Hindawi
Journal of Sensors
Volume 2023, Article ID 2306189, 11 pages
https://doi.org/10.1155/2023/2306189

Research Article

Wireless Sensor Deployment Based on Multiobjective Adaptive Fish Migration Optimization

Lin-Dong Xu,[1] Wei-Min Zheng ,[1] Qing-Wei Chai ,[1] Li-Li Xu ,[2] and Zhuang Geng[2]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*Affiliated Hospital of Qingdao University, Qingdao 266590, China*

Correspondence should be addressed to Li-Li Xu; qdfyxll@qdu.edu.cn

Received 16 March 2023; Revised 8 June 2023; Accepted 16 June 2023; Published 3 July 2023

Academic Editor: Sunil Kumar Singh

Wireless Sensor Network (WSN) is a powerful tool to help humans monitor a specific area, and the deployment strategy of sensors profoundly determines the performance of WSN. How to find the best deployment method has become the research topic for many scholars. The deployment strategy aims to expand the deployment scope, reduce energy consumption, and reduce duplicate coverage areas. Many multiobjective heuristic algorithms have been proposed to solve this problem. This paper proposes a multiobjective adaptive fish migration optimization (MAFMO) algorithm, which adds an adaptive-based repository and crowding degree-selection strategy for multiobjective optimization. The simulation results reveal that the MAFMO algorithm has more advantages in malleability and distribution than other famous algorithms. Finally, the algorithm is applied to the WSN deployment problem, and the simulation results are compared with other algorithms. The results show that a better solution can be found using MAFMO.

1. Introduction

In recent years, wireless sensor network (WSN) have been fully developed. As sensors get smaller, cheaper, and more intelligent, WSNs become more widespread [1].

With the development of information technology in modern society, the deployment requirements of WSNs in various fields are getting higher and higher [2]. In the field of detection, people deploy a certain number of wireless sensors to detect various parameters in the target area, such as air humidity, temperature, and pollutant content. WSN can be well used for forest protection and timely prevention of fires. With the development of smart homes [3], more and more homes and offices deploy WSN to control indoor temperature, light, humidity, and flow of people [4]. In addition, in some industrial production scenarios, WSN can be deployed to detect essential parameters in the production process. Enabling the person in charge of industrial production to obtain the relevant data efficiently and accurately, improves the management level of industrial production, and significantly reduces the production risks. In the positioning field, several wireless sensors are deployed to achieve full coverage of the monitoring range, and the sensors are used to locate something in the area [5]. For example, an alarm will be triggered when people exceed the detection area in the real-time positioning of older people and children. Furthermore, wireless sensors can also be installed in moving objects to determine the trajectory of movement, such as cars. WSN plays an important role in many fields. With the development of WSN, it also constantly faces new challenges and opportunities [6].

It can be seen that many important areas need to deploy a certain number of wireless sensors to control the area. Since wireless sensors are independent, they cannot be charged once deployed [7]. Therefore, when deploying wireless sensors, the energy consumption of all wireless sensors needs to be considered. At the same time, the repeated coverage of a subarea will also cause resource consumption. To satisfy the coverage requirements, the coverage of the wireless sensor to the target area needs to be as large as possible. Therefore, it is necessary to consider the coverage, energy consumption, and overcoverage areas when deploying wireless sensors in the target space.

The use of multiobjective optimization algorithms can generate multiple optimal solutions, which can provide more choices and possibilities, allowing decision makers to make better decisions. Furthermore, these solutions can provide better robustness for decision makers. In addition, some researchers have also achieved many research results on wireless sensor deployment problems through multiobjective optimization algorithms. Jia et al. [8] proposed multiobjective optimization for coverage control of WSN with adjustable sensing radius, and Benatia et al. [9] proposed multiobjective WSN deployment using genetic algorithm under connection constraints. In addition, there are also some improved solutions for single-objective optimization algorithms for reference. Such as, an improved cuckoo search algorithm based on compact and parallel techniques [10], a novel selection optimization differential evolution (DE) algorithm [11], a multigroup discrete symbiotic organisms search (SOS) algorithm [12], an information feedback models [13], and so on.

In this paper, a new multi-objective adaptive fish migration optimization (MAFMO) algorithm is introduced, which seeks the optimal solution in the multidimensional object space through the growth and migration of fish schools. We propose the three important indicators related to WSN deployment: coverage rate, energy consumption, and overcoverage rate. The above three indicators are considered because these three objectives have a great effect on the efficient deployment of wireless sensors.

The rest of the paper is organized as follows: related work is given in Section 2. In Section 3, we introduce the MAFMO. The modeling of the important indicators of WSN is given in Section 4. Experiment results of different multiobjecive heuristic algorithms in many test functions are shown in Section 5. Finally, the conclusions of this paper are given in Section 6.

2. Related Work

2.1. Multiobjective Heuristic Algorithm. Over the past period, the heuristic algorithm has attracted more and more attention of scholars, especially in multiobjective heuristic algorithms. The multiobjective optimization algorithms are of crucial importance in real life [14]. Every decision in life will consider many factors, and the same is true for multiobjective optimization. Due to the lack of suitable techniques, the multiobjective problems were converted into single-objective problems for solving. Therefore, the solution does not balance optimal solutions for individual objectives well, and can only generate optimal solutions for single-objective problems. When solving multiobjective problems, the candidate solution does not contain only one optimal solution, but also a solution set with multiple optimal solutions. In the multiobjection heuristic algorithm, the candidate solution is named as Pareto optimal solution, and these solutions constitute the Pareto Front in solution space, as shown in Figure 1.

Coello and Lechuga [15] proposed the multiobjective particle swarm optimization (MOPSO) algorithm in 2002, which is a multiobjective based on the particle swarm optimization (PSO) [16], where PSO is a heuristic algorithm inspired by the bird's behavior. In recent years, researches

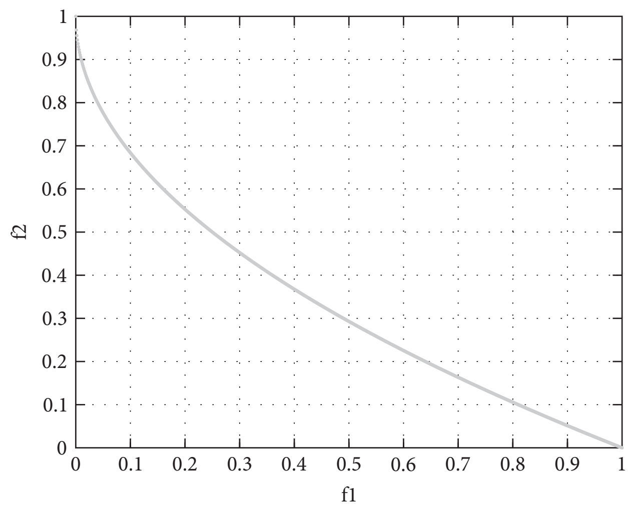

Figure 1: Pareto front.

based on the PSO algorithm have emerged in an endless stream, such as compact adaptive particle swarm optimization algorithm [17], dynamic transfer function [18], and so on. MOPSO uses external repositories and a location-based approach to maintain diversity. Repositories are used to store nondominated solutions, and solutions remain malleable by using a location-based approach. The so-called location-based method divides the target space of particles into multiple hypercubes and adds them to the repository according to the number of particles in the hypercube, making the solution distribution in the repository more even.

Zhang and Li [19] proposed a multiobjective evolutionary algorithm based on decomposition (MOEA/D) in 2007, which decomposes a multiobjective optimization problem into multiple scalar optimization subproblems and optimizes them simultaneously. The three main decomposition methods are proposed, the weighted sum method, the Chebyshev decomposition method and the boundary intersection method. Each subproblem is optimized by using only a few adjacent subproblems, and each subproblem in the adjacent region is updated accordingly.

Deb et al. [20] proposed a fast and elitist multiobjective genetic (NSGA-II) algorithm in 2002. In this algorithm, the concept of nondominated solution sorting is proposed. The goal is to have a nondominated ranking of all individuals. The higher the degree, the less likely it is to be dominated. The algorithm proposes an elite strategy to prevent the loss of excellent individuals in the evolution process. The specific method of this elite strategy is to continuously merge the parent population and the offspring population, and then select the best individual. Furthermore, a crowding degree-selection strategy is proposed to overcome the shortcomings of artificially specifying the shared parameters. When selecting a solution, select the solution from high to low according to the degree of nondomination of the solution. If the two solutions are the same, then select the relatively uncrowded solution. Therefore, the algorithm performs very well in maintaining the diversity of the solution set. In 2014, Deb and Jain [21] proposed an evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach (NSGA-III). The algorithm is similar to the framework used by NSGA-II. The method of reference point association is used instead of the solution of the larger crowding distance value in NSGA-II. NSGA-III was applied to the problems with three or more objectives.

Mirjalili et al. [22] proposed a multiobjective gray wolf optimization (MOGWO) algorithm in 2016. The algorithm mainly simulates the hunting behavior of gray wolves in nature and produces four leading gray wolves, including three steps hunting, surrounding prey, and attacking prey. In the multiobjective algorithm, the wolf moves its position through the positional relationship with the leader, and the leader will also continuously update his position. The loop searches for a nondominated solution in the target space by continuously updating the wolf pack positions [23].

Liu et al. [24] proposed a decomposition-based MOEAD algorithm. The algorithm adapts the topology of the reference vector and the scalar function to enhance the search ability of the algorithm. Addressed a noticeable performance degradation of the original algorithm in problems solving irregular Pareto fronts. For large-scale multiobjective optimization problems, He et al. [25] proposed an adaptive offspring generation method. This method enables multiobjective algorithms to generate excellent candidate solutions in high-dimensional spaces effectively. For large-scale multiobjective optimization, Yang et al. [26] focused on decision variables. In 2021, a fuzzy decision variable framework for large-scale multiobjective optimization was proposed, and the evolution process was divided into fuzzy evolution and precise evolution. In addition, to solve the problem of too many decision variables and poor versatility in the large-scale multiobjective optimization problems, Li et al. [27] proposed a large-scale MOEA framework based on reference-guided offspring generation in 2022. Guiding the sampling of promising solutions during offspring generation is by constructing multiple reference vectors in the decision space.

2.2. Adaptive Fish Migration Optimization. The adaptive fish migration optimization (AFMO) was proposed by Chai and Zheng [28] in 2020. Compared with the original FMO algorithm, this algorithm introduces the weight value that changes with the number of iterations to enhance the local search ability. In AFMO, the fish school seeks the optimal solution by exploring the area around the optimal individual, and continuously updates the optimal individual, which enhances the fish school's search ability in the local range. When the fish moves, there will be better or worse situations. If the fitness value becomes better, then move will be in the direction closer to the current individual. If the fitness value becomes worse, then move will be in the opposite direction.

$$V_i^{t+1} = w \cdot \left(p_{\text{best}}^t - p_i^t\right) \cdot \frac{e_i^t}{E_{\max}} + \frac{fv_i - fv_r}{|fv_i - fv_r|} \cdot c \cdot rand \cdot \left(p_r^t - p_i^t\right), \tag{1}$$

$$p_i^{t+1} = p_i^t + V_i^{t+1}. \tag{2}$$

The Equation (1) shows the velocity update formula of the fish school in AFMO, and the Equation (2) shows the position update formula of the fish school, where w is the weight factor, which decreases from 2 to 0.4 with the increase of the number of iterations, and p_{best}^t is the optimal individual, e_i^t represents the energy of the individual. The greater the energy, the farther the distance traveled. $E_{\max}$ represents the maximum energy, which is used to measure whether the individual enters the next growth stage. fv_i and fv_r represent the fitness value of the i-th individual and the fitness value of the randomly selected individual in t iterations. c is a constant value and *rand* is a random number from 0 to 1.

When an individual does not obtain a better fitness value after moving to a new position, then the individual energy needs to be updated. The energy update formula is shown in the Equation (3):

$$e_i^{t+1} = e_i^t + R_1 \cdot E_{\max} \cdot \frac{\mathrm{fv_i} - \mathrm{fv_{best}}}{\mathrm{fv_{max}} - \mathrm{fv_{best}}}, \tag{3}$$

$$g = \begin{cases} [(g+1) \bmod 5]+, & \text{if eng} > 2 \times \mathrm{E_{max}} \\ g, & \text{otherwise.} \end{cases} \tag{4}$$

The R_1 is a random number between 2 and 12, and $fv_{\max}$ is the maximum fitness value. Equation (4) represents the formula for updating the growth stage of the fish. When the energy consumption of the fish reaches a certain amount, it will grow to the next stage. The growth stage of the fish is divided into four stages. The probability of initialization is also increasing.

2.3. WSN Deployment. Unlike traditional networks, WSN is set up for specific applications, including object tracking, environmental detection, industrial detection, and 3D terrain localization [28]. WSN is also limited by its properties, such as battery capacity, communication range, communication protocol [1], and so forth. Different network models will have different impacts on the lifetime of WSN because different communication protocols communicate in different ways, such as channel monitoring, collision mechanism, and idle listening [29]. Figure 2 shows the deployment of sensors in the monitoring area.

In the resource-constrained WSN, ensuring energy maximization and collection of information are the essential requirements to achieve WSN deployment [30]. In recent years, there has been more and more research on WSN reliability, among which there are two typical technologies: retransmission and redundancy [31]. Most of the current research is based on retransmission, and the research on redundancy technology is relatively less. When WSN is based on retransmission technology, lost data will be recovered through retransmission, but it will also cause network congestion and secondary energy loss. In addition, when WSN is based on redundancy technology, it will use some coding

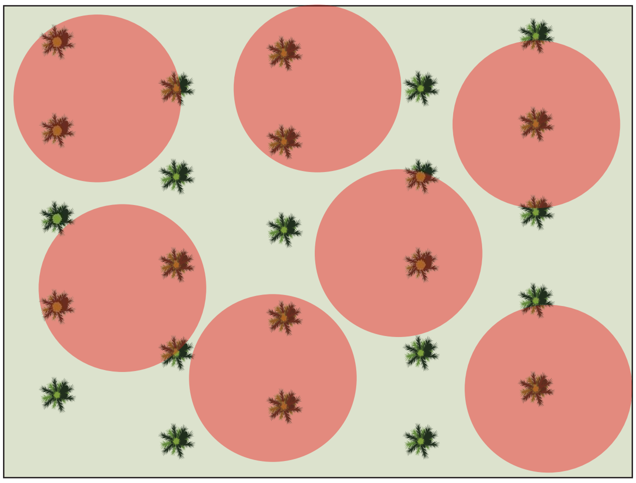

Figure 2: Sensor coverage of an area.

methods to recover the lost data. Using this method will significantly reduce the energy consumption of retransmission.

In the WSN deployment process, each wireless sensor is independent, and the data transmission is difficult, so a method is needed to connect each wireless sensor. In the past period, researchers have studied this problem in-depth and developed the concept of clustering. The purpose is to divide multiple wireless sensors into several clusters and select a cluster head for each cluster. The wireless sensors in the same cluster will transmit data to the cluster head to realize the connection between wireless sensors. To better divide wireless sensors into clusters, there are currently two clustering algorithms: K-means and K-medoids algorithm. In this article, we use the k-medoids method. Although K-means is very efficient, it is sensitive to outliers [32]. The K-medoids algorithm considers the most central object in the cluster, so it is less sensitive to outliers [33].

3. MAFMO

To achieve multiobjective optimization of the AFMO algorithm, we applied a repository strategy and a crowding degree-selection strategy to AFMO. The repository strategy is to open up dedicated storage space and store the nondominated solutions found so far [34, 35].

If there is a set of solutions Q for a multiobjective problem, $A \in Q$, then A is considered to be a nondominated solution if and only if the following conditions are satisfied [36]:

(i) All dimensions of A are better than or equal to the dimensions of other solutions.

(ii) Among all dimensions of A, at least one dimension is superior to the other solutions.

The second component is the crowding degree-selection strategy, which partitions the nondominated solution space into multiple grids. It calculates how crowded the particles are by the number of particles in the grid. The size of the repository is custom. During the iteration, compare the nondominated solution produced by this iteration with the solution in the repository, and update the repository according to the following conditions:

(i) Store solutions directly in the repository when the number of solutions in the repository is less than the repository size.

(ii) If the number of solutions in the repository equals the size of the repository, compare the congestion levels, remove the more congested solutions from the repository, and put the less congested solutions into the repository, making the final solutions more evenly distributed.

Based on the above two components, we propose the MAFMO algorithm. The following formulas update the velocity and position of the particles:

$$V_i^{t+1} = w \cdot \left(p_{\text{best}}^t - p_i^t\right) \cdot \frac{e_i^t}{E_{\max}} + \frac{\sum\left(fv_i - fv_r\right)}{\left|\sum\left(fv_i - fv_r\right)\right|} \cdot c \cdot rand \cdot \left(p_r^t - p_i^t\right), \tag{5}$$

$$p_i^{t+1} = p_i^t + V_i^{t+1}, \tag{6}$$

where w is the weight factor, which decreases from 2 to 0.4 as the number of iterations increases. In the single-objective algorithm, p_{best}^t is the current global optimal solution of the population, while in the multiobjective algorithm, we define it as a random solution in the nondominated solution repository and reselect it at each iteration. *rand* is a random function between 0 and 1. p_r^t is an individual randomly selected in the population, and fv_r is the fitness function value of p_r^t. It can be seen from the second half of Equation (5) that the randomly selected individual p_r^t can play a positive or negative role in the current particle. In the formula, we decide whether to play a positive or negative role by adding up the fitness values of each target of the current particle and then looking at their signs.

According to the description of the AFMO algorithm in the previous chapter, when no individual with a better fitness value is found, the fish will update its energy consumption. The growth stage also increases accordingly. The updated formula for the growth stage of the fish is shown in Equation (4). The update of the fish's energy expenditure is given by the following equation:

$$e_i^{t+1} = e_i^t + R_1 \cdot E_{\max} \cdot \frac{\sum \dfrac{\mathrm{fv_i} - \mathrm{fv_{best}}}{\mathrm{f_{max}} - \mathrm{fv_{best}}}}{\mathrm{D}}, \tag{7}$$

where R_1 is a random number between 0.2 and 0.4, $f_{\max}$ is the set of the maximum fitness values of all individuals currently, D represents the number of objectives, and fv_{best} represents the fitness of randomly selected individuals in the repository. The calculation range of the summation symbol in the formula is also the overall target quantity. The purpose of this is to be able to take into account the effects of all target dimensions.

```
Initialize the school of fish
  while less than the maximum number of iterations do
    for each particle do
        Update the position of the current particle by Equation (5) and Equation (6).
      end for
      if any of the new solutions are located outside the hypercubes then
        Update the gird to cover the new solution.
      end if
      if A dominated pbest then
        Update energy consumption.
        Update growth stage.
      end if
      The particle that reaches the fourth stage is initialized.
      Update the repository.
      if the repository is full then
        Run the grid mechanism, and update the repository.
      end if
    end while
```

Algorithm 1: MAFMO.

4. Problem Statement

In this paper, coverage rate, energy consumption, and overcoverage rate are regarded as the evaluation criteria for WSN performance, and the detail is presented below [37]. Among them, the coverage rate and energy consumption ensure two important goals when deploying WSN, such as covering a more extensive range and having a longer life cycle. If the coverage rate and energy consumption are not well optimized, it can be said that the WSN is not perfect. The overcoverage rate can limit the increase in deployment costs. Preventing WSN is from continuously increasing deployment costs to meet coverage and energy consumption.

4.1. Space Division. To facilitate the calculation of objective function, the deployment space is divided into multiple grids [38]. Each grid is abstracted as a point, each one square meter in size. If the deployment space is $L \times W \mathrm{m}^2$, then it is divided into $L \times W$ grids.

4.2. Coverage. Coverage is an important metric to evaluate whether our deployed WSN is reliable [39]. Calculate the distance between the grid and the wireless sensor. The grid is covered when the distance is less than the sensor radius [40]. At this time, the coverage matrix can be updated. The following equation gives the updated formula for the coverage matrix:

$$C(i, j) = \begin{cases} 1, & \text{if a sensor or sink node is deployed at position } x(i, j) \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where i and j are in the range $[0, L]$ and $[0, W]$, respectively, and then we solve for the overall sensor deployment coverage through the coverage matrix:

$$\mathrm{cover} = \frac{\sum_{i=1}^{l} \sum_{j=1}^{w} C(i, j)}{L \times W}. \tag{9}$$

4.3. Energy Consumption. When WSN is deployed, the battery power supply is difficult. Especially with many wireless sensors, the difficulty of charging can be imagined [41]. Therefore, we need to pay more attention to the energy consumption of sensors when deploying WSN [42]. Wireless sensors can be divided into sending and receiving sensors [43]. The sending sensor transmits the data collected to the receiving sensor [44, 45], and the star connection is used between the receiving sensor and the sending sensor. In the k-medoids algorithm, the population is divided into multiple clusters [46], each cluster is assigned a cluster head, and the particles closest to the cluster head are added to the cluster [33]. The following equation calculates the energy consumption:

$$\exp = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \left| \left(R_j - S_i \right) \right|}{M}. \tag{10}$$

4.4. Overcoverage. According to the above-mentioned approach, we know that the energy consumption of wireless sensors is a significant indicator for deploying WSN [47]. Repeated coverage of an area when deploying a wireless sensor network will waste the resources and reduce the service life of the WSN. Therefore, we propose the overlapping area calculation formula and the formula is as follows:

$$D(i,j) = \begin{cases} +1, & \text{if sensor or sink node is deployed at position } x(i,\ j) \\ +0, & \text{otherwise}, \end{cases} \tag{11}$$

$$\mathrm{Oc} = \frac{\sum_{i=1}^{l}\sum_{j=1}^{w} D(i,j)}{L \times W}. \tag{12}$$

4.5. Fitness Function of the Problem. The above three objectives have the same characteristics as the multiobjective algorithm, they cannot be optimal simultaneously. Therefore, we must balance these three objectives for the optimal deployment scheme. We propose a multiobjective weighted calculation formula. In Equation (13), the influence of each objective function on the formula result is balanced by adding the corresponding weight to each objective function. For example, if we have high requirements on the coverage of the target area, then we can increase the weight of the target "coverage". Finally, by comparing the results of Equation (13), we can find the optimal deployment scheme, that is, the deployment scheme with the smallest result after weighted evaluation. After adding weights to the objective function, we can flexibly select the optimal solution to deal with various application scenarios.

$$\mathrm{Fin} = W_1 \times \mathrm{cover} + W_2 \times \exp + W_3 \times \mathrm{Oc}. \tag{13}$$

5. Simulation Results and Discussion

In this section, the MAFMO algorithm is utilized to find the optimal deployment strategy with three objectives of WSN. The performance of the new algorithm and the ability of the new algorithm to solve practical problems are tested, respectively. The article uses Matlab software and experiments on a PC with an Intel Core i5-8500 3.0 GHz central processing unit (CPU) and 8 GB RAM.

5.1. Relevance Test. Over the past period, various deployment objectives have been used by many researchers. For example, cost, connectivity, life cycle, load balancing, monitoring quality, and so forth. Due to the nature of the multiobjective heuristic algorithm, there needs to be a negative correlation between the optimized objectives. We conducted a negative correlation test on the deployment objectives used in this paper to choose an appropriate solution.

The so-called negative correlation means that when one increases, the other decreases. Figure 3 shows that there is a negative correlation between every two objectives. Figure 3(a) shows the relationship between the coverage rate and the number of repeated coverage nodes. Figure 3(b) shows the relationship between coverage rate and energy consumption. Figure 3(c) shows the relationship between the number of repeated coverage nodes and energy consumption.

5.2. The Ability to Find the Pareto Front. To verify the ability of the proposed MAFMO algorithm to find the Pareto front, we use the ZDT, Schaffer and Kursawe algorithms for verification. During verification, the Pareto front given by the test function is printed in the figure, and the gray line is used to measure whether the solution obtained by the MAFMO algorithm is distributed on the Pareto front. The above several test functions are the commonly used multiobjective test functions, including various convex, nonconvex, and piecewise test functions. In Figure 4, the red asterisks represent nondominated solutions in the repository, and $f1$ and $f2$ represent the fitness values of the two objective functions. Figure 4 shows that no matter what kind of test function, the solution obtained by using the MAFMO algorithm can be uniformly distributed on the Pareto front, which almost coincides with the gray line (Pareto front).

5.3. Performance of MAFMO Algorithm. Furthermore, we test the distribution and malleability of the MAFMO algorithm. Distribution is an indicator to measure whether the solution set is uniformly distributed on the Pareto front. Malleability is an indicator to measure whether the solution set is densely distributed in the Pareto front boundary region. The evaluation indicators of multiobjective optimization mainly include counting indicators, convergence indicators, diversity indicators, and comprehensive indicators. The evaluation indicators of distribution and malleability we use here are the evaluation indicators that belong to diversity. For the evaluation indicators of distribution and malleability, researchers have proposed many calculation formulas, and here we choose SP and M_3^* calculation formulas. SP is to measure the distribution of the algorithm by comparing the distance between the two closest solutions in the solution set [48]. M_3^* calculates the sum of the distances between the farthest individuals in the solution set on each objective [49]. The calculation formula is as follows:

$$\mathrm{SP} = \sqrt{\sum_{i=1}^{|S|} \frac{\left(d_i - \overline{d}\right)^2}{|S| - 1}}, \tag{14}$$

$$M_3^* = \sqrt{\sum_{k=1}^{m} \max \left\| f_k(x^i) - f_k(x^j) \right\|}. \tag{15}$$

We used the evolutionary multiobjective optimization platform (platEMO) to test the indicators [50]. The platEMO platform is an open-source and free code base that integrates a large number of multiobjective algorithms, evaluation indicators, and test functions. During testing, we compare MOPSO, MOWOA, and NSGA-II algorithms using ZDT, Schaffer, and Kursawe as test functions. Tables 1 and 2 show that in terms of distribution, MAFMO is better than the other three functions except for Schaffer, and in terms of malleability, ZDT1 and Kursawe are better than the other three multiobjective functions.

5.4. WSN Deployment Under Multiobjective Algorithm. In the experiment, we consider setting the deployment space as a 2D map, the size of the deployment range is defined as 100×100 meters, the population number is set as 50. The

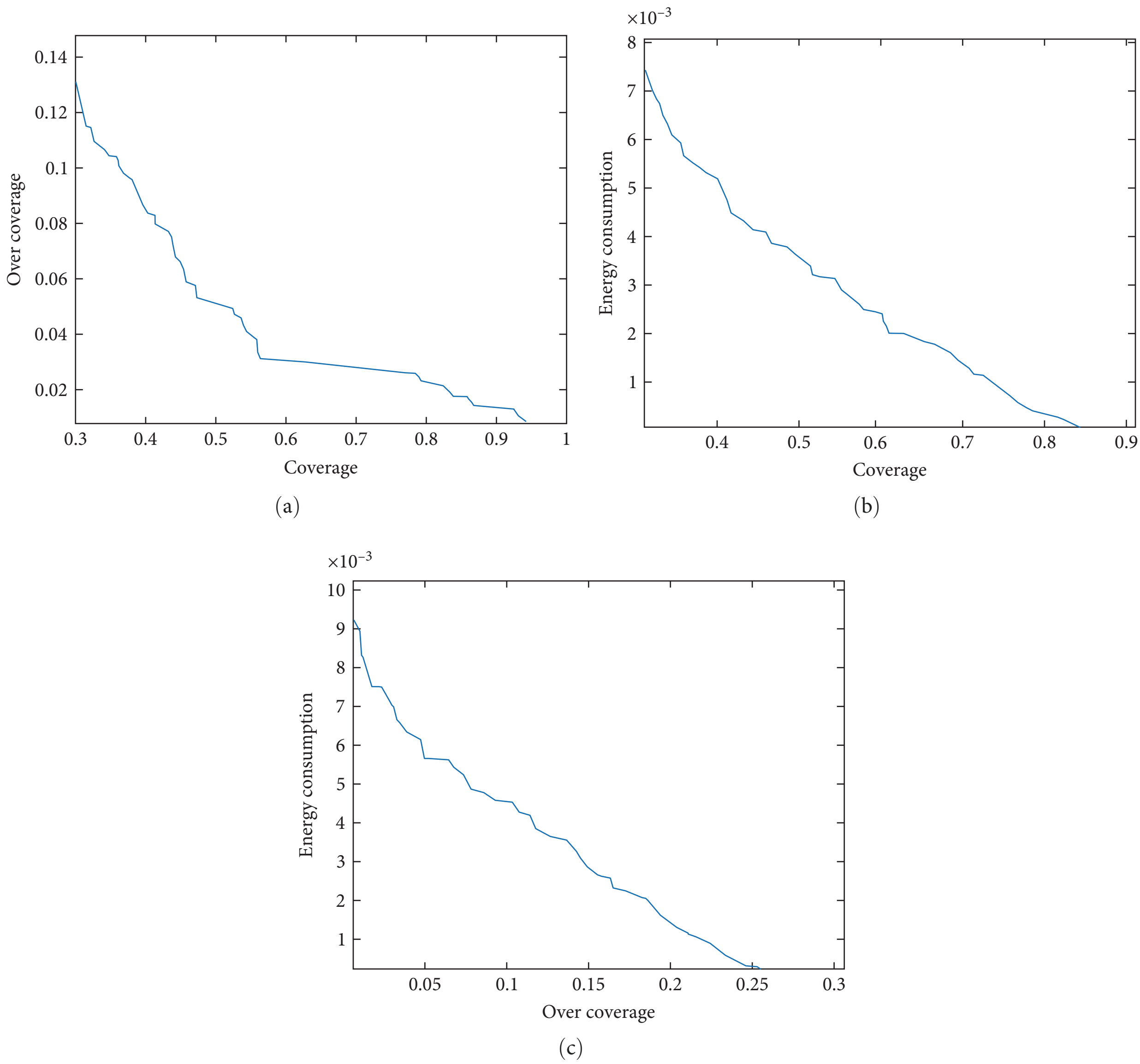

Figure 3: (a, b, and c) The relationship between objectives.

number of iterations is 1,000 generations. Next, we use MAFMO, MOPSO, MOWOA, and NSGA-II for optimization [51]. In order to be able to choose the most suitable solution from the many solution sets, we assign weights to different objective functions. Here, we set the proportion of coverage weight W_1 to 0.4 and the proportions of energy consumption and repeated coverage weight W_2 and W_3 to 0.3.

Next, we test the performance of the multiobjective heuristic by setting the number of sensors, sensor radius, and the number of receiving sensors in the WSN to different values.

5.4.1. Different Number of Sensors. When deploying a WSN, the number of sensors has a significant impact on the performance of the WSN. In the experiments, we used 30, 40, and 50 sensors for verifying the performance of multiobjective heuristic algorithms, the sensor radius is set to 10, and the number of receiving sensors is set to 10. Then find 50 optimal deployment schemes through four algorithms and calculate the final fitness value according to the corresponding weighting formula. The comparison results of the four algorithms are shown in Figure 5.

Figure 5 shows that under the use of different numbers of sensors, the variation of the results obtained by the four algorithms is relatively gentle. Regardless of the number of sensors, the fitness value obtained by the MAFMO algorithm is better than the other three algorithms.

5.4.2. Different Sensor Radius. In this subsection, we fix the total number of sensors and the number of receiving sensors. Set the sensor radius to a different value. Since the sensor coverage area is a circle, when the radius of the sensor is set to a different size, the gap between the sensor and the sensor will be different. In the experiments in this subsection, we set the sensor radius to 8, 10, and 12, and we used 40 sensors,

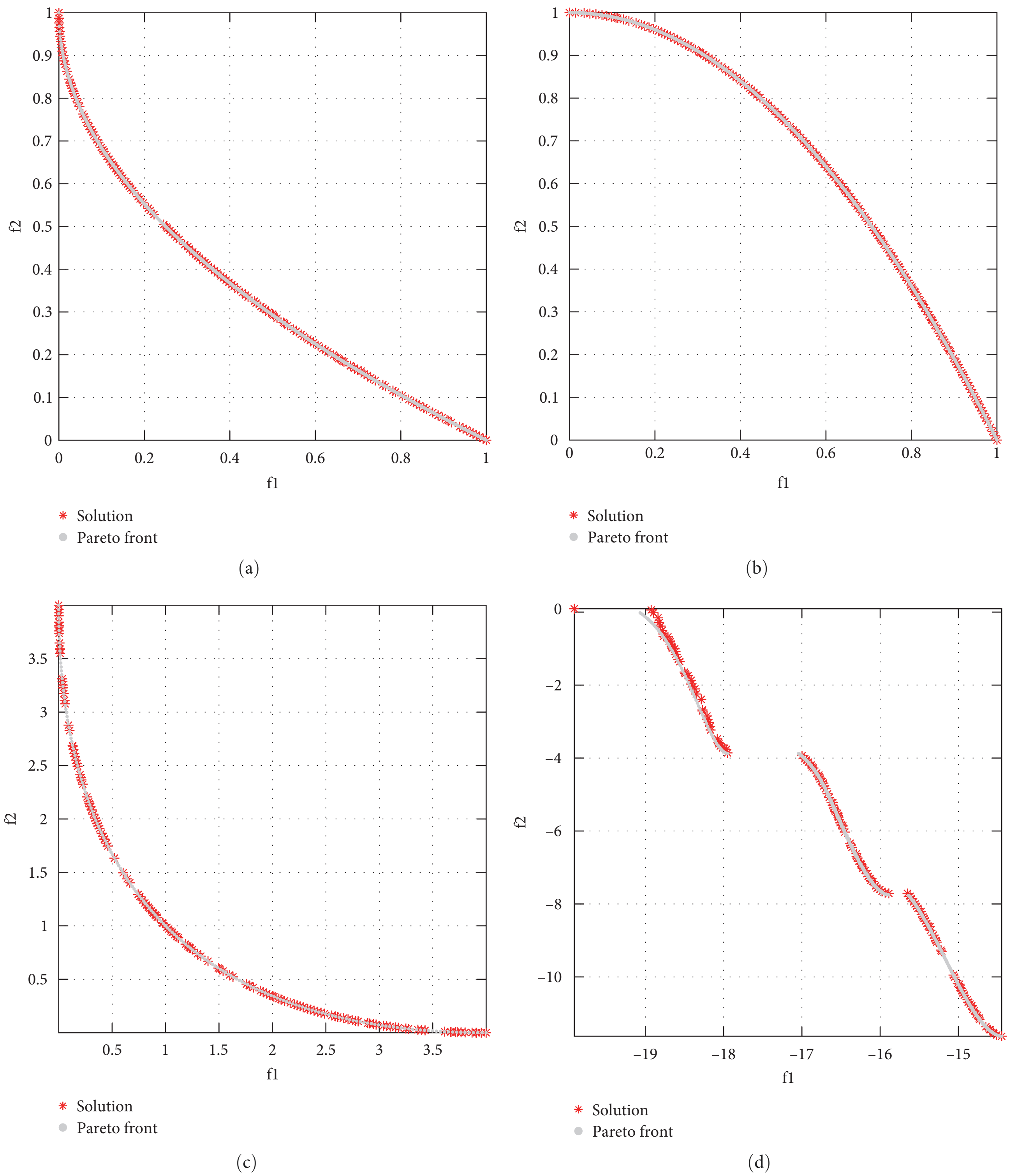

Figure 4: (a, b, c, and d) MAFMO test results using ZDT1, ZDT2, Schaffer and Kursawe test functions.

including 10 receiving sensors. The experimental procedure is similar to the previous subsection.

Figure 6 shows that when the radius of the sensor is 8, the MAFMO algorithm is better than MOWOA and MOPSO, but slightly worse than NSGA-II. When the radius of the sensor is 10, the MAFMO algorithm is significantly better than the MOWOA and NSGA-II algorithms. However, there is still a particular gap compared with the MOPSO algorithm. When the sensor radius is 12, the MAFMO algorithm is better than the other three.

Table 1: The experiment results of different algorithms in distribution test.

Function	MAFMO	MOPSO	NSGA-II	MOWOA
ZDT1	**0.0057**	0.0067	0.0067	0.0103
ZDT2	**0.0042**	0.0095	0.0061	0.0171
Kursawe	0.0458	**0.0135**	0.0273	0.0468
Schaffer	**0.0179**	0.0933	0.0201	0.0974

Bold value represents the optimal solutions of the four algorithms under the same test problem.

Table 2: The experiment results of different algorithms in ductility test.

Function	MAFMO	MOPSO	NSGA-II	MOWOA
ZDT1	**1.4142**	1.2639	0.4012	1.1025
ZDT2	1.2125	1.1757	0.1484	**1.3235**
Kursawe	**3.0621**	2.5547	2.4229	2.8286
Schaffer	2.4474	3.5791	2.5687	**3.7816**

Bold value represents the optimal solutions of the four algorithms under the same test problem.

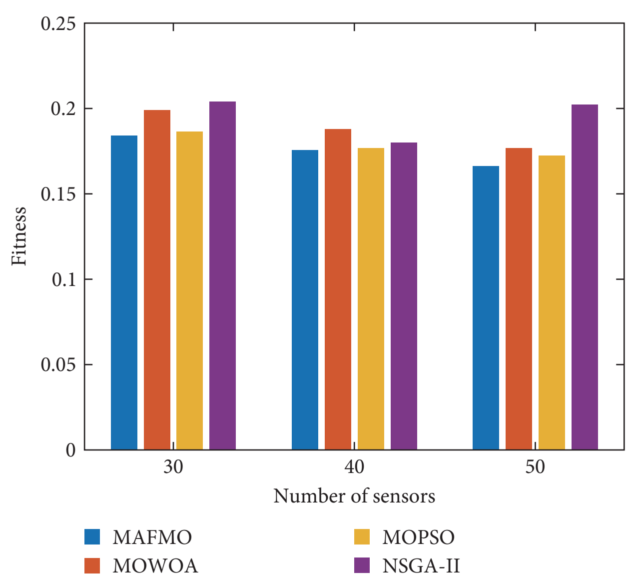

Figure 5: Solutions of MAFMO, MOWOA, MOPSO and NSGA-II algorithms with different number of sensors.

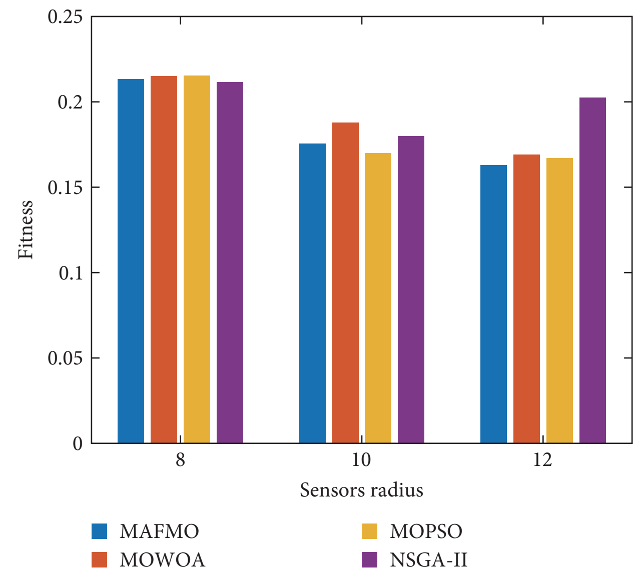

Figure 6: Solutions of MAFMO, MOWOA, MOPSO and NSGA-II algorithms with different radius.

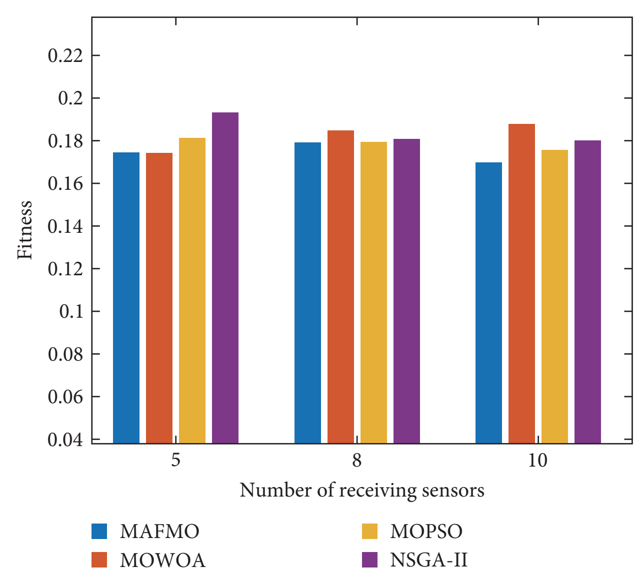

Figure 7: Solutions of MAFMO, MOWOA, MOPSO and NSGA-II algorithms with different numbers of receiving sensors.

5.4.3. Different Number of Receiving Sensors. In the experiments in this subsection, we use different numbers of receiving sensors to test the WSN deployment effect under each multiobjective algorithm. The receiving sensor mentioned here is the cluster head mentioned in the clustering. Different numbers of receiving sensors have different effects on WSN deployment. Here, we experiment with 5, 8, and 10 receiving sensors. The experimental results are shown in Figure 7.

Figure 7 shows that when the number of receiving sensors is 5, the solution results of the four algorithms are almost the same. The MAFMO algorithm is better than the MOPSO and NSGA-II algorithms, but slightly worse than the MOWOA algorithm. When the number of receiving sensors is 8, the experimental result of the MAFMO algorithm is better than that of MOWOA and NSGA-II, and it is almost the same as that of MOPSO. When the number of receiving sensors is 10, the MAFMO algorithm is better than the other three.

6. Conclusion

This paper proposes a new multiobjective heuristic algorithm MAFMO based on the AFMO algorithm and tests it through the ZDT, Schaffer and Kursawe test function. The test results show that the MAFMO can be quickly and uniformly distributed on the Pareto front. We apply the proposed multiobjective heuristic algorithm to WSN deployment, taking into account the deployment requirements of sensor coverage, energy consumption, and repeated coverage. Finally, compared with the WSN deployment scheme using MOPSO, MOWOA, and NSGA-II multiobjective heuristic algorithms. The results show that the deployment schemes solved by the MAFMO algorithm work better under the same weight addition.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the Laboratory for Marine Geology, Qingdao National Laboratory for Marine Science and Technology, grant no. MGQNLM-KF201807.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
[2] O. Singh, V. Rishiwal, R. Chaudhry, and M. Yadav, "Multi-objective optimization in WSN: opportunities and challenges," *Wireless Personal Communications*, vol. 121, pp. 127–152, 2021.
[3] D. Snoonian, "Smart buildings," *IEEE Spectrum*, vol. 40, no. 8, pp. 18–23, 2003.
[4] J. Keenan and A. J. Motley, "Radio coverage in buildings," *British Telecom Technology Journal*, vol. 8, pp. 19–24, 1990.
[5] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control*, pp. 719–724, IEEE, Limassol, Cyprus, 2005.
[6] G. Molina, E. Alba, and E.-G. Talbi, "Optimal sensor network layout using multi-objective metaheuristics," *Journal of Universal Computer Science*, vol. 14, no. 15, pp. 2549–2565, 2008.
[7] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 2, pp. 472–476, IEEE, Helsinki, Finland, 2001.
[8] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1767–1775, 2009.
[9] M. A. Benatia, M. Sahnoun, D. Baudry, A. Louis, A. El-Hami, and B. Mazari, "Multi-objective WSN deployment using genetic algorithms under cost, coverage, and connectivity constraints," *Wireless Personal Communications*, vol. 94, pp. 2739–2768, 2017.
[10] P.-C. Song, J.-S. Pan, and S.-C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Applied Soft Computing*, vol. 94, Article ID 106443, 2020.
[11] D. Gao, G.-G. Wang, and W. Pedrycz, "Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3265–3275, 2020.
[12] Z.-G. Du, J.-S. Pan, S.-C. Chu, and Y.-J. Chiu, "Multi-group discrete symbiotic organisms search applied in traveling salesman problems," *Soft Computing*, vol. 26, pp. 4363–4373, 2022.
[13] G.-G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 542–555, 2019.
[14] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction," in *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing*, L. Wang, A. Ng, and K. Deb, Eds., pp. 3–34, Springer, London, 2011.
[15] C. A. C. Coello and M. S. Lechuga, "Mopso: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 2, pp. 1051–1056, IEEE, Honolulu, HI, USA, 2002.
[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, WA, Australia, 1995.
[17] W.-M. Zheng, N. Liu, Q.-W. Chai, and S.-C. Chu, "A compact adaptive particle swarm optimization algorithm in the application of the mobile sensor localization," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 1676879, 15 pages, 2021.
[18] P. Hu, J.-S. Pan, S.-C. Chu, and C. Sun, "Multi-surrogate assisted binary particle swarm optimization algorithm and its application for feature selection," *Applied Soft Computing*, vol. 121, Article ID 108736, 2022.
[19] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[21] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
[22] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. dos S. Coelho, "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, vol. 47, pp. 106–119, 2016.
[23] J.-S. Pan, X.-X. Sun, S.-C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved SMS applied for QR code," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104049, 2021.
[24] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, "Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 439–453, 2019.
[25] C. He, R. Cheng, and D. Yazdani, "Adaptive offspring generation for evolutionary large-scale multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 786–798, 2022.
[26] X. Yang, J. Zou, S. Yang, J. Zheng, and Y. Liu, "A fuzzy decision variables framework for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 445–459, 2023.
[27] L. Li, C. He, R. Cheng, H. Li, L. Pan, and Y. Jin, "A fast sampling based evolutionary algorithm for million-dimensional

multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 75, Article ID 101181, 2022.

[28] Q.-W. Chai and J. W. Zheng, "Rotated black hole: a new heuristic optimization for reducing localization error of WSN in 3D Terrain," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9255810, 13 pages, 2021.

[29] Z. T. Ayoub, S. Ouni, and F. Kamoun, "Energy consumption analysis to predict the lifetime of IEEE 802.15. 4 wireless sensor networks," in *Third International Conference on Communications and Networking*, pp. 1–6, IEEE, Hammamet, Tunisia, 2012.

[30] M. A. Mahmood, W. K. G. Seah, and I. Welch, "Reliability in wireless sensor networks: a survey and challenges ahead," *Computer Networks*, vol. 79, pp. 166–187, 2015.

[31] H. Wen, C. Lin, F. Ren, Y. Yue, and X. Huang, "Retransmission or redundancy: transmission reliability in wireless sensor networks," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1–7, IEEE, Pisa, Italy, 2007.

[32] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, Project euclid, Oakland, CA, USA, 1967.

[33] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3336–3341, 2009.

[34] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

[35] B. C. Arnold, "Pareto distribution," in *Wiley StatsRef: Statistics Reference Online*, N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri, and J. L. Teugels, Eds., pp. 1–10, John Wiley & Sons, Ltd., 2015.

[36] R. Dunford, Q. Su, and E. Tamang, "The pareto principle," *The Plymouth Student Scientist*, vol. 7, no. 1, pp. 140–148, 2014.

[37] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle, "Artificial potential field approach in WSN deployment: cost, QoM, connectivity, and lifetime constraints," *Computer Networks*, vol. 55, no. 1, pp. 84–105, 2011.

[38] M. Al-Turjman Fadi, E. Al-Fagih Ashraf, S. H. Hossam, and A. I. Mohamed, "Deploying fault-tolerant grid-based wireless sensor networks for environmental applications," in *IEEE Local Computer Network Conference*, pp. 715–722, IEEE, Denver, CO, USA, 2010.

[39] A. Bereketli and O. B. Akan, "Communication coverage in wireless passive sensor networks," *IEEE Communications Letters*, vol. 13, no. 2, pp. 133–135, 2009.

[40] P.-J. Wan and C.-W. Yi, "Coverage by randomly deployed wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2658–2669, 2006.

[41] H. Wang, H. E. Roman, L. Yuan, Y. Huang, and R. Wang, "Connectivity, coverage and power consumption in large-scale wireless sensor networks," *Computer Networks*, vol. 75, Part A, pp. 212–225, 2014.

[42] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1976–1984, IEEE, Miami, FL, USA, 2005.

[43] R. Alageswaran, R. Usha, R. Gayathridevi, and G. Kiruthika, "Design and implementation of dynamic sink node placement using particle swarm optimization for life time maximization of WSN applications," in *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012)*, pp. 552–555, IEEE, Nagapattinam, India, 2012.

[44] F. Chen and R. Li, "Sink node placement strategies for wireless sensor networks," *Wireless Personal Communications*, vol. 68, pp. 303–319, 2013.

[45] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, 2005.

[46] P. Arora, Deepali, and S. Varshney, "Analysis of K–means and K–medoids algorithm for big data," *Procedia Computer Science*, vol. 78, pp. 507–512, 2016.

[47] C.-C. Lai, C.-K. Ting, and R.-S. Ko, "An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications," in *2007 IEEE Congress on Evolutionary Computation*, pp. 3531–3538, IEEE, Singapore, 2007.

[48] J. R. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Massachusetts Institute of Technology, 1995.

[49] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization Methods and Applications*, Vol. 63, Citeseer, 1999.

[50] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[51] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.