a set of sensors S, the joint detection probability of point P can be calculated by the Eq. 11.

$$C_{x,y}(S) = 1 - \prod_{s_i \in S} \left(1 - C_{x,y}(P, s_i)\right) \tag{12}$$

Let C_{th} is the threshold of predefined effective detection probability. This implies that the point P (x, y) can be effectively covered if

$$\min_{x,y} \left\{ C_{x,y}(S) \right\} \ge C_{th} \tag{13}$$

4.2 Dynamic deployment based on BP-QUATRE algorithm

The purpose of sensor deployment algorithm is to determine an optimal sensor distribution in the region of interest, which is similar to the swarm intelligence algorithm for solving complex optimization problems. Therefore, it is possible to apply BP-QUATRE algorithm to the dynamic deployment problem of WSN.

In the BP-QUATRE algorithm, the individual is composed of the coordinate representing its position in the solution space. In dynamic deployment, the individual represents the deployment of the sensors in the sensed area. Supposing the number of sensors is N, the dimension of the individual is set to $2N$ and the individual encoding is expressed as $X_i = [x_{i1}^1, x_{i1}^2, x_{i2}^1, x_{i2}^2, \ldots, x_{iN,}^1 \ x_{iN}^2]$. The elements represent the x and y coordinates of sensors from 1 to N in turn.

The fitness function of the BP-QUATRE corresponds to the coverage rate of the network. Coverage rate is an

Table 1 Parameters settings

Algorithm	Parameters settings
BP-QUATRE	Fmax = 0.9, *F*min = 0.4
QUATRE variants	*F* = 0.7
PSO-IW	*W*max = 0.9, *W*min = 0.4, c1 = 2, c2 = 2
DE	*F* = 0.7, Cr = 0.1

Table 2 Performance for BP-QUATRE, QUATRE/target-to-best/1, and QUATRE/best/1

30D	QUATRE/target-to-best/1			QUATRE/best/1			BP-QUATRE		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
1	0.0000E+ 00	4.5475E-15	3.2155E-14	0.0000E+ 00	2.2737E-14	6.8905E-14	0.0000E+ 00	5.9117E-14	1.0075E-13
2	3.4918E+ 04	2.0307E+ 05	9.1401E+ 04	6.5779E+ 04	3.2526E+ 05	1.7428E+ 05	8.0342E+ 04	3.2499E+ 05	1.8932E+ 05
3	3.3344E-08	1.2152E+ 04	4.2407E+ 04	6.4313E-02	1.1505E+ 06	3.0040E+ 06	4.0302E-05	6.8721E+ 05	2.9931E+ 06
4	1.3712E+ 00	9.6559E+ 00	7.2335E+ 00	4.1346E+ 00	2.1215E+ 01	1.5656E+ 01	6.7090E+ 00	3.8516E+ 01	2.5066E+ 01
5	0.0000E+ 00	1.0914E-13	2.2504E-14	0.0000E+ 00	1.0914E-13	2.2504E-14	0.0000E+ 00	1.1369E-13	2.2968E-14
6	2.8398E-09	1.8153E+ 00	6.2792E+ 00	2.9877E-04	7.3092E+ 00	1.0870E+ 01	4.5328E-03	1.0260E+ 01	8.0606E+ 00
7	1.9057E-01	3.8445E+ 00	4.0489E+ 00	1.1364E+ 00	2.1349E+ 01	1.7591E+ 01	7.2738E-02	4.9636E+ 00	3.9591E+ 00
8	2.0749E+ 01	2.0933E+ 01	5.1788E-02	2.0849E+ 01	2.1004E+ 01	5.8828E-02	2.0827E+ 01	2.0953E+ 01	4.7373E-02
9	1.0130E+ 01	2.6120E+ 01	6.0410E+ 00	5.8076E+ 00	1.6079E+ 01	5.5873E+ 00	9.7462E+ 00	2.2635E+ 01	5.7088E+ 00
10	5.6843E-14	2.7047E-02	1.5235E-02	0.0000E+ 00	2.3154E-02	1.5496E-02	7.3960E-03	2.0791E-02	1.1579E-02
11	2.3082E+ 01	2.7925E+ 01	2.5361E+ 00	1.3929E+ 01	2.6411E+ 01	8.3326E+ 00	1.7053E-13	4.4972E+ 00	1.8006E+ 00
12	9.0630E+ 01	1.1724E+ 02	1.3886E+ 01	3.9359E+ 01	7.5733E+ 01	1.8919E+ 01	1.9899E+ 01	5.3234E+ 01	1.5281E+ 01
13	1.0132E+ 02	1.3263E+ 02	1.5255E+ 01	5.2197E+ 01	1.1252E+ 02	3.2964E+ 01	5.8213E+ 01	1.0634E+ 02	2.4064E+ 01
14	9.9796E+ 02	1.3914E+ 03	1.9533E+ 02	1.0081E+ 02	8.1082E+ 02	2.6067E+ 02	1.0878E+ 01	1.9290E+ 02	1.2680E+ 02
15	5.3901E+ 03	6.3005E+ 03	3.4954E+ 02	3.3512E+ 03	5.1268E+ 03	7.7891E+ 02	2.7376E+ 03	4.0668E+ 03	5.3740E+ 02
16	1.8510E+ 00	2.3565E+ 00	2.4224E-01	1.3247E+ 00	2.4246E+ 00	4.5226E-01	9.4808E-01	1.8071E+ 00	4.0821E-01
17	5.5094E+ 01	6.1017E+ 01	2.5301E+ 00	2.1370E+ 01	5.5165E+ 01	1.1812E+ 01	1.7609E+ 00	3.0571E+ 01	7.3126E+ 00
18	1.6116E+ 02	1.8968E+ 02	9.6887E+ 00	1.0816E+ 02	1.6029E+ 02	2.5169E+ 01	5.2189E+ 01	1.0584E+ 02	2.3166E+ 01
19	2.7404E+ 00	4.6348E+ 00	4.7267E-01	1.6863E+ 00	3.6567E+ 00	7.7921E-01	1.0161E+ 00	1.7223E+ 00	3.4349E-01
20	1.0903E+ 01	1.1871E+ 01	3.6616E-01	1.0342E+ 01	1.2062E+ 01	6.4629E-01	1.0035E+ 01	1.1254E+ 01	4.9221E-01
21	2.0000E+ 02	3.0384E+ 02	7.7553E+ 01	2.0000E+ 02	3.1932E+ 02	8.3201E+ 01	2.0000E+ 02	3.0271E+ 02	8.2769E+ 01
22	1.0740E+ 03	1.4937E+ 03	1.9286E+ 02	4.5570E+ 02	8.3787E+ 02	2.5055E+ 02	6.2475E+ 01	2.4699E+ 02	1.0913E+ 02
23	4.5981E+ 03	6.0880E+ 03	5.3171E+ 02	3.7200E+ 03	5.3213E+ 03	8.4120E+ 02	2.6820E+ 03	4.0320E+ 03	7.2841E+ 02
24	2.0017E+ 02	2.1733E+ 02	1.4867E+ 01	2.1116E+ 02	2.3769E+ 02	1.1610E+ 01	2.0020E+ 02	2.3468E+ 02	1.6071E+ 01
25	2.3653E+ 02	2.5235E+ 02	8.2988E+ 00	2.4178E+ 02	2.5758E+ 02	8.0331E+ 00	2.4144E+ 02	2.5694E+ 02	1.2474E+ 01
26	2.0001E+ 02	2.4606E+ 02	6.6115E+ 01	2.0001E+ 02	2.4558E+ 02	6.4221E+ 01	2.0001E+ 02	2.1684E+ 02	4.6111E+ 01
27	3.2346E+ 02	6.1723E+ 02	1.7389E+ 02	5.5559E+ 02	6.9637E+ 02	9.3585E+ 01	4.1966E+ 02	7.5219E+ 02	1.5375E+ 02
28	1.0000E+ 02	3.5818E+ 02	2.5128E+ 02	1.0000E+ 02	3.7948E+ 02	2.8873E+ 02	3.0000E+ 02	3.0000E+ 02	2.9792E-13
win	9	10	14	3	1	4	12	16	9
lose	15	17	13	21	26	23	13	12	19
draw	4	1	1	4	1	1	3	0	0

important aspect to measure the performance of WSN. Let each sensor can cover an area C_{i} and A is the total size of the region of interest. Then, the coverage rate CR is calculated by the Eq. 13.

$$\mathrm{CR} = \frac{\cup \mathrm{C_i}}{\mathrm{A}} \quad \mathrm{i} \in \mathrm{N} \tag{14}$$

However, it is too complicated to calculate the coverage rate of randomly deployed sensor networks by Eq. 13. Therefore, this paper uses the grid scanning method [37] to evaluate the coverage rate. According to [37], CR is evaluated as the Eq. 14.

$$\mathrm{CR} = \frac{\mathrm{m}}{\mathrm{n}} \tag{15}$$

5 Experimental results and discussion

A set of experiments was conducted to evaluate the performance of the proposed algorithm BP-QUATRE and its application to dynamic deployment in WSN.

5.1 Experimental results for BP-QUATRE

In this subsection, we evaluate the performance of the proposed BP-QUATRE algorithm on CEC2013 [40] test

Table 3 Performance for BP-QUATRE, PSO-IW, and DE algorithms

30D	PSO-IW			DE/best/1/bin			BP-QUATRE		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
1	1.8516E-02	2.0880E+ 03	2.3796E+ 03	2.2737E-13	2.2737E-13	0.0000E+ 00	0.0000E+ 00	5.9117E-14	1.0075E-13
2	9.2004E+ 05	2.9077E+ 07	4.8633E+ 07	1.4886E+ 07	2.6850E+ 07	7.1628E+ 06	8.0342E+ 04	3.2499E+ 05	1.8932E+ 05
3	7.5454E+ 08	9.3345E+ 10	1.8679E+ 11	1.8850E+ 08	6.7124E+ 08	2.5438E+ 08	4.0302E-05	6.8721E+ 05	2.9931E+ 06
4	1.4853E+ 03	5.8973E+ 03	5.1338E+ 03	2.3767E+ 04	3.7255E+ 04	6.3675E+ 03	6.7090E+ 00	3.8516E+ 01	2.5066E+ 01
5	8.1621E-01	1.2112E+ 03	1.2376E+ 03	1.1369E-13	1.3642E-13	4.5936E-14	0.0000E+ 00	1.1369E-13	2.2968E-14
6	3.8139E+ 01	2.4618E+ 02	2.6320E+ 02	1.6157E+ 01	2.4642E+ 01	1.1567E+ 01	4.5328E-03	1.0260E+ 01	8.0606E+ 00
7	7.4531E+ 01	2.3039E+ 02	1.4010E+ 02	4.5255E+ 01	5.8459E+ 01	7.7884E+ 00	7.2738E-02	4.9636E+ 00	3.9591E+ 00
8	2.0758E+ 01	2.0918E+ 01	5.8419E-02	2.0778E+ 01	2.0945E+ 01	4.7712E-02	2.0827E+ 01	2.0953E+ 01	4.7373E-02
9	2.2437E+ 01	2.9920E+ 01	3.3769E+ 00	2.3097E+ 01	2.9119E+ 01	1.8981E+ 00	9.7462E+ 00	2.2635E+ 01	5.7088E+ 00
10	1.4411E+ 00	5.7765E+ 02	4.3368E+ 02	6.7846E+ 00	1.5520E+ 01	4.3221E+ 00	7.3960E-03	2.0791E-02	1.1579E-02
11	7.2539E+ 01	1.6044E+ 02	3.8346E+ 01	5.6843E-14	6.9647E-01	1.0101E+ 00	1.7053E-13	4.4972E+ 00	1.8006E+ 00
12	9.8398E+ 01	1.8126E+ 02	6.7573E+ 01	1.1538E+ 02	1.5072E+ 02	1.5277E+ 01	1.9899E+ 01	5.3234E+ 01	1.5281E+ 01
13	1.5549E+ 02	2.4254E+ 02	4.7152E+ 01	1.2579E+ 02	1.6918E+ 02	1.3208E+ 01	5.8213E+ 01	1.0634E+ 02	2.4064E+ 01
14	2.1910E+ 03	3.6743E+ 03	8.1801E+ 02	1.3049E+ 00	2.6590E+ 01	5.0098E+ 01	1.0878E+ 01	1.9290E+ 02	1.2680E+ 02
15	3.0712E+ 03	4.6861E+ 03	8.3189E+ 02	5.1246E+ 03	6.1285E+ 03	3.9430E+ 02	2.7376E+ 03	4.0668E+ 03	5.3740E+ 02
16	5.2554E-01	1.2755E+ 00	3.6444E-01	1.5328E+ 00	2.3124E+ 00	3.2555E-01	9.4808E-01	1.8071E+ 00	4.0821E-01
17	1.1866E+ 02	1.7987E+ 02	3.7182E+ 01	2.6033E+ 01	3.1227E+ 01	9.1319E-01	1.7609E+ 00	3.0571E+ 01	7.3126E+ 00
18	1.1506E+ 02	1.6382E+ 02	2.8799E+ 01	1.9757E+ 02	2.2046E+ 02	1.1380E+ 01	5.2189E+ 01	1.0584E+ 02	2.3166E+ 01
19	6.9242E+ 00	2.8457E+ 03	6.4680E+ 03	2.7883E+ 00	3.8670E+ 00	3.9209E-01	1.0161E+ 00	1.7223E+ 00	3.4349E-01
20	1.0638E+ 01	1.2595E+ 01	7.4314E-01	1.2114E+ 01	1.2813E+ 01	2.5585E-01	1.0035E+ 01	1.1254E+ 01	4.9221E-01
21	2.0438E+ 02	3.9378E+ 02	1.9091E+ 02	2.0000E+ 02	2.9010E+ 02	7.6842E+ 01	2.0000E+ 02	3.0271E+ 02	8.2769E+ 01
22	2.4780E+ 03	3.8977E+ 03	7.9507E+ 02	1.1651E+ 02	2.5046E+ 02	1.8276E+ 02	6.2475E+ 01	2.4699E+ 02	1.0913E+ 02
23	2.9470E+ 03	5.0143E+ 03	9.3015E+ 02	5.5175E+ 03	6.5088E+ 03	4.1567E+ 02	2.6820E+ 03	4.0320E+ 03	7.2841E+ 02
24	2.8274E+ 02	2.9936E+ 02	1.0622E+ 01	2.5730E+ 02	2.7222E+ 02	6.3487E+ 00	2.0020E+ 02	2.3468E+ 02	1.6071E+ 01
25	2.8413E+ 02	3.0698E+ 02	9.4788E+ 00	2.7937E+ 02	2.8866E+ 02	4.6272E+ 00	2.4144E+ 02	2.5694E+ 02	1.2474E+ 01
26	2.0007E+ 02	3.6095E+ 02	5.3876E+ 01	2.0112E+ 02	2.0202E+ 02	5.0229E-01	2.0001E+ 02	2.1684E+ 02	4.6111E+ 01
27	1.0119E+ 03	1.1676E+ 03	7.9394E+ 01	9.4887E+ 02	1.0521E+ 03	4.0959E+ 01	4.1966E+ 02	7.5219E+ 02	1.5375E+ 02
28	1.1640E+ 02	1.9266E+ 03	7.1657E+ 02	3.0000E+ 02	3.2379E+ 02	1.6825E+ 02	3.0000E+ 02	3.0000E+ 02	2.9792E-13
win	3	2	0	2	4	17	22	22	11
lose	25	26	28	25	24	11	5	6	17
draw	0	0	0	1	0	0	1	0	0

suite for real-parameter optimization, which includes unimodal functions (f1-f5), multimodal functions (f6-f20), and composition functions (f21-f28). The names and search ranges of this 28 benchmark functions can be found in [40], and they are shifted to the same global best location $O\{o_1, o_2, \dots, o_d\}$.

Firstly, we compare the BP-QUATRE with the two QUATRE variants “QUATRE/target-to-best/1” and “QUATRE/best/1” as BP-QUATRE employs these two mutation strategies. Then, we compare the BP-QUATRE with inertia weight PSO and standard DE due to the relationship among them as described in ref [5]. The parameter settings of the algorithms are shown in Table 1. The dimensions of all functions are set to 30. The population size ps is set to 100 for each algorithm, and the maximal number of function evaluation (NFE) is 3,000,000. We run each algorithm on each benchmark function 50 times independently. The best, mean, and standard deviation of the function error are collected in Table 2 and Table 3. The simulation results of some benchmark functions are shown in Fig. 2.

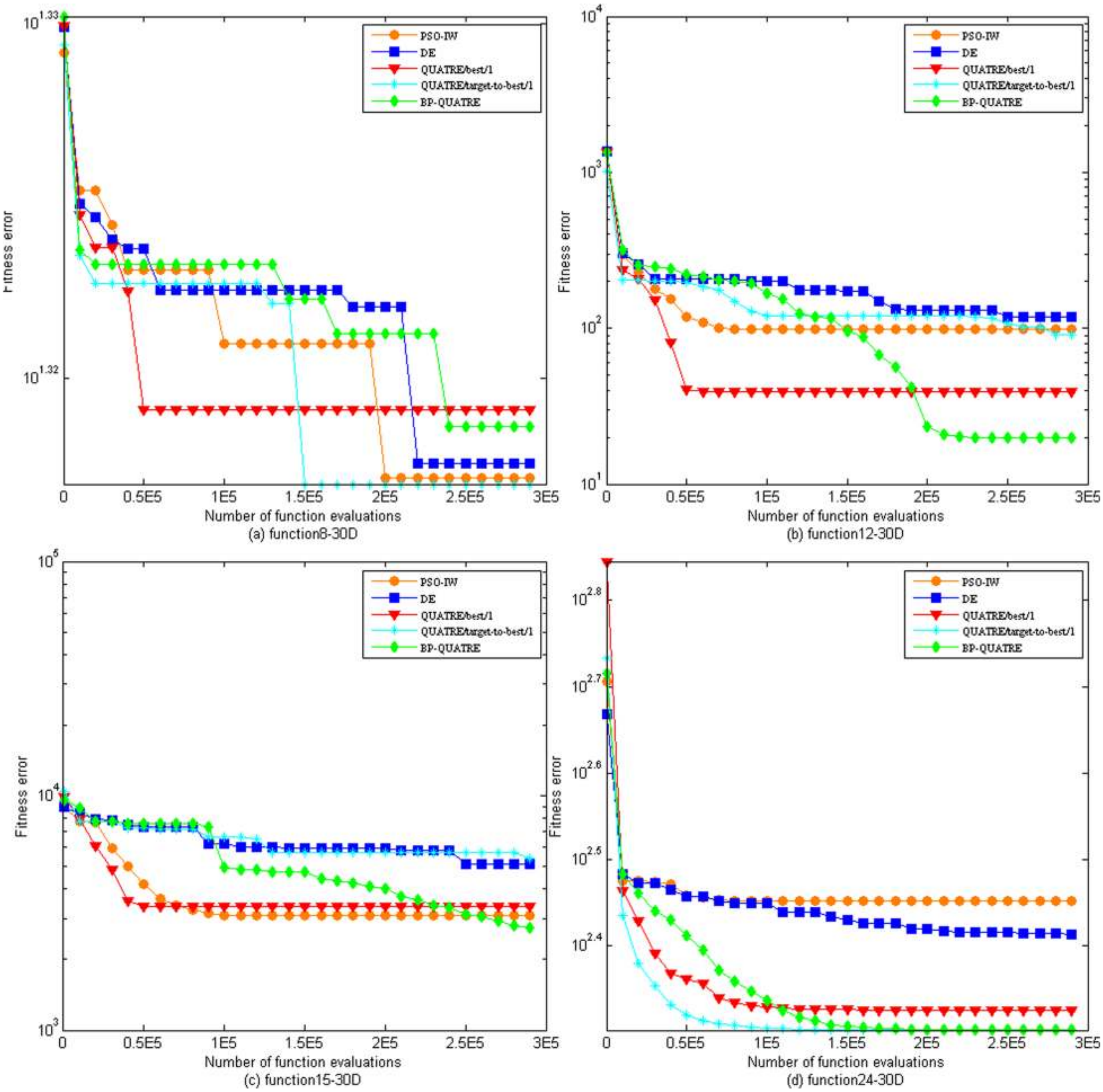

Fig. 2 Fitness errors vs. number of function evaluations of functions f8, f12, f15, and f24. The figure presents the fitness error and the convergence speed comparison by employing the best value of 50 runs obtained by each competing algorithm on 30-D optimization. The functions f8, f12, f15, and f24 figures are presented here. NFE means number of function evaluations

From Table 2, we can see that BP-QUATRE has significantly better performance than the other two QUATRE variants over 28 benchmark functions. The BP-QUATRE finds 12 best values and 16 mean values of CEC2013 benchmark functions in comparison QUATRE variants. This is because the BP-QUATRE can take advantage of different mutation strategies to maintain population diversity, and its linear decrease scale factor control strategy is helpful to balance exploration and exploitation ability. For the standard deviation, the “QUATRE/target-to-best/1” has better performance than “QUATRE/best/1” and BP-QUATRE algorithms, and BP-QUATRE algorithm has better performance than “QUATRE/best/1.” In addition, we can observe that QUATRE variants with different mutation strategies have different performance. The “QUATRE/target-to-best/1” performs

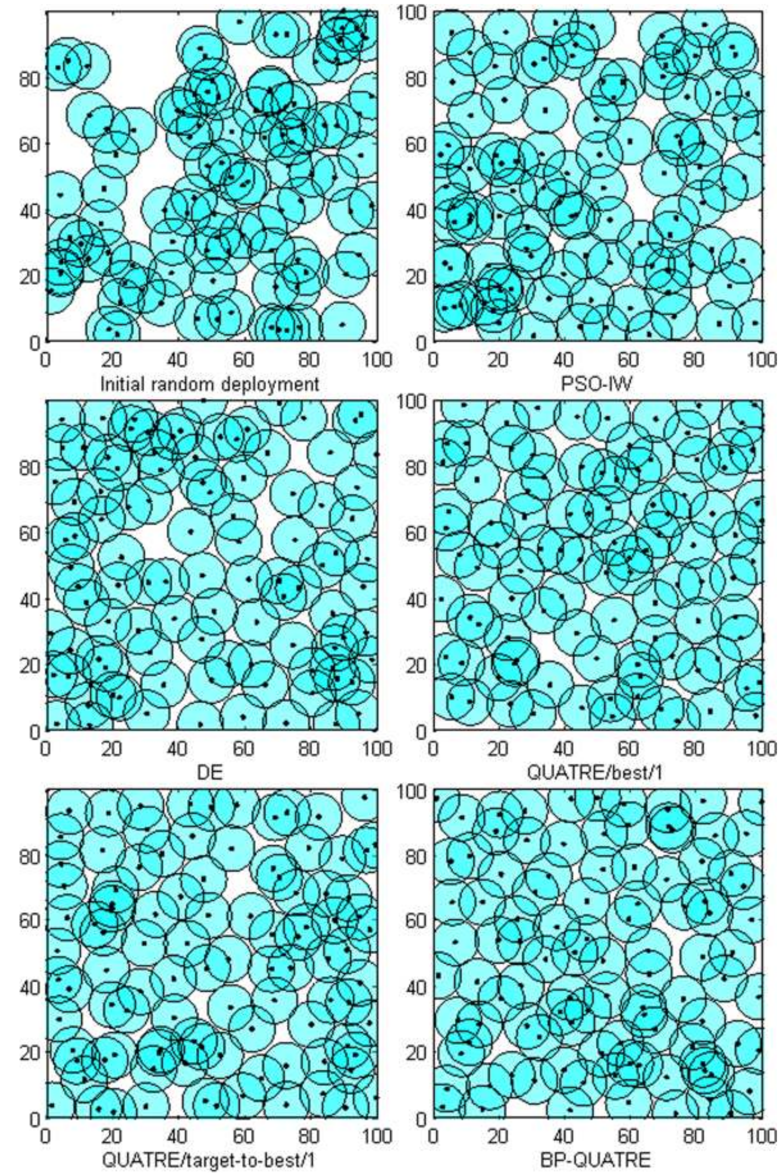

Fig. 3 Initial random deployment and final sensor positions based on each algorithm. The figure presents Initial random deployment and the final deployment of 100 sensor nodes in the monitored target area based on each competing algorithm (i.e., PSO-IW, DE, QUATRE/best/1, QUATRE/target-to-best/1, and BP-QUATRE). The black dots represent the final positions of the sensor nodes, and the circles represent the relative monitoring areas of the sensor nodes. The coverage rates obtained by PSO-IW, DE, QUATRE/best/1, QUATRE/target-to-best/1, and BP-QUATRE are 88.73%, 91.83%, 93.67%, 93.05%, and 93.89%, respectively

better on unimodal and composition functions than "QUATRE/best/1,", while the "QUATRE/best/1" performs better on multimodal functions than "QUATRE/target-to-best/1."

From Table 3, we can see that, for the best value, the PSO-IW algorithm finds 3 minimum values of 28 benchmark functions. The DE algorithm finds 2 minimum values of 28 benchmark functions. While our proposed BP-QUATRE algorithm finds 22 minimum values of 28 benchmark functions in comparison with PSO-IW and DE algorithms, and thus, it has overall better performance than the contrasted algorithms. For the mean, our proposed algorithm also has significantly better performance than the competing algorithms. For the standard deviation, the DE algorithm has better performance than PSO-IW and BP-QUATRE algorithms, and BP-QUATRE algorithm has better performance than PSO-IW algorithm. Overall, our proposed

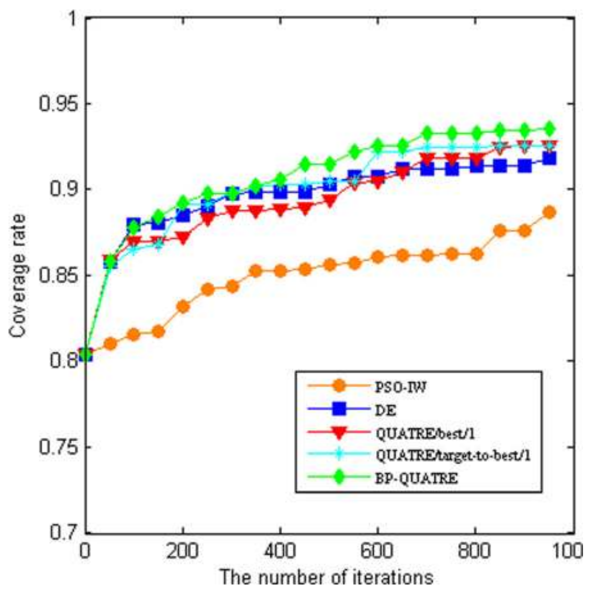

Fig. 4 Comparison of coverage rates of the proposed algorithm with the other algorithms, e.g., PSO-IW, DE, QUATRE/best/1 and QUATRE/targetto-best/1

BP-QUATRE algorithm has better performance than the other two competing algorithms.

5.2 Simulation results for dynamic deployment in WSN

Simulations are conducted to evaluate the performance of BP-QUATRE algorithm in the dynamic deployment of WSN. The simulation results of the proposed algorithm are compared with the results of the PSO-IW, DE, and two QUATRE variants.

To make the simulation results more reliable, the parameter settings such as the target area, the number of sensors, and their detection radius are according to [34]. The monitored target area is a square region with a size of 100 m × 100 m, and 100 sensor nodes are scattered randomly on this target region. The parameter settings for the probabilistic detection model are $\alpha_1 = 1$, $\alpha_2 = 0$, $\beta_1 = 1$, and $\beta_2 = 1.5$. And the detection radius of each sensor node is 7 m, the uncertainty parameter of measurement r_e is 3.5 m, and the communication radius r_c is 21 m. The effective detection threshold c_{th} is 0.7. The control parameters of each algorithm are the same as in Section 5.1 except that the acceleration coefficients c1 and c2 of the PSO are set to 1. The population size ps is set to 40, and the number of iterations is 1000. We run each algorithm 10 times independently with the same initialization.

One of initial deployments and the final best deployments of WSN after executing all competing algorithms based on the probabilistic detection model are shown in Fig. 3. The best convergences of each algorithm are shown in Fig. 4 by coverage rate for the number of iterations. The best, mean, and standard deviation of the coverage rates for the mentioned algorithms are given in Table 4. Obviously, it can be seen that our proposed BP-QUATRE has better performance than other two QUATRE variants and all QUATRE algorithms have better performance than PSO-IW and DE algorithm. In other words, BP-QUATRE has better coverage rate than the other four competing algorithms in the dynamic deployment of WSN. This is certainly related to the more powerful exploration and exploitation capability of the BP-QUATRE algorithm.

6 Conclusion

This paper proposes a novel BP-QUATRE algorithm for optimization problems. In BP-QUATRE, the population is divided into two subpopulations with sort strategy, and each subpopulation employs a different mutation strategy to balance between the diversity and convergence rate. In addition, adjusting scale factor with linear decrease strategy is adopted in BP-QUATRE algorithm to balance between exploration and exploitation ability. The proposed algorithm is verified under CEC2013 test suite. The experimental results demonstrate that the proposed BP-QUATRE algorithm not only has better performance than QUATRE variants “QUATRE/target-to-best/1” and “QUATRE/best/1,” but also has better performance than the PSO-IW algorithm and DE algorithm. We also apply the proposed BP-QUATRE algorithm to dynamic deployment in WSN. The simulation results demonstrate that the proposed BP-QUATRE algorithm has better coverage rate than the other competing algorithms. In the future work, we will apply BL-QUATRE algorithm to classify music genre [41].

Table 4 Performance for competing algorithms in the dynamic deployment of WSN

	PSO-IW	DE	QUATRE/best/1	QUATRE/target-to-best/1	BP-QUATRE
Best	0.8873	0.9183	0.9367	0.9305	0.9389
Mean	0.8629	0.9143	0.9212	0.9187	0.9362
Worse	0.8443	0.9084	0.9149	0.9123	0.9331
Std	0.0117	0.0030	0.0067	0.0053	0.0021

Abbreviations

BP-QUATRE: Bi-population QUATRE; DE: Differential evolution; PSO: Particle swarm optimization; PSO-IW: Inertia weight PSO; QUATRE: QUasi-Affine TRansformation Evolution algorithm; Std: Standard deviation; WSN: Wireless sensor networks

Acknowledgements

The authors would like to thank Prof. Zhenyu Meng for providing the code of QUATRE.

Authors' contributions

NL proposed the idea of the BP-QUATRE algorithm. He also carried out the simulations and drafted the paper. JP supervised the work and introduced the idea of applying the proposed BP-QUATRE algorithm into dynamic deployment in WSN. TN gave some suggestions for the paper and revised the manuscript. All authors read and approved the final manuscript.

Authors' information

Nengxian Liu is a Ph.D. candidate in the College of Mathematics and Computer Science, Fuzhou University. His research interest includes computing intelligence and sensor networks.
Jeng-Shyang Pan received the Ph.D. degree in Electrical Engineering from the University of Edinburgh, UK, in 1996. Now, he is the Dean in the College of Information Science and Engineering, Fujian University of Technology. His current research interests include soft computing, sensor networks and signal processing.
Trong-The Nguyen received his Ph.D. degree in Communication Engineering from the National Kaohsiung University of Applied Sciences, Taiwan, in 2016. He is currently a lecturer in the College of Information Science and Engineering, Fujian University of Technology. His current research interests include computational intelligence and sensor networks.

Funding

No funding.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that there are no competing interests.

Author details

[1]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China. [2]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China. [3]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China. [4]Department of Information Technology, University of Manage and Technology, Haiphong 180000, Vietnam.

Received: 18 February 2019 Accepted: 29 May 2019
Published online: 05 July 2019

References

1. R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
2. J. Kennedy, R. Eberhart, *Particle Swarm Optimization* (Proceedings of IEEE International Conference on Neural Networks, Perth, 1995)
3. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B Cybern. **26**(1), 29–41 (1996)
4. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
5. Z.Y. Meng, J.S. Pan, X. HR, QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization. Knowl.-Based Syst. **109**, 104–121 (2016)
6. Y.H. Shi, R. Eberhart, *A Modified Particle Swarm Optimizer* (Proceedings of the IEEE Conference on Evolutionary Computation, Anchorage, 1998)
7. B. Liu, L. Wang, et al., Improved particle swarm optimization combined with chaos. Chaos Solison. Fract. **25**(5), 1261–1271 (2005)
8. R. Cheng, Y.C. Jin, A social learning particle swarm optimization algorithm for scalable optimization. Inf. Sci. **291**, 43–60 (2015)
9. J.H. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm. Soft. Comput. **9**(6), 448–462 (2005)
10. J. Brest, S. Greiner, et al., Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. **10**(6), 646–657 (2006)
11. S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution. IEEE Trans. Evol. Comput. **12**(1), 64–79 (2008)
12. Z.Y. Meng, J.S. Pan, *QUasi-affine TRansformation Evolutionary (QUATRE) algorithm: A parameter-reduced differential evolution algorithm for optimization problems* (IEEE Congress on Evolutionary Computation (CEC), Vancouver, 2016)
13. Z.Y. Meng, J.S. Pan, *QUasi-Affine TRansformation Evolutionary (QUATRE) Algorithm: The Framework Analysis for Global Optimization and Application in Hand Gesture Segmentation* (2016 IEEE 13th International Conference on Signal Processing, Chengdu, 2016)
14. Z.Y. Meng, J.S. Pan, Monkey king evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. Knowl.-Based Syst. **97**, 144–157 (2016)
15. Z.Y. Meng, J.S. Pan, X.Q. Li, *The QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: An Overview* (The Euro-China Conference on Intelligent Data Analysis and Applications, Málaga, 2017)
16. J.S. Pan, Z.Y. Meng, et al., *QUATRE algorithm with sort strategy for global optimization in comparison with DE and PSO variants* (The Euro-China Conference on Intelligent Data Analysis and Applications, Málaga, 2017)
17. Z.Y. Meng, J.S. Pan, QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): An enhanced structure for differential evolution. Knowl.-Based Syst. **155**, 35–53 (2018)
18. I. Loshchilov, M. Schoenauer, M. Sebag, *Bi-Population CMA-ES Agorithms with Surrogate Models and Line Searches* (Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, Amsterdam, 2013), pp. 1177–1184
19. J.F. Chang, S.C. Chu, et al., A parallel particle swarm optimization algorithm with communication strategies. J. Inf. Sci. Eng. **21**(4), 809–818 (2005)
20. Y. WJ, J. Zhang, *Multi-population differential evolution with adaptive parameter control for global optimization* (Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference, Dublin, 2011)
21. L.Z. Cui, G.H. Li, et al., Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. Comput. Oper. Res. **67**, 155–173 (2016)
22. P.W. Tsai, J.S. Pan, *Parallel Cat Swarm Optimization* (Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 2008)
23. S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art. IEEE Trans.Evol. Comput. **15**(1), 4–31 (2011)
24. S. Das, A. Konar, et al., *Two improved differential evolution schemes for faster global search* (Genetic and Evolutionary Computation Conference, Washington DC, 2005)
25. J.S. Pan, L.P. Kong, T.W. Sung, et al., α-Fraction first strategy for hierarchical model in wireless sensor networks. J Internet Technol. **19**(6), 1717–1726 (2018)
26. S. Abdollahzadeh, N.J. Navimipour, Deployment strategies in the wireless sensor network: A comprehensive review. Comput.Commun., **91**, 1–16 (2016)
27. J.S. Pan, L.P. Kong, T.W. Sung, et al., A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set. J Internet Technol. **19**(6), 1111–1118 (2018)
28. H. YFa, C.H. Hsu, Energy efficiency of dynamically distributed clustering routing for naturally scattering wireless sensor networks. J Netw Intell **3**(1), 50–57 (2018)
29. X.C. Bao, C.Z. Deng, et al., Topology optimization of fault tolerant target monitoring for energy harvesting wireless sensor network. J Netw Intell **2**(4), 310–321 (2017)
30. Y. Zou, K. Chakrabarty, *Sensor Deployment and Target Localization Based on Virtual Forces* (Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, 2003)
31. G.L. Wang, G.H. Cao, Movement-assisted sensor deployment. IEEE T Mobile Comput. **5**(6), 640–652 (2006)
32. X.L. Wu, L. Shu, et al., *Swarm Based Sensor Deployment Optimization In Ad-Hoc Sensor Networks* (Proceedings of the 2nd International Conference on Embedded Software and Systems, Xi'an, 2005)
33. M. Rout, R. Roy, Optimal wireless sensor network information coverage using particle swarm optimization method. Int. J. Electron. Lett. **5**(4), 491–499 (2017)

34. X.Y. Yu, J.X. Zhang, et al., A faster convergence artificial bee colony algorithm in sensor deployment for wireless sensor networks. Int. J. Distrib. Sens. N **1-9**, 2013 (2013)
35. L.Z. Jin, J. Jia, et al., *Node Distribution Optimization in Mobile Sensor Network Based on Multi-Objective Differential Evolution Algorithm* (2010 Fourth International Conference on Genetic and Evolutionary Computing, Shenzhen, 2010)
36. Y. Zou, K. Chakrabarty, *Sensor Deployment and Target Localization Based on Virtual Forces* (IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, 2003)
37. K. Chakrabarty, S.S. Iyengar, et al., Grid coverage for surveillance and target location in distributed sensor networks. IEEE. T. Comput. **51**(12), 1448–1453 (2002)
38. S.J. Li, C.F. Xu, et al., *Sensor Deployment Optimization for Detecting Maneuvering Targets* (Proceedings of the 7th international conference on information fusion, Philadelphia, 2005)
39. S.M.A. Salehizadeh, A. Dirafzoon, et al., *Coverage in Wireless Sensor Networks Based on Individual Particle Optimization* (Proceedings of the IEEE International Conference on Networking, Sensing and Control, Chicago, 2010)
40. J. Liang et al., *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization* (Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical report 2012, 2013)
41. Y.T. Chen, C.H. Chen, et al., A two-step approach for classifying music genre on the strength of AHP weighted musical features. Math **7**(1), 19 (2019)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.