Balancing the trade-off between cost and reliability for wireless sensor networks: a multi-objective optimized deployment method

Long Chen1 · Yingying Xu1 · Fangyi Xu1 · Qian Hu1 · Zhenzhou Tang1

Accepted: 7 June 2022 / Published online: 5 August 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

The deployment of the sensor nodes (SNs) always plays a decisive role in the system performance of wireless sensor networks (WSNs). In this work, we propose an optimal deployment method for practical heterogeneous WSNs which gives a deep insight into the trade-off between the reliability and deployment cost. Specifically, this work aims to provide the optimal deployment of SNs to maximize the coverage degree and connection degree, and meanwhile minimize the overall deployment cost. In addition, this work fully considers the heterogeneity of SNs (i.e. differentiated sensing range and deployment cost) and three-dimensional (3-D) deployment scenarios. This is a multi-objective optimization problem, non-convex, multimodal and NP-hard. To solve it, we develop a novel swarm-based multi-objective optimization algorithm, known as the competitive multi-objective marine predators algorithm (CMOMPA) whose performance is verified by comprehensive comparative experiments with ten other state-of-the-art multi-objective optimization algorithms. The computational results demonstrate that CMOMPA is superior to others in terms of convergence and accuracy and shows excellent performance on multimodal multi-objective optimization problems. Sufficient simulations are also conducted to evaluate the effectiveness of the CMOMPA based optimal SNs deployment method. The results show that the optimized deployment can balance the trade-off among deployment cost, sensing reliability and network reliability. The source code is available on https://github.com/iNet-WZU/CMOMPA.

Keywords Cost-saving deployment optimization, K-coverage, C-connectivity, Multi-objective marine predator algorithms · Wireless sensor network

1 Introduction

In recent years, wireless sensor networks (WSNs) have been widely used in industry, military, agriculture, health, and other fields [1], where the deployment of the sensor nodes (SNs) always plays a decisive role in the system performance. In practical WSNs, it is common that SNs may run out of power or become damaged and fail to work, which may lead to network disconnection or monitoring blind spots. Therefore, to ensure the effectiveness of sensing and the robustness of the network, an appropriate node deployment should be able to achieve the $K$-coverage over the target points to be monitored and the $C$-connectivity of the network.

Specifically, $K$-coverage means that each target point to be monitored is sensed by at least $K$ SNs so that $K - 1$ SNs failure can still guarantee the sensing coverage. On the other hand, $C$-connectivity means that each SN is wirelessly connected with other $C$ SNs to effectively avoid network bottleneck nodes and network segmentation. Figure 1 illustrates some examples to more intuitively describe $K$-coverage and $C$-connectivity. SNs represented by black circles and blue circles are with different sensing ranges. $SN_1$ is 2-connectivity with $SN_2$, $SN_7$. $SN_5$ is 3-connectivity with $SN_4$, $SN_6$, and $SN_9$. Target point $t_1$ is 3-coverge with $SN_2$, $SN_7$, and $SN_8$. $t_2$ is 1-coverge with $SN_8$, $t_3$ realizes 0-coverage.

Studies on WSN deployment considering connectivity and coverage have been frequently reported. The review of literature on this area is summarized in Section 2 and
Balancing the trade-off between cost and reliability for wireless sensor networks: a multi-objective optimized...

Table 1. Although these existing works have done much research on optimizing the coverage and connectivity of WSNs, there are still some problems that remain open.

Firstly, they did not consider the deployment cost when optimizing the coverage and connectivity. For large-scale WSNs deployed in hard environments, such as forests and deep water, it is very important to carefully consider the total network costs, including node deployment cost and node manufacturing cost. Obviously, K-coverage, C-connectivity, and the total cost of network deployment are mutually restrictive goals. To be specific, in a given monitoring area, the more SNs deployed, the higher the connectivity and coverage, however, the higher the cost. Therefore, it is very meaningful to study the trade-off among K-coverage, C-connectivity, and the total cost. Secondly, the existing works ignored the fact that SNs in practical applications are typically heterogeneous. That is, there exist gaps in the sensing radius and deployment cost between different SNs. Thirdly, most works assumed that sensor nodes are deployed in a two-dimensional (2-D) environment, however, the deployment area in practical applications is often three-dimensional (3-D).

In view of the above problems, this paper investigates the optimal WSN deployment problem featuring the following highlights:

1) Three optimization objectives are considered in our work. Specifically, we aim to maximize the coverage degree which is defined as the number of SNs that can sense the target, the connection degree which is defined as the number of SNs that an SN can directly communicate with, and meanwhile minimize the overall deployment cost. Our work gives a deep insight into the trade-off among effectiveness, reliability and cost.

2) When modeling the network, we fully consider the heterogeneity of SNs, which is reflected in the differences of sensing ranges and deployment costs. Besides, the cost of deploying SNs also varies in different locations.

3) The deployment scenario is a 3-D environment, which is more in line with the actual deployment application.

The above optimization deployment problem is very complex, since it is a non-convex, multimodal multi-objective optimization problem. Besides, it is also a high-dimensional multi-objective optimization problem with large-scale decision variables because of the large number of SNs. It is an extremely tough challenge for classical convex optimization methods to find the Pareto optimal solutions. Fortunately, in recent years, swarm intelligence optimization algorithms such as the marine predators algorithm (MPA) [2], the artificial bee colony algorithm [3] and the particle swarm optimization algorithm [4], etc., have provided promising methods to search for the approximate optimal solutions of complex optimization problems [5–12]. There have also been some multi-objective versions of swarm intelligence optimization algorithms developed to deal with multi-objective optimization problems, such as multi-objective artificial bee colony algorithm [13], multi-objective seagull optimization algorithm [14] and multi-objective whale optimization algorithm [15], etc. The effectiveness of these multi-objective swarm intelligence optimization algorithms has also been verified in scientific research and engineering practices [16–21].

Multi-objective swarm intelligence optimization algorithms have two significant advantages for solving multi-objective optimization problems. Firstly, we can obtain a set of approximated Pareto optimal solutions, instead of one single in a single run. This set of solutions provides the entire range of solutions and the shape of the approximated Pareto-optimal front, which can more efficiently demonstrate the trade-off between the objectives to the decision-maker, who after that he/she chooses one out of the set of alternatives. Secondly, swarm intelligence optimization
| Ref  | Objections                              | 2-D/3-D | Methodology                      |
|------|-----------------------------------------|---------|----------------------------------|
| [30] | ✓ Network uniformity, deployment time   | 2-D     | Heuristic Algorithm              |
| [31] | ✓ Energy consumption                    | 2-D     | Heuristic Algorithm              |
| [32] | ✓ Time to calculate the fitness function| 2-D     | Heuristic Algorithm              |
| [33] | ✓ Power consumption of SNs              | 2-D     | Heuristic Algorithm              |
| [34] | ✓ Coverage duration                     | 2-D     | Heuristic Algorithm              |
| [35] | ✓ Indicator of availability, time, reliability | 2-D | Mathematical optimization |
| [36] | ✓ Target detection                      | 2-D     | Mathematical optimization        |
| [37] | ✓ Fixed budget                          | 2-D     | Graph theory                     |
| [38] | ✓ Residual energy of the SNs            | 2-D     | Heuristic Algorithm              |
| [39] | ✓ Sent Bytes, movement cost of SNs      | 2-D     | Mathematical optimization        |
| [40] | ✓ The number of relay nodes             | 2-D     | Graph theory                     |
| [41] | ✓ The number of sensors to activate     | 2-D     | Mathematical optimization        |
| [42] | ✓ The number of SNs                     | 2-D     | Heuristic Algorithm              |
| [43] | ✓ The deployment cost of network        | 2-D     | Heuristic Algorithm              |
| [44] | ✓ The wakeup scheduling scheme of SNs   | 3-D     | Heuristic Algorithm              |

Symbol ✓ indicates that the corresponding objective has been optimized
Conn. and Cov. are abbreviations for connectivity and coverage, respectively

Given the above, we propose a multi-objective swarm intelligence optimization algorithm, known as CMOMPA (competitive multi-objective MPA), to solve the multi-objective optimization problem formulated in this work. The motivation behind using MPA is two-fold. On the one hand, balancing the trade-off between cost and reliability for a WSN with a large number of nodes is a multimodal optimization problem with considerably high-dimensional decision variables. MPA is very competitive in solving the above problems [2]. On the other hand, MPA has the advantages of low computational complexity, few parameters, and fast convergence, which has attracted many interests [22–24].

Moreover, CMOMPA incorporates the idea of a competitive swarm optimizer [25] to further improve its performance on multimodal multi-objective optimization problems. The algorithms are not sensitive to the convexity of the objective functions and the hardness of the problems. It is particularly attractive to optimization problems in engineering since most of them are non-convex and NP-hard.
competitive swarm optimizer updates the population by the competitive learning among individuals in each iteration, rather than the personal and/or global best individual. It has a better ability to exploit the small gap between two individuals whose fitness values are very close, which makes competitive swarm optimizer more powerful in dealing with multi-objective optimization problem with a large number of local optima [25]. In addition, CMOMPA uses the solutions selection method based on reference points [26], which can enhance the distribution diversity of solutions.

We have carried out extensive comparative experiments to verify the performance of CMOMPA on 20 popular benchmark functions selected from three popular benchmark suites on multi-objective optimization problems, i.e. ZDT [27], WFG [28], and DTLZ [29]. Moreover, CMOMPA has been compared with ten other state-of-the-art multi-objective optimization algorithms. The results show that the proposed CMOMPA outperforms other algorithms in terms of global convergence and accuracy.

To sum up, the main contributions of this paper are summarized as follows:

1) We build a practical optimal deployment model for WSNs. The model includes three optimization objectives, i.e., deployment cost, connection degree, and coverage degree. This optimization problem is constrained by the full coverage requirement of the monitoring area, the full connectivity of the network, and the minimum degree of coverage and connectivity. Moreover, the model fully considers the heterogeneity of SNs and 3-D deployment scenarios.

2) We propose a multi-objective swarm intelligence optimization algorithm, known as CMOMPA, to solve the above optimal deployment model. CMOMPA takes MPA as the optimizer and the reference point based method to determine elite individuals. In addition, CMOMPA leverages the individual competition mechanism to produce offspring with stronger diversity and distribution. CMOMPA can efficiently solve multimodal problems. The source code is available on https://github.com/iNet-WZU/CMOMPA.

3) Finally, we present comprehensive comparative experiments to evaluate the performance of the proposed CMOMPA. The results demonstrate that CMOMPA is superior to other algorithms in terms of convergence and accuracy and shows excellent performance for multimodal multi-objective optimization problems. In addition, we apply CMOMPA to the cost-saving optimal deployment scenario of WSNs. The results show that the optimized deployment can well balance the trade-off among deployment cost, sensing reliability, and network reliability.

For ease of reading, we have summarized the acronyms used in this paper and their full names as follows.

- **SNs**: Sensor nodes;
- **WSNs**: Wireless sensor networks;
- **MPA**: Marine predators algorithm
- **MOMPA**: Multi-objective MPA
- **CMOMPA**: Competitive multi-objective MPA
- **IGD**: Inverted generational distance
- **PFs**: Pareto fronts
- **NSGA-II**: Non-dominated sorting genetic algorithm II
- **NSGA-III**: Non-dominated sorting genetic algorithm III
- **PESA-II**: Pareto Envelope based Selection II
- **CMOPSO**: Competitive multi-objective particle swarm optimization
- **NSLS**: Non-dominated sorting and local search
- **CCMO**: Coevolutionary constrained multi-objective optimization
- **DCNSGA-III**: Dynamically constrained NSGA-III
- **MOEA/D**: Multi-objective evolutionary algorithm based on decomposition
- **CMOEA-MS**: Constrained multi-objective evolutionary algorithms with a two-stage

The rest of this paper is organized as follows. Section 2 introduces the related work on network deployment and multi-objective swarm intelligence optimization algorithms. Section 3 describes the system models and problem formulation. Section 4 presents the proposed CMOMPA and the optimal SN deployment method. The computational and simulation results and analysis are illustrated in Section 5. Finally, Section 6 concludes the paper and identifies the future work.

2 Related works

This section firstly reviews the existing models optimizing coverage and connectivity in WSNs. After that, we summarize some state-of-the-art multi-objective swarm intelligence optimization algorithms, particularly the existing multi-objective MPAs (MOMPAs), and highlight the necessity of proposing CMOMPA.

2.1 Optimal WSNs deployment

The optimal deployment of SNs has always been one of the hottest topics in the field of WSNs, where coverage and connectivity are two important optimization objectives.

In terms of coverage optimization, some works improved the coverage, network lifetime, energy consumption, and connectivity by optimally controlling the locations of mobile
nodes [30–32, 37, 38]. On the other hand, heuristic algorithms have shown promising effectiveness in optimizing network coverage [33–36, 39]. To maximize network coverage, a genetic algorithm based on Monte Carlo standard distribution method was used to deploy SNs in [33]. The study in [34] combined the greedy migration mechanism with the ant colony algorithm to deploy the SNs and improved the network coverage while reducing the deployment cost. The authors in [35] proposed a restrained Lloyd algorithm and a deterministic annealing algorithm to optimize the deployment of nodes in sensor networks and improve the overall performance of the network. In [36], a hybrid memetic framework was put forward to optimize the coverage of sensors so that the WSN can always maintain good coverage. In [39], a genetic algorithm was proposed to solve maximum coverage deployment in the 2-D environment.

In connectivity optimization, the authors in [42] proposed the path coloring-based connectivity detection algorithm and theoretically proves the superiority of the proposed algorithm. In [43], the connectivity of underwater optical sensor networks were analyzed using graph theory. In [40], the connectivity analysis models of WSNs under unicast and broadcast were provided. The study in [44] considered the deployment of sensor networks under the most general constraints. The study in [41] proposed three clustering algorithms to analyze the leading indicators of network connectivity, including availability indicators, time indicators, and reliability indicators. In [45], an improved non-dominated sorting genetic algorithm II (NSGA-II) algorithm was proposed to solve the connectivity problem of sensor networks to improve the network lifetime. A distributed connectivity restoration approach for a heterogeneous WSN was proposed in [46], which effectively improves the robustness of the network.

There have also been some works on the joint optimization of coverage and connectivity. In [53], the study took the coverage and connectivity of sensor networks into consideration comprehensively and proposed an optimal deployment method for a tunnel monitoring system. The study in [54] proposed a genetic algorithm based node placement method for 2-D WSNs, which improved the results by wavelets to achieve maximum coverage and connectivity. In [55], decentralized solutions were developed for determining and adjusting the levels of coverage and connectivity of a given network. In [56], an improved artificial bee colony algorithm was proposed to solve the optimal coverage and connectivity of WSNs. The [57] put forward the connectivity-based k-coverage hole detection algorithm, which can effectively monitor coverage holes and improve network coverage and connectivity. In [58], a sequential hybrid method was proposed to solve the coverage problem of sensor networks while taking into account connectivity. The study in [59] proposed four heuristic algorithms to solve the connectivity and coverage problems in wireless networks. In [60], an energy efficient wakeup scheduling scheme based on an improved Memetic algorithm was proposed, where four constraints, namely energy consumption, coverage, connectivity, and optimal length of wakeup schedule list were considered. The work in [61] ensured the coverage and connectivity of nodes in the network by miming a spider canvas.

Some works also consider the K-coverage of targets and the C-connectivity of nodes. In [47], the authors adopted the decomposition-based multi-objective algorithm to solve the problems of node C-connectivity and energy allocation in WSNs. In [48], considering different communication ranges of nodes, the authors proposed a C-connected scheme based on the injection of relay nodes in WSNs. The proposed method can better solve the problem of the connectivity of nodes. Both [47] and [48] considered the C-connectivity of nodes, but do not consider the K-coverage of targets. In [49], the authors proposed a novel algorithm called partial coverage learning automata, which ensures the energy of the network by adjusting the sleep state of nodes, to simultaneously maintain the network coverage and connectivity. However, the proposed method can only ensure effective coverage and connectivity within a given sub-area. In [50], the K-coverage and C-connectivity problem was handled by a mathematical model which integrated the Nelder–Mead method and the shuffled frog leading algorithm. In [51, 52], the authors had proposed solutions based on a single objective heuristic algorithm, considering the minimum number of nodes, K-coverage, and C-connectivity. In [50–52], the multi-objective problems were transformed into single objectives by weighted methods. But the weighted solution can not well reflect the trade-off between various objectives, and the heterogeneity of nodes was not considered. In [45], the authors dealt with four objective optimization problems by NSGA-II, i.e., minimizing the number of nodes, K-coverage, C-connectivity, and maximizing the energy level of nodes. However, the performance of the NSGA-II algorithm usually degrades when the number of optimization objectives is greater than two.

Table 1 summarizes the aforementioned literature. We can observe that none of them jointly optimize the cost, K-coverage, and C-connectivity. As far as we know, this work is the first to investigate the trade-off among the cost, network reliability, and sensing effectiveness.

### 2.2 Multi-objective swarm intelligence optimization algorithms

Recently, a lot of studies have been devoted to multi-objective swarm intelligence optimization algorithms. In general, a multi-objective swarm intelligence optimization algorithm involves two essential strategies, i.e., offspring...
Balancing the trade-off between cost and reliability for wireless sensor networks: a multi-objective optimized...

The commonly used elite selection strategies are non-dominated sorting [69], the crowding distance approach [69], the neighborhood based method [70], the reference point based method [26], and so on.

In [13], the authors proposed a multi-objective version of artificial bee colony algorithm, which uses the artificial bee colony algorithm to generate offspring individuals and non-dominated sorting to select elite individuals. In [71], a multi-objective swarm intelligence optimization algorithm that combines differential-evolution and particle swarm optimization was proposed which uses a hybrid algorithm to generate offspring individuals and non-dominated sorting to select elite individuals. A multi-objective seagull optimization algorithm was proposed in [14]. The authors used the seagull optimization algorithm to generate offspring individuals, and selected elite individuals using non-dominated sorting and grid methods. In [72], the authors proposed a multi-objective whale optimization algorithm, which uses the whale optimization algorithm to generate offspring individuals, and uses non-dominated sorting and Nelder-Mead simplex to select elite individuals. In [73], the authors proposed a multi-objective red deer algorithm, which uses a modified red deer algorithm to generate offspring individuals and selects elite individuals with roulette wheel selection. A hybrid multi-objective evolutionary algorithm was proposed in [74], which uses a crossover mutation operation to generate offspring individuals and selects elite individuals by comparing the Pareto front. In [75], the authors proposed a multi-objective optimization framework, which uses a hybrid swarm intelligence optimization algorithm to generate offspring individuals. In [76], the authors proposed a multi-objective scheduling fruit fly algorithm, which uses the fruit fly algorithm to generate offspring individuals and selects elite individuals with roulette wheel selection. A hybrid multi-objective evolutionary algorithm was proposed in [74], which uses a crossover mutation operation to generate offspring individuals and selects elite individuals by comparing the Pareto front. In [75], the authors proposed a multi-objective optimization framework, which uses a hybrid swarm intelligence optimization algorithm to generate offspring individuals. In [76], the authors proposed a multi-objective scheduling fruit fly algorithm, which uses the fruit fly algorithm to generate offspring individuals and selects elite individuals with roulette wheel selection.

There have also been some reports on multi-objective MPA, which have been summarized in Table 2. At the algorithm level, these algorithms are mainly different in the generation and selection strategy of offspring individuals. In [62], the authors proposed the two-objective MPA to deal with the problem of decreasing the emission of greenhouse gases and the fuel cost. In [65], an enhanced multi-objective optimization algorithm of MPA was proposed to handle problems with three objectives. In [64], the authors provided an improved multi-objective MPA for the optimized performance of combined alternating/direct current electrical grids. The multi-objective MPA proposed in [66] selects elite individuals by non-dominated sorting and a niche approach. In [63, 67], two versions of MOMPA based on crowding distance were put forward, however, its performance degrades when the number of objectives is more than two. In our previous work, we also proposed the MOMPA[68], which adopts the reference points based mechanism to select elite offsprings and achieves satisfactory performance in solution convergence and diversity for most multi-objective optimization problems, except for some complex multimodal problems.

According to the no free lunch theorem [77], there should be no any optimization algorithms that can be general-purpose universally-best. So, researchers are encouraged to develop more variety of optimization algorithms. There is a great demand for optimization algorithms to solve complex multi-objective optimization problems in practical engineering applications. These multi-objective optimization problems typically require optimization algorithms with excellent global and local search capabilities, uniform and complete solution distribution, and to be quite effective in dealing with multimodal objective functions. In view of the above, we developed CMOMPA, which uses a competitive mechanism based on learning strategies to generate offspring and utilizes a reference point-based strategy for individual selection. It has the attractive features of fast and accurate convergence and well distributed solutions.

3 System model and problem formulation

In this section, we firstly present the coverage model of a WSN, the communication model of an SN, and the overall deployment cost model. After that, we formulate the multi-objective optimization deployment problem based on the above three models.

3.1 Coverage model

Consider a deterministic deployment of a WSN. Let \( L = \{ l_1, l_2, \cdots, l_i, \cdots, l_{\text{site}} \} \) represent the set of the potential locations of SNs, where \( l_{\text{site}} \) is the total number of locations. In the 3-D target area \( \mathcal{A} \), the coordinate of \( l_i \) is denoted as \( L_i = (x_i, y_i, z_i) \). Let \( T = \{ t_1, t_2, \cdots, t_j, \cdots, t_{\text{target}} \} \) represent the set of target points to be sensed in \( \mathcal{A} \), where \( t_{\text{target}} \) is the total target points. The area sensed by each SN of type \( v \) is assumed to be a sphere with the radius of \( R_v(v) \), as shown in Fig. 2.

Suppose that there are \( I_{\text{type}} \) types of SNs. We use \( H^v_i, v \in \{ 1, 2, \cdots, I_{\text{type}} \}, i \in \{ 1, 2, \cdots, I_{\text{site}} \} \) to indicate whether the SN of type \( v \) is deployed at the location \( l_i \). If deployed, \( H^v_i = 1 \), otherwise it is 0. Moreover, only one SN can be
Table 2 Review of Literature: variants of multi-objective MPA

| Ref | Offspring generation strategy | Elite selection strategy |
|-----|-----------------------------|--------------------------|
| [62] | MPA | Non-dominated sorting, Fuzzy set method |
| [63] | MPA, Gaussian Mutation | Non-dominated sorting, Crowded distance |
| [64] | MPA | Non-dominated sorting, Fuzzy set method |
| [65] | MPA | Non-dominated sorting, Niche approach |
| [66] | MPA | Non-dominated sorting, Niche approach |
| [67] | MPA | Non-dominated sorting, Crowded distance, |
| [68] | MPA, Gaussian perturbation | Non-dominated sorting, Reference points |
| Ours | MPA, Gaussian perturbation, Competition mechanism based learning strategy | Non-dominated sorting, Reference points |

placed at each potential location, that is, \( \sum_{v=1}^{I_{\text{type}}} H^v_i \leq 1 \) for any given \( l_i \). The deployment solution of a WSN can be denoted by

\[
H = [H_1, \ldots, H_t, \ldots, H_{n_{\text{site}}}],
\]

where \( H_i = [H^1_i, H^2_i, \ldots, H^v_i, \ldots, H^{I_{\text{type}}} i] \).

We adopt the commonly used target point coverage model [51]. Specifically, let \( s(l_i, t, v) \) be the boolean indicator indicating whether the target point \( t \) can be effectively sensed by the SN of type \( v \) located at \( l_i \). \( s(l_i, t, v) \) is determined as follows:

\[
s(l_i, t, v) = \begin{cases} 
1, & \|l_i,t\|_2 \leq R_s(v), \\
0, & \text{others}
\end{cases}
\]

We further define the joint sensing indicator indicating whether \( t \) can be effectively sensed by at least one SN, denoted as \( S(t) \mid L \). Obviously,

\[
S(t) \mid L = 1 - \prod_{l_i \in L} \left[ 1 - \sum_{v=1}^{I_{\text{type}}} s(l_i, t, v) H^v_i \right].
\]

Then, we can obtain the overall coverage rate of \( T \) as follows:

\[
S/T = \frac{1}{n_{\text{target}}} \sum_{j=1}^{n_{\text{target}}} S(t_j) \mid L.
\]

In addition, the total number of SNs that sense \( t_j \) can be calculated as follows:

\[
k_{\text{cov},t_j} = \sum_{l_i \in L} \sum_{v=1}^{I_{\text{type}}} s(l_i, t_j, v) H^v_i.
\]

We take the average of \( k_{\text{cov},t_j} \) over the target points set \( T \) as follows:

\[
\sigma_{\text{cov}} = \frac{1}{n_{\text{target}}} \sum_{j=1}^{n_{\text{target}}} k_{\text{cov},t_j}.
\]
3.3 Cost model

We propose a total cost model for deploying a WSN in a 3-D heterogeneous environment, including node deployment cost, and node manufacturing cost. We use $C_{SN}(v)$, $v = \{1, 2, \cdots, I_{type}\}$ to represent the manufacturing cost of an SN of type $v$, and use $C_{loc}(i)$, $i = \{1, 2, \cdots, I_{site}\}$ to represent the deployment cost of each potential location. The total deployment cost can be calculated as follows:

$$ C_{\Sigma} = \sum_{i=1}^{I_{site}} \sum_{v=1}^{I_{type}} C_{SN}(v) \cdot C_{loc}(i) \cdot H^v_i. $$

(11)

3.4 Problem description

As mentioned above, in this work, we investigate the tradeoff among cost, effectiveness and reliability by jointly optimizing three objectives, namely the overall cost, coverage degree ($K$-coverage) and connection degree ($C$-connectivity), under the constraints of complete coverage of the monitoring area and a connected network. To sum up, the cost-saving deployment problem of WSNs can be formulated as follows:

$$ \min_{H} \{C_{\Sigma}, -\sigma_{cov}, -\sigma_{conn}\}, $$

s.t. $C1: S_{E}|T = 1,$
$$ C2: \mathcal{C}(G_{WSN}) = 1,$$
$$ C3: k_{cov,t} \geq K, \forall t,$$
$$ C4: c_{conn,l} \geq C, \forall l, \sum_{v=1}^{I_{type}} H^v_i = 1,$$
$$ C5: \sum_{i=1}^{I_{site}} H^v_i \leq 1, \forall i,$$
$$ C6: H^v_i \in \{0, 1\}, \forall i, v, $$

(12)

where $H^v_i$ is the decision vector defined in (1) whose search space is the set of vertices of a $I_{type} \times I_{site}$-dimensional unit hypercube, $C1$ ensures that all target points can be covered, $C2$ ensures the full network connectivity, $C3$ indicates that the coverage degree of any target point should be no less than a predefined value $K$, $C4$ indicates that the connection degree of any SN deployed in $l_i$ should be no less than a predefined value $C$, $C5$ ensures that only one SN can be placed at any potential location, and $C6$ indicates that $H^v_i$ must be a binary integer.

4 Methodology

Considering the multi-objective optimization problem formulated in (12) is a non-convex, discontinuous, multimodal and NP-hard problem, we develop the novel CMOMPA (competitive multi-objective MPA) to solve it. In this section, CMOMPA is first introduced in detail, including...
the initialization of the algorithm, iteration procedure, offspring generation strategy and complexity analysis. After that, we further present the CMOMPA-based optimal WSN deployment method.

4.1 CMOMPA

As the name suggests, CMOMPA is a multi-objective variant of the well-known single-objective optimizer MPA [2] that mimics the movement of marine predators and their prey using three phases. In our previous work, we have proposed a multi-objective MPA, known as MOMPA [68]. However, through extensive experiments, we found that the performance of MOMPA degrades in the face of multi-objective optimization problems with a large number of locally optimal solutions. Figure 4 shows the Pareto fronts (PFs) of DTLZ1 and DTLZ3 benchmark functions obtained by MOMPA. We can see that they are far away from the true PFs, which indicates that the algorithm is trapped in the optimal local solution. In order to improve the global search ability, we propose an improved version of MOMPA, named CMOMPA, without increasing the complexity of the algorithm.

4.1.1 Movement strategy

CMOMPA involves two kinds of movements, namely the Lévy flight and Brownian motion. Lévy flight is a random walk in which the random step-lengths follow the Lévy distribution as follows:

\[ R_L = \frac{0.05c}{|b|^{1/\lambda}}, \]

where \( R_L \) is the random step-length; \( c \) and \( b \) are normally distributed, \( c \sim N(0, \sigma_c^2) \), \( b \sim N(0, 1) \), \( \lambda = 1.5 \), and \( \sigma_c \) is given by

\[ \sigma_c = \left[ \frac{\Gamma(1+\lambda) \cdot \sin(\frac{\pi \lambda}{2})}{\Gamma(\frac{1+\lambda}{2}) \cdot \lambda \cdot 2^{\frac{\lambda-1}{2}}} \right]^{1/\lambda}, \]

where \( \Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \) represents the Gamma function. Brownian motion is a Gaussian process with mean zero and variance one.

4.1.2 Initialization

In CMOMPA, we use \( P_0 \) to denote the initial population, which is described as follows:

\[ P_0 = [X_1, X_2, \cdots, X_N]^T, \]

where \( X_i = [X_{i,1}, X_{i,2}, \cdots, X_{i,d}] \) is the \( i \)th individual in \( P_0 \), \( d \) represents the dimension of the problem, and \( N \) is the number of individuals in the population. Each individual represents a feasible solution. The initial solution \( X_i \) is uniformly distributed over the search space as follows:

\[ X_i = X_{\text{min}} + \text{rand}() \cdot (X_{\text{max}} - X_{\text{min}}), \]

where \( X_{\text{min}} \) and \( X_{\text{max}} \) are the lower and upper bounds of the solution space, and \( \text{rand}() \) returns a random number in \((0, 1)\).

After that, we select \( N_a \) solutions in \( P_0 \) by the reference points based method to create the initial archive \( A_0 \) which is defined as follows:

**Definition 1** Archive (A). The archive \( A \) is a set that is used to store the \( N_a \) best solutions obtained in each iteration. The size of \( A \) equals the number of reference points, which can be calculated as

\[ N_a = \binom{m + p - 1}{p}, \]
where \( m \) is the number of optimization objectives, and \( p \) is the number of divisions on each objective in the normalized hyperplane.

Then, we obtain the set of non-dominated individuals of \( A_0 \) by the non-dominated sorting method, which is used to construct the initial predator matrix (\( E_1 \)) whose size is \( N_a \) by Algorithm 1 (\( k=0 \)).

**Algorithm 1** Update \( E_k \) in the \( k \)th iteration.

**Input:** The set of non-dominated individuals of \( A_0 \): \( A_k^{nd} \)  
**Output:** \( E_k \)
1. \( E_k = \left\{ A_k^{nd}, \ldots, A_k^{nd} \right\} \).
2. Repeat \( \left\lfloor \frac{N_a}{\|A_k^{nd}\|_2} \right\rfloor \) times.
3. if \( N_a \) is not an integer multiple of \( \|A_k^{nd}\|_2 \) then
4. Randomly select \( \{N_a - \left\lfloor \frac{N_a}{\|A_k^{nd}\|_2} \right\rfloor \|A_k^{nd}\|_2 \} \) individuals from \( A_k^{nd} \).
5. Add these non-dominated individuals to \( E_k \).
6. return \( E_k \)

### 4.1.3 The iterative process

In CMOMPA, the iterative process is divided into three stages. Specifically, the first stage is from the first iteration to the \( (k_{\text{max}}/3) \)th iteration, where \( k_{\text{max}} \) is the maximum number of iterations. For the \( k \)th \( (1 \leq k < k_{\text{max}}/3) \) iteration, the moving step matrix of the preys, denoted as \( S_k \), and the updated \( P_k \) can be obtained as follows:

\[
S_k = R_L \odot (E_k - R_L \odot A_{k-1}),
\]

\[
P_k = A_{k-1} + \theta \cdot R \odot S_k,
\]

where \( \theta \) is a constant, \( R \) is a random vector in which each element is given by \( \text{rand}() \), \( R_L \) is the Brownian motion vector and \( \odot \) is the dot product operator.

The second stage is from the \( (k_{\text{max}}/3) \)th iteration to the \( (2k_{\text{max}}/3) \)th iteration. In this stage, \( S_k \) and \( P_k \) \( (k_{\text{max}}/3 \leq k < 2k_{\text{max}}/3) \) are updated as follows:

\[
S_k = R_L \odot \left( E_k \left| \frac{N_a}{k_{\text{max}}/3} \right. - R_L \odot A_{k-1} \left| \frac{N_a}{k_{\text{max}}/3} \right. \right).
\]

\[
S_k = R_R \odot \left( R_R \odot E_k \frac{N_a}{k_{\text{max}}/3} - A_{k-1} - \frac{N_a}{k_{\text{max}}/3} + 1 \right),
\]

\[
P_k = A_{k-1} + \theta \cdot R \odot S_k \left| \frac{N_a}{k_{\text{max}}/3} \right. + 1,
\]

\[
P_k = E_k \left| \frac{N_a}{k_{\text{max}}/3} \right. + 1 + \theta \cdot \gamma \odot S_k \left| \frac{N_a}{k_{\text{max}}/3} \right. + 1,
\]

where \( R_L \) is the random step length vector in Lévy flight pattern; \( R_R \) is the random step length vector in Brownian movement pattern; \( \lfloor \cdot \rfloor \) is the floor operator; \( V \left| b \right. \) returns the sub-vector of \( V \) from the \( a \)th element to the \( b \)th element, and \( \gamma \) is an adaptive parameter changing with the increase of \( k \) as follows:

\[
\gamma = \left( 1 - \frac{k}{k_{\text{max}}} \right) \frac{2k}{k_{\text{max}}}. \tag{20}
\]

The last stage is from the \( (2k_{\text{max}}/3) \)th iteration to the end. In this stage, \( S_k \) and \( P_k \) \( (k > 2k_{\text{max}}/3) \) can be obtained as follows:

\[
S_k = R_L \odot (R_L \odot E_k - A_{k-1}),
\]

\[
P_k = E_k + \theta \cdot \gamma \odot S_k. \tag{21}
\]

### 4.1.4 Gaussian elite perturbation

In the \( k \)th iteration, the Gaussian elite perturbation strategy is applied to \( P_k \) to generate another \( N_a \) individuals. We denote this new individual set as \( P_{\text{GEP}} \). For the \( i \)th individual in \( P_k \), select a dimension randomly, namely the \( j \)th dimension, then recalculate the value of \( X_{i,j} \) as follows:

\[
X_{i,j}^{\text{GEP}} = X_{i,j} + (X_{\text{max},j} - X_{\text{min},j}) G \tag{22}
\]

where \( G \sim N(0, 1) \), \( X_{\text{max},j} \) and \( X_{\text{min},j} \) are the upper and lower bounds of the \( j \)-dimensional variables, respectively.

The pseudo-code of the Gaussian elite perturbation is presented in Algorithm 2.

**Algorithm 2** Gaussian elite perturbation.

**Input:** \( P_k \)  
**Output:** \( P_{\text{GEP}} \)
1. for all individuals in \( P_k \) do
2. Randomly select a dimension \( j \), \( j = 1, 2, \ldots, d \).
3. Generate the new individual for \( P_k \) by (22).
4. return \( P_{\text{GEP}} \)

### 4.1.5 Competition mechanism based learning strategy

CMOMPA leverages the idea of the competition mechanism based learning strategy [25] to improve global search ability. The competition mechanism based learning consists of two components, namely pairwise competition and individual learning. Specifically, in the \( k \)th iteration, CMOMPA randomly selects one individual each from \( P_k \) and \( P_{\text{GEP}} \), respectively, and calculates their Fitness values by the shift-based density estimation [79] strategy as follows:

\[
\text{Fitness}(X) = \min_{Y \in P_{\text{GEP}}} \left( \sum_{i=1}^{m} \left( \max(0, f_i(Y) - f_i(X)) \right)^2 \right), \tag{23}
\]
where $X$ is an individual in the population $P$, $f_i(X)$ denotes the $i$th objective value of $X$ and $m$ denotes the number of objectives. The shift-based density estimation strategy can evaluate the quality of a solution in terms of convergence and diversity.

After that, the fitness values of these two individuals are compared. The individual with a worse fitness value is a loser and is recorded as $X_l$. It is updated according to the individual with a better fitness, denoted as $X_w$, as follows.

$$X_l = X_l + \eta (X_w - X_l)$$

(24)

where $\eta$ is a uniformly random value in $[0, 1]$. Besides, to further improve the performance of CMOMPA in exploration and exploitation capability, both $X_l$ and the updated $X_w$ are slightly mutated by polynomial mutation [80].

This procedure repeats until all individuals in $P_k$ and $P_{GEP}$ are updated. Algorithm 3 depicts the procedure of the competition mechanism based learning strategy.

**Algorithm 3** Competition mechanism based learning strategy.

**Input:** $P$, $P_{GEP}$

**Output:** $P_{CML}$

1. Fitness $\leftarrow$ Calculate the Fitness of each individual in $P$ and $P_{GEP}$ by (23).
2. $P_{CML} \leftarrow \emptyset$.
3. while $P \neq \emptyset$ and $P_{GEP} \neq \emptyset$ do
   4. Randomly select an individual $p$ from $P$.
   5. Randomly select an individual $q$ from $P_{GEP}$.
   6. $P \leftarrow P \setminus p$, $P_{GEP} \leftarrow P_{GEP} \setminus q$.
   7. if Fitness $(p) <$ Fitness $(q)$ then
      8. $X' \leftarrow p$.
      9. $X^w \leftarrow q$.
   else
      10. $X' \leftarrow q$.
      11. $X^w \leftarrow p$.
   12. Update $X'$ by learning from $X^w$ by (24).
   13. Mutate $X'$ and $X^w$ by polynomial mutation.
   14. $P_{CML} \leftarrow P_{CML} \cup \{X', X^w\}$
15. return $P_{CML}$

**4.1.6 Elite selection**

**Definition 2** File ($Q$).

The file $Q$ is a set that is used to store all individuals generated in each iteration. Specifically, in the $k$th iteration, $Q_k = A_{k-1} \cup P_k \cup P_{GEP}^{k} \cup P_{CML}^{k}$. Individuals in $Q$ are candidates for elite selection.

**Definition 3** Dominate.

An individual $X$ is said to dominate another one $X'$ if both of the following conditions are true, 1) $X$ is no worse than $X'$ in all objectives, and 2) $X$ is strictly better than $X'$ in at least one objective. $X$ dominating $X'$ is denoted as $X \succ X'$.

In the $k$th iteration, we obtain $Q_k$ containing $5N_a$ individuals. After that, we need to select $N_a$ individuals among $Q_k$ to generate $A_k$. Firstly, we use the non-dominated sorting method to divide $Q_k$ into several levels, namely $L_1, L_2, \ldots$. For two different levels, namely $L_i$ and $L_j$ ($i < j$), any individual in $L_i$ dominates all individuals in $L_j$, whereas individuals in the same level do not dominate each other. Then, all the individuals in $\bigcup_{i=1}^{j} L_i$ ($l \geq 1$) are added to $A_k$ where $l$ satisfies the following conditions: 1) $\left|\bigcup_{i=1}^{l} L_i\right| \leq N_a$; and 2) $\left|\bigcup_{i=1}^{l+1} L_i\right| > N_a$. If $\left|\bigcup_{i=1}^{l} L_i\right|$ equals to $N_a$, then $A_k = \bigcup_{i=1}^{l} L_i$. Otherwise, the first $N_a - \left|\bigcup_{i=1}^{l} L_i\right|$ best individuals in $L_{l+1}$ are further added to $A_k$. In this case, the reference point-based method proposed in [26] is employed to choose elite individuals in $L_{l+1}$. The pseudo-code of elite selection is presented in Algorithm 4.

**Algorithm 4** Elite selection method.

**Input:** $Q$

**Output:** $A$

1: $l \leftarrow 1$.
2: while $|Q| > 0$ do
   3: for all $X_i \in Q$ do
      4: if $\exists X_j > X_i (j \neq i)$ then
         5: $L_i \leftarrow L_i \cup X_i$.
   6: $Q \leftarrow Q - L_i$.
   7: $l \leftarrow l + 1$.
8: $A \leftarrow \emptyset$, $l \leftarrow 1$.
9: while $|A \cup L_i| \leq N_a$ do
10: $A \leftarrow A \cup L_i$.
11: $l \leftarrow l + 1$.
12: if $|A| < N_a$ then
13: Sort all individuals in $L_i$ in descending order by reference point based method.
14: $A \leftarrow A \cup L_i(1 : N_a - |A|)$.
15: return $A$.

**4.1.7 CMOMPA and computational complexity**

The overall procedure of CMOMPA is described as follows.

**Initialization:**

Step1: Initialize $P_0$ by (16), $Q_1 \leftarrow \emptyset$. Select $N_a$ solutions in $P_0$ by the reference points based method to create $A_0$.

Step2: Construct the initial predator matrix $E_1$ by Algorithm 1.

**Iterations** ($k \geq 1$):

Step3: In the $k$th iteration, generate $P_k$ by (18), (19), or (21), accordingly.
Step4: Generate $P_{k}^{GEP}$ by Algorithm 2.
Step5: Generate $P_{k}^{CML}$ by Algorithm 3.
Step6: Generate $Q_{k} = A_{k-1} \cup P_{k} \cup P_{k}^{GEP} \cup P_{k}^{CML}$.
Step7: Generate $A_{k}$ by Algorithm 4, $k = k + 1$.
Step8: Update $E_{k}$ by Algorithm 1.
Step9: Repeat the loop from Step 3 to 8 until the maximum number of iterations is reached.

Algorithm 5 describes the detailed algorithm of CMOMPA.

**Algorithm 5 CMOMPA procedure.**

1. $k \leftarrow 0$, initialize $P_{k}$ by (16).
2. Select $N_{a}$ solutions in $P_{k}$ to construct $A_{k}$ by the reference points based method.
3. Initialize $E_{1}$ by Algorithm 1.
4. for $k = 1$ to $k_{max}$ do
5. if $k < k_{max}/3$ then
6. Update $P_{k}$ by (18).
7. else if $k \leq 2k_{max}/3$ then
8. Update $P_{k}$ by (19).
9. else
10. Update $P_{k}$ by (21).
11. Generate $P_{k}^{GEP}$ by Algorithm 2.
12. Generate $P_{k}^{CML}$ by Algorithm 3.
13. $Q_{k} \leftarrow A_{k-1} \cup P_{k} \cup P_{k}^{GEP} \cup P_{k}^{CML}$.
14. Generate $A_{k}$ by Algorithm 4, $k = k + 1$.
15. Update $E_{k}$ by Algorithm 1.
16. return Approximate Pareto optimal solutions $A_{k}$.

The complexity of CMOMPA is within our acceptable range. We use $n$ to represent the size of the population, $d$ to represent the dimension of the decision variable, and $m$ to represent the number of objectives. The three-stage evolutionary complexity of the population is $O(n \times d)$. Gaussian disturbance complexity is $O(n)$. The complexity of non-dominated sorting and solution selection based on reference point is $O(m \times n^{2})$. The complexity of the competition mechanism based learning strategy is $O(n)$. The complexity of the polynomial mutation is $O(n)$. We use $cof_{1}$ to represent the computational complexity of the benchmark function, so the computational complexity of CMOMPA is $O(k_{max}(n \times d + n + cof_{1} \times n + m \times n^{2}))$.

### 4.2 CMOMPA based optimal deployment

#### 4.2.1 Application model

The CMOMPA has been discussed in detail above. Next, the optimal deployment method based on CMOMPA is presented in this subsection.

Considering $H_{i}^{v} \in [0, 1], i \in \{1, 2, \ldots, \llbracket \text{size} \rrbracket \}, v \in \{1, 2, \ldots, \llbracket \text{type} \rrbracket \}$, is a binary integer. To solve the solution of (12), $H_{i}^{v}$ is relaxed as a real number $\tilde{H}_{i}^{v} \in [0, 1]$. Concretely, $H_{i}^{v} \rightarrow \tilde{H}_{i}^{v} \in [0, 1]$ with the constraint of $\tilde{H}_{i}^{v} - (\tilde{H}_{i}^{v})^{2} \leq \varepsilon$, where $\varepsilon$ is a sufficiently small number.

Consequently, the search space of $\tilde{H}_{i}^{v}$ is also extended to the entire $\llbracket \text{type} \rrbracket \times \llbracket \text{site} \rrbracket$-dimensional unit hypercube and the problem formulated in (12) can be transformed as follows:

$$\min_{\tilde{H}} \{C_{\Sigma} - \sigma_{\text{cov}}, -\sigma_{\text{conn}}\},$$

s.t. $C_{1} : S_{\Sigma} \mid T = 1,$
   $C_{2} : \mathcal{C}(G_{\text{WSN}}) = 1,$
   $C_{3} : k_{\text{conn},ij} \geq K, \forall i,j,$
   $C_{4} : c_{\text{conn},ij} \geq C, \forall i, \text{round} \left( \sum_{v=1}^{\llbracket \text{type} \rrbracket} H_{i}^{v} \right) = 1,$
   $C_{5} : \sum_{v=1}^{\llbracket \text{type} \rrbracket} H_{i}^{v} \leq 1, \forall i,$
   $C_{6} : \tilde{H}_{i}^{v} \in [0, 1], \forall i, v,$
   $C_{7} : \tilde{H}_{i}^{v} - (\tilde{H}_{i}^{v})^{2} \leq \varepsilon, \forall i, v,$

where $\tilde{H} = \{\tilde{H}_{1}, \ldots, \tilde{H}_{i}, \ldots, \tilde{H}_{\llbracket \text{site} \rrbracket} \}$ is the relaxed $\mathbf{H}$, round($\cdot$) is a rounding operation. It is worth noting that the main difference between (12) and (25) is that (25) relaxes the discrete variables $H_{i}^{v}$ into continuous variables $\tilde{H}_{i}^{v}$ subject to $C_{6}$ of $C_{7}$.

#### 4.2.2 Constraint processing

The problem optimal deployment problem formulated in (25) is a multi-objective optimization problem with constraints. Consequently, the constrained-domination should be used to judge the dominant relationship between solutions during elite individual selection presented in Algorithm 4, instead of the unconstrained domination. The constrained-domination is defined as follows:

**Definition 4** Feasible and Infeasible solution.

An solution $\tilde{H} = \{\tilde{H}_{1}, \ldots, \tilde{H}_{i}, \ldots, \tilde{H}_{\llbracket \text{site} \rrbracket} \}$ is said to be a feasible solution of the optimal deployment problem formulated in (25) if it satisfies the constraints $C_{1} - C_{7}$. Otherwise, it is an infeasible solution.

**Definition 5** Constrained-Dominate.

A solution $\tilde{H} = \{\tilde{H}_{1}, \ldots, \tilde{H}_{i}, \ldots, \tilde{H}_{\llbracket \text{site} \rrbracket} \}$ is said to constrained-dominate another solution $\tilde{H}' = \{\tilde{H}'_{1}, \ldots, \tilde{H}'_{i}, \ldots, \tilde{H}'_{\llbracket \text{site} \rrbracket} \}$ if any of the following conditions is true, 1) both are feasible solutions and $\tilde{H} \succ \tilde{H}'$, 2) both are infeasible solutions and $\tilde{H}$ has a smaller constraint violation, 3) $\tilde{H}$ is feasible and $\tilde{H}'$ is infeasible, where the constraint violation ($\Delta$) of an individual $\tilde{H}$ can be calculated as follows:

$$\Delta(\tilde{H}) = \langle 1 - S_{\Sigma} | T \rangle + (1 - \mathcal{C}(G_{\text{WSN}})) + (\tilde{H}_{i}^{v} - 1) + \left( \tilde{H}_{i}^{v} - (\tilde{H}_{i}^{v})^{2} - \varepsilon \right),$$

$$\tilde{H}_{i}^{v} - (\tilde{H}_{i}^{v})^{2} \leq \varepsilon.$$
where \( \langle \alpha \rangle \) returns a zero if \( \alpha \leq 0 \), otherwise it returns \( |\alpha| \).

### 4.2.3 The process of the optimal SNs deployment method

The cost-saving optimal deployment method for WSNs based on CMOMPA is presented as follows.

- **Initialization:**
  1. Initialize the deployment area and all parameters, \( k \leftarrow 0, N \leftarrow 100 \). Randomly generate \( N \) individuals (feasible solutions) to create \( \text{P}_k \), as modeled in (16). The \( i \)th potential deployment solution. Specifically, \( \text{H}_i \) is defined as follows:

\[
\text{H}_i = \left[ \tilde{H}_1, \ldots, \tilde{H}_i, \ldots, \tilde{H}_{\text{max}} \right]
\]

where \( \tilde{H}_j = \left[ H_j^1, H_j^2, \ldots, H_j^i, \ldots, H_j^{\text{max}} \right] \).

  2. Create \( \text{A}_k \) by selecting \( N_{\alpha} \) individuals by the reference points based method and construct \( \text{E}_i \) by Algorithm 1.

- **Iterations** \( (k \geq 1) \):
  1. In the \( k \)th iteration, generate \( \text{P}_k \) by (18), (19), or (21), accordingly.
  2. Generate \( \text{P}_k \) by Algorithm 2.
  3. Generate \( \text{P}_k \) by Algorithm 3.
  4. Update \( \text{Q}_k = \text{A}_{k-1} \cup \text{P}_k \cup \text{P}_k^\text{GEP} \cup \text{P}_k^\text{CML} \).
  5. Update \( \text{A}_k \) by the constrained-domination version of Algorithm 4, \( k = k + 1 \).
  6. Update \( \text{E}_k \) by Algorithm 1.
  7. Repeat the loop from Step 3 to 8 until the maximum number of iterations is reached.
  8. The individuals in the final \( \text{A} \) are the obtain approximate Pareto optimal deployment solutions.

Figure 5 shows the flow chart of the CMOMPA based optimal SNs deployment method.

### 5 Performance evolution

The previous section has introduced the proposed CMOMPA and the cost-saving SNs deployment method based on it. In this section, we present the performance evaluations and discuss the results. Specifically, the experimental setup, the benchmark functions, the comparison algorithms and the performance metrics are firstly introduced. Then, CMOMPA and the WSN deployment method based on it are extensively evaluated, respectively. The experimental results are discussed in detail.

#### 5.1 Experimental setup

The performances of the proposed CMOMPA and the cost-saving SNs deployment based on it were extensively evaluated in this article. The algorithms and deployment methods were implemented in MATLAB 2020b. The performance evaluations were conducted on a server with an Intel(R) Xeon (R) E5-2620 3.0 GHz CPU, 64 GB RAM, a Windows Server 2019 operating system.

### 5.2 Benchmark function and comparison algorithms

In this paper, 20 popular benchmark functions selected from ZDT, WFG and DTLZ benchmark suites were used to evaluate the performance of CMOMPA. The detailed descriptions of these benchmark functions are listed in Table 3. These benchmark functions have different features and can comprehensively test the performance of the algorithms. Specifically, some multimodal functions, e.g., ZDT3-6, etc., require that the algorithms must have strong global convergence ability. The functions with irregular PFs, e.g., DTLZ5-7, impose great challenges upon the diversity of algorithm. The functions in WFG suite are scaled to different ranges. These functions have high requirements for both global convergence and solution distribution of the algorithm. The maximum number of iterations for functions in the ZDT suite is 500, and that for the other suites is 3000. The population size is 100.

We compared CMOMPA with the following ten state-of-the-art multi-objective optimization algorithms, namely NSGA-II [69], non-dominated sorting genetic algorithm III (NSGA-III) [26], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [70], Pareto envelope based selection II (PESA-II) [81], competitive multi-objective particle swarm optimization (CMOPSO) [82], non-dominated sorting and local search (NSLS) [83], coevolutionary for constrained multi-objective optimization (CCMO) [84], constrained multi-objective evolutionary algorithms with a two-stage (CMOEA-MS) [85], dynamically constrained NSGA-III (DCNSGA-III) [86] and MOMPA [68]. All the compared algorithms are implemented in PlatEMO [87], except for MOMPA. The detailed parameters of the algorithms are shown in Table 4. The above parameters are either exactly recommended by its developers or in the range of the recommendations to have the best performance for each algorithm.

#### 5.3 Performance metric

In our experimental comparison, we used the inverted generational distance (IGD) [88] and the hypervolume (HV) [89] to evaluate the performance of different algorithms. The performance of the algorithm can be fully measured by the IGD and HV metrics [90, 91].

The inverted generational distance (IGD): Assuming that the approximate non-dominated solution set obtained by the multi-objective optimization algorithm is \( \text{P}_d \), and the
reference points set sampled along the true PF is $P_t$. The value of IGD can be calculated as follows.

$$\text{IGD}(P_a, P_t) = \frac{\sum_{\nu \in P_a} \min_{\chi \in P_t} \|\nu, \chi\|_2}{|P_t|},$$  

(28)

Table 3 Detailed descriptions of benchmark functions

| Name  | Features                  | Objectives | Dimension |
|-------|---------------------------|------------|-----------|
| ZDT1  | Unimodal, convex          | 2          | 30        |
| ZDT2  | Unimodal, concave         | 2          | 30        |
| ZDT3  | Multimodal, disconnected  | 2          | 30        |
| ZDT4  | Multimodal, convex        | 2          | 10        |
| ZDT6  | Multimodal, concave       | 2          | 10        |
| DTLZ1 | Linear, multimodal        | 3          | 7         |
| DTLZ2 | Concave                   | 3          | 12        |
| DTLZ3 | Concave, multimodal       | 3          | 12        |
| DTLZ4 | Concave, biased           | 3          | 12        |
| DTLZ5 | Concave, Degenerate       | 3          | 12        |
| DTLZ6 | Concave, Degenerate, Biased| 3        | 12        |
| DTLZ7 | Multimodal, Mixed, Disconnected | 3   | 22        |
| WFG2  | Multimodal, nonseparable, convex | 3   | 12        |
| WFG3  | Linear, degenerate        | 3          | 12        |
| WFG4  | Multimodal, concave       | 3          | 12        |
| WFG5  | Concave, deceptive        | 3          | 12        |
| WFG6  | Concave, nonseparable     | 3          | 12        |
| WFG7  | Concave, biased           | 3          | 12        |
| WFG8  | Concave, biased, nonseparable | 3        | 12        |
| WFG9  | Concave, biased, multimodal, deceptive | 3   | 12        |

where $|P_t|$ is the size of the set $P_t$ and $\|\cdot\|$ returns the Euclidean distance. Intuitively, the smaller the IGD is, the better performance is achieved. Moreover, in this paper, we use the method reported in [87] to sample reference points.

Hypervolume (HV): Let $z = (z_1, z_2, \cdots, z_m)$ be a reference point, where $m$ denotes the number of objectives. Then, HV can be defined as follows.

$$\text{HV}(P_a) = \text{Leb} \left( \bigcup_{\mathbf{X} \in P_a} [f_1(X), z_1] \times \cdots \times [f_m(X), z_m] \right),$$  

(29)

where $\text{Leb}(\cdot)$ denotes the Lebesgue measures. A higher value of HV indicates a better overall performance.

All experiments were run 30 times, independently, and the average value (Avg.) and standard deviation (Std.) of the IGD and HV were used as the final performance metric.

5.4 Performance evaluation of CMOMPA

Tables 5 and 6 show the comparison results on IGD and HV, respectively, among CMOMPA and other multi-objective optimization algorithms. We also rank the algorithms for each benchmark function in these two tables. It can be observed that CMOMPA is considerably promising. For IGD, CMOMPA ranks top three in 14 out of all 20 benchmark functions, namely four functions in ZDT and DTLZ suites, respectively, and six in WFG suite. For HV, it also shows strong competitiveness. In addition, Table 7 summarizes the total scores and the final rankings in terms of the integrated IGD and HV metrics. An algorithm’s score
Table 4 The parameters of comparative algorithms

| Algorithm        | Parameters                                                                 | Value |
|------------------|-----------------------------------------------------------------------------|-------|
| NSGA-II, NSGA-III| Crossover probability                                                      | 1     |
| DCNSGA-III, CCMO | Mutation probability 1/D                                                   |       |
| CMOEA-MS         | Distribution indexes for SBX                                               | 20    |
| PESA-II, MOEA/D  | Distribution indexes for polynomial mutation                              | 20    |
| MOEA/D           | Penalty parameter of the PBI function                                      | 5     |
|                  | Neighbourhood size Pop-size/10^2                                           |       |
| CMOPSO           | Size of elite particle set                                                | 10    |
| NSLS             | Local Search rates                                                         |       |
|                  | θ                                                                           | 0.5   |
|                  | FA                                                                         | 0.2   |
| MOMPAS           | θ                                                                           | 0.5   |
|                  | Distribution indexes for polynomial mutation                              | 20    |
| DCNSGA-III       | cp                                                                         | 5     |
| CMOEA-MS         | λ                                                                           | 0.5   |

1 D denotes the dimensionality of the problem decision variable
2 Pop-size denotes population size

is derived from its ranking on each benchmark function (as shown in Tables 5 and 6). Concretely, the first-ranked algorithm gets one point, the second-ranked algorithm gets two points, and so on. Finally, we rank the algorithms according to the total scores. It can be found that CMOMPA also ranks the first in total score with 169 points.

The reasons for these results can be interpreted twofold. Firstly, benefiting from the competition mechanism-based learning strategy, CMOMPA has a considerably high global and local search performance. This means CMOMPA outperforms MOMPAS, which ranks fifth in total score and adopts the same elite selecting method as CMOMPA. Secondly, the results indicate that for DTLZ1 and DTLZ3, CMOMPA can jump out of the optimal local solution and approach the true PF. However, for benchmark functions with irregular PFs, such as DTLZ5-7, the performance of CMOMPA algorithm degrades. This may be because the solutions selection method based on uniformly distributed reference points is not very suitable for such problems. CMOMPA also shows unsatisfactory performance in WFG3 and WFG6. Since the PF of WFG3 is a degenerated curve, competition mechanism based on learning strategy may generate too many uncertain solutions which result in poor performance. WFG6 is a nonseparable-reduced problem. NSLS has strong local exploration ability and can deal with WFG6 well. Compared with NSLS, CMOMPA's local exploration capability needs to be further strengthened to more effectively handle such problems.

The Wilcoxon signed-rank test method [92] was taken to evaluate the significance of the involved algorithms. From the Wilcoxon signed-rank test results of the IGD shown in Table 8, we can see that most of the p-values obtained by comparing CMOMPA with other multi-objective optimization algorithms are much less than 0.05, which demonstrates significant differences between them in a statistical sense. In addition, in the last row, the number of plus signs, minus signs and equal signs represent the number of benchmark functions in which CMOMPA is superior to, inferior to and non-significantly different from the counterpart multi-objective optimization algorithms, respectively. It can be seen that there are always more plus signs than minus signs, which indicates that CMOMPA has a significant statistical superiority compared to other algorithms.

To investigate the sensitivity of the tuning probability θ which can attenuate/magnify the individual optimization step in CMOMPA, we separately tested the convergence performance of CMOMPA with different values of θ, i.e., θ = {0.1, 0.3, 0.5, 0.7, 0.9} on ZDT6, DTLZ1, WFG7. Other parameter settings are consistent with the previous ones. The results were averaged from 30 independent runs. As can be seen from Fig. 6, for three different functions, the IGD value of CMOMPA finally converges to the same level, which indicates that CMOMPA is not sensitive to the parameter θ.

5.5 Performance evaluation of the CMOMPA based deployment method

Suppose a 55 × 55 × 20 m³ three-storey factory area, as shown in Fig. 7, the location set L where nodes can be deployed is known. The deployment cost of each candidate
Table 5 Comparisons on computational results among CMOMPA and other multi-objective optimization algorithms about the IGD metric (The best results are shown in bold.)

| Algorithm | ZDT1       | ZDT2       | ZDT3       | ZDT4       |
|-----------|------------|------------|------------|------------|
| NSGA-II   | 4.82E-03   | 1.75E-04   | 4.86E-03   | 1.94E-04   |
| NSGA-III  | 3.91E-03   | 1.20E-05   | 3.86E-03   | 2.78E-05   |
| MOEA/D    | 1.19E-02   | 8.49E-03   | 2.60E-02   | 2.70E-02   |
| PESA-II   | 1.15E-02   | 3.44E-03   | 1.14E-02   | 1.85E-03   |
| CMOPSO    | 4.19E-03   | 8.96E-05   | 4.13E-03   | 8.80E-05   |
| NSLS      | 2.34E-01   | 4.24E-02   | 4.05E-01   | 5.69E-02   |
| CCMO      | 3.96E-03   | 7.11E-05   | 3.95E-03   | 6.06E-05   |
| CMOEA-MS  | 4.85E-03   | 5.09E-05   | 3.97E-03   | 9.50E-05   |
| MOEA/D    | 1.08E-03   | 2.52E-05   | 3.83E-03   | 1.65E-05   |
| MOMPA     | 3.90E-03   | 7.58E-05   | 3.80E-03   | 8.92E-06   |
| CMOMPA    | 2.99E-03   | 8.01E-06   | 2.14E-01   | 2.34E-03   |

| Algorithm | ZDT6       | WFG2       | WFG3       | WFG4       |
|-----------|------------|------------|------------|------------|
| NSGA-II   | 3.72E-03   | 1.75E-04   | 3.80E-03   | 1.14E-04   |
| NSGA-III  | 3.21E-03   | 1.20E-05   | 3.80E-03   | 2.78E-05   |
| MOEA/D    | 7.00E-03   | 4.24E-03   | 1.93E-01   | 1.85E-03   |
| CCMO      | 3.15E-03   | 7.51E-05   | 1.28E-01   | 6.33E-03   |
| DCNSGA-III| 3.90E-03   | 7.52E-05   | 2.03E-01   | 1.35E-02   |
| MOMPA     | 3.00E-03   | 1.01E-05   | 1.57E-01   | 2.34E-03   |
| CMOMPA    | 2.99E-03   | 8.01E-06   | 2.14E-01   | 2.34E-03   |

| Algorithm | WFG5       | WFG6       | WFG7       | WFG8       |
|-----------|------------|------------|------------|------------|
| NSGA-II   | 2.80E-01   | 9.50E-03   | 1.80E-03   | 1.10E-03   |
| NSGA-III  | 2.30E-01   | 9.22E-06   | 1.70E-03   | 1.34E-03   |
| MOEA/D    | 2.47E-01   | 7.16E-03   | 2.68E-01   | 1.11E-02   |
| PESA-II   | 2.78E-01   | 9.47E-03   | 3.11E-01   | 1.58E-02   |
| CMOPSO    | 2.50E-01   | 5.05E-03   | 2.37E-01   | 4.58E-03   |
| NSLS      | 6.23E-03   | 1.87E-03   | 1.74E-01   | 4.18E-03   |
| CCMO      | 3.15E-03   | 7.51E-05   | 1.75E-01   | 5.23E-03   |
| CMOEA-MS  | 3.17E-03   | 5.92E-05   | 2.03E-01   | 1.35E-02   |
| DCNSGA-III| 3.37E-03   | 4.23E-04   | 1.65E-01   | 1.23E-02   |
| MOMPA     | 3.00E-03   | 1.01E-05   | 1.57E-01   | 2.34E-03   |
| CMOMPA    | 2.99E-03   | 8.01E-06   | 1.58E-01   | 2.28E-03   |

| Algorithm | WFG9       | DTLZ1      | DTLZ2      | DTLZ3      |
|-----------|------------|------------|------------|------------|
| NSGA-II   | 2.76E-01   | 1.39E-02   | 2.74E-02   | 1.34E-03   |
| NSGA-III  | 2.21E-01   | 5.77E-04   | 2.06E-02   | 2.71E-06   |
| MOEA/D    | 2.48E-01   | 2.03E-02   | 2.06E-02   | 6.79E-07   |
| PESA-II   | 2.76E-01   | 2.69E-02   | 2.47E-02   | 1.43E-03   |
| CMOPSO    | 2.19E-01   | 3.35E-04   | 2.07E-02   | 3.74E-04   |
| NSLS      | 2.40E-01   | 5.68E-03   | 2.39E-01   | 1.69E-01   |
| CCMO      | 2.16E-01   | 4.08E-03   | 2.01E-02   | 1.28E-04   |
| CMOEA-MS  | 2.46E-01   | 2.18E-02   | 2.09E-02   | 3.04E-04   |
| DCNSGA-III| 2.25E-01   | 2.19E-02   | 2.06E-02   | 2.85E-06   |

9163
Table 5 (continued)

| Algorithm  | DTLZ4 | DTLZ5 | DTLZ6 | DTLZ7 |
|------------|-------|-------|-------|-------|
| NSGA-II    | 6.82E-02 2.58E-03 | 4       | 5.85E-03 3.16E-04 | 5       |
| NSGA-III   | 1.20E-01 1.68E-01 | 7       | 1.22E-02 1.57E-03 | 8       |
| MOEA/D     | 5.45E-02 7.92E-06 | 3       | 3.39E-02 2.11E-06 | 5       |
| PESA-II    | 9.44E-02 1.61E-01 | 6       | 1.22E-02 3.03E-03 | 7       |
| CMOPSO     | 2.09E-01 3.35E-01 | 11      | 5.21E-03 4.95E-04 | 3       |
| NSLS       | 1.54E-01 1.09E-01 | 9       | 4.50E-03 6.88E-05 | 3       |
| CCMO       | 1.65E-01 2.36E-01 | 10      | 4.34E-03 8.17E-05 | 2       |
| CMOEA-MS   | 8.66E-02 1.23E-01 | 5       | 4.31E-03 9.15E-05 | 1       |
| DCNSGA-III | 1.36E-01 1.85E-01 | 8       | 1.19E-02 2.62E-03 | 8       |

Table 6 Comparisons on computational results among CMOPMA and other multi-objective optimization algorithms about the HV metric (The best results are shown in bold.)

| Algorithm  | ZDT1 | ZDT2 | ZDT3 | ZDT4 |
|------------|-----|-----|-----|-----|
| NSGA-II    | 0.719120 0.000266 | 7   | 0.443780 0.000266 | 8   |
| NSGA-III   | 0.720090 0.000070 | 4   | 0.444780 0.000114 | 3   |
| MOEA/D     | 0.713050 0.004180 | 9   | 0.404350 0.004880 | 10  |
| PESA-II    | 0.711580 0.002160 | 10  | 0.432750 0.004640 | 9   |
| CMOPSO     | 0.719300 0.000267 | 6   | 0.444000 0.000250 | 7   |
| NSLS       | 0.406270 0.012600 | 11  | 0.055655 0.026500 | 11  |
| CCMO       | 0.720080 0.000109 | 5   | 0.444700 0.000099 | 5   |
| CMOEA-MS   | 0.718550 0.000352 | 8   | 0.444670 0.000143 | 6   |
| DCNSGA-III | 0.720100 0.000069 | 2   | 0.444750 0.000098 | 4   |
| MOMPA      | 0.720100 0.000049 | 1   | 0.444800 0.000050 | 2   |
| CMOPMA     | 0.720100 0.000095 | 3   | 0.444900 0.000031 | 1   |

1 The Avg. of CMOPMA, MOMPA, DCNSGA-III are specified as 0.003900, 0.003900, 0.003903
Table 6 (continued)

| Algorithm     | WFG5 | WFG6 | WFG7 | WFG8 |
|---------------|------|------|------|------|
| NSGA-II       | 0.493290 | 0.004160 | 9 | 0.521400 | 0.003970 | 9 | 0.444410 | 0.002790 | 9 |
| NSGA-III      | 0.518470 | 0.000060 | 2 | 0.508340 | 0.012600 | 3 | 0.559380 | 0.000060 | 2 |
| MOEA/D        | 0.506420 | 0.004740 | 7 | 0.504830 | 0.014800 | 5 | 0.540810 | 0.002130 | 6 |
| PESA-II       | 0.470630 | 0.006490 | 10 | 0.456900 | 0.015900 | 11 | 0.483430 | 0.000100 | 9 |
| CMOPSO        | 0.478590 | 0.004850 | 10 | 0.522890 | 0.003280 | 2 | 0.525330 | 0.0003220 | 8 |
| NSLS          | 0.507120 | 0.001160 | 6 | 0.545720 | 0.005990 | 1 | 0.498660 | 0.005100 | 10 |
| CCMO          | 0.513740 | 0.001590 | 5 | 0.498200 | 0.011500 | 6 | 0.536300 | 0.001990 | 5 |
| CMOEA-MS      | 0.504470 | 0.002170 | 8 | 0.496590 | 0.013200 | 8 | 0.535470 | 0.003220 | 8 |
| DCNSGA-III    | 0.518470 | 0.000067 | 1 | 0.507160 | 0.010900 | 4 | 0.559380 | 0.000048 | 1 |
| MOMPA         | 0.518100 | 0.000109 | 4 | 0.478300 | 0.006920 | 9 | 0.524000 | 0.000617 | 4 |
| CMOPMA        | 0.518300 | 0.000262 | 3 | 0.498400 | 0.006950 | 9 | 0.551000 | 0.000530 | 3 |
| NSGA-II       | 0.505850 | 0.006110 | 9 | 0.824360 | 0.003310 | 10 | 0.533300 | 0.003630 | 10 |
| NSGA-III      | 0.545950 | 0.001400 | 3 | 0.841720 | 0.000025 | 3 | 0.559610 | 0.000014 | 4 |
| MOEA/D        | 0.514900 | 0.024100 | 7 | 0.841720 | 0.000026 | 4 | 0.559620 | 0.000014 | 2 |
| PESA-II       | 0.482460 | 0.005900 | 11 | 0.839710 | 0.000127 | 6 | 0.541570 | 0.003010 | 9 |
| CMOPSO        | 0.519540 | 0.003450 | 5 | 0.498200 | 0.013200 | 8 | 0.535470 | 0.003220 | 8 |
| NSLS          | 0.541370 | 0.0003460 | 10 | 0.358150 | 0.328000 | 11 | 0.561700 | 0.000601 | 1 |
| CCMO          | 0.517880 | 0.004680 | 6 | 0.842580 | 0.000289 | 1 | 0.555090 | 0.001130 | 7 |
| CMOEA-MS      | 0.510630 | 0.007010 | 8 | 0.839610 | 0.001180 | 8 | 0.556470 | 0.000096 | 6 |
| DCNSGA-III    | 0.544940 | 0.002440 | 4 | 0.841730 | 0.000008 | 2 | 0.559620 | 0.000005 | 3 |
| MOMPA         | 0.549500 | 0.001300 | 2 | 0.832700 | 0.020000 | 8 | 0.558100 | 0.000209 | 5 |
| CMOPMA        | 0.549500 | 0.000839 | 1 | 0.841000 | 0.002580 | 5 | 0.554000 | 0.004400 | 8 |
| NSGA-II       | 0.505170 | 0.113000 | 10 | 0.199120 | 0.000226 | 4 | 0.199460 | 0.000115 | 5 |
| NSGA-III      | 0.537400 | 0.067300 | 5 | 0.193940 | 0.001270 | 6 | 0.191140 | 0.001560 | 8 |
| MOEA/D        | 0.545070 | 0.055400 | 3 | 0.181850 | 0.000001 | 11 | 0.181850 | 0.000000 | 11 |
| PESA-II       | 0.537270 | 0.004510 | 6 | 0.192430 | 0.003400 | 8 | 0.190560 | 0.005580 | 9 |
| CMOPSO        | 0.488950 | 0.135000 | 11 | 0.198710 | 0.003373 | 5 | 0.200180 | 0.000040 | 1 |
| NSLS          | 0.541370 | 0.038400 | 4 | 0.199850 | 0.000073 | 1 | 0.199890 | 0.000088 | 4 |
| CCMO          | 0.519960 | 0.078600 | 9 | 0.199680 | 0.001060 | 3 | 0.200060 | 0.000038 | 2 |
| CMOEA-MS      | 0.534630 | 0.063500 | 7 | 0.199700 | 0.000112 | 2 | 0.200060 | 0.000039 | 3 |
| DCNSGA-III    | 0.529360 | 0.099600 | 8 | 0.193860 | 0.001270 | 7 | 0.191290 | 0.001420 | 7 |
| MOMPA         | 0.557400 | 0.004062 | 1 | 0.191300 | 0.001800 | 10 | 0.190100 | 0.001800 | 10 |
| CMOPMA        | 0.555500 | 0.001100 | 2 | 0.191300 | 0.001300 | 9 | 0.192900 | 0.001300 | 6 |

Table 7 Scores and ranking of various algorithms about the IGD and HV

| Algorithm     | CCMO | NSGA-III | DCNSGA-III | MOMPA | CMOEA-MS | NSLS | CMOPSO | MOEA/D | NSGA-II | PESA-II |
|---------------|------|----------|------------|-------|----------|------|--------|--------|---------|----------|
| Scores of IGD | 74   | 77       | 104        | 100   | 84       | 118  | 123    | 131    | 164     | 155      | 185    |
| Scores of HV  | 95   | 94       | 72         | 81    | 100      | 133  | 133    | 139    | 138     | 150      | 185    |
| Total         | 169  | 171      | 176        | 181   | 184      | 251  | 256    | 270    | 302     | 305      | 370    |
| Ranking       | 1    | 2        | 3          | 4     | 5        | 6    | 7      | 8      | 9       | 10       | 11     |
| Function | NSGA-II | NSGA-III | MOEA/D | PESA-II | CMOPSO | NSLS | MOMPA | CCMO | CMOEA-MS | DCNSGA-III |
|----------|---------|----------|--------|---------|--------|------|-------|------|----------|------------|
| ZDT1     | 3.02E-11(+) | 1.10E-08(+) | 3.02E-11(+) | 3.02E-11(+) | 2.23E-09(+)) | 3.02E-11(+) | 3.02E-11(+) | 7.22E-06(+)) | 3.02E-11(+) | 4.62E-10(+)) |
| ZDT2     | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 9.92E-11(+)) |
| ZDT3     | 5.09E-08(+) | 6.84E-01(=) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(-) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.50E-09(+)) |
| ZDT4     | 6.05E-07(-) | 1.73E-07(+) | 7.38E-10(+) | 2.23E-09(+)) | 2.37E-10(+)) | 3.02E-11(+) | 5.57E-10(-) | 2.88E-06(-) | 3.01E-07(-) | 3.26E-07(-) |
| ZDT6     | 3.02E-11(+) | 3.69E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) |
| WFG2     | 3.02E-11(+) | 7.39E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(-) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 1.21E-10(+)) |
| WFG3     | 3.02E-11(-) | 3.69E-11(-) | 3.02E-11(-) | 4.64E-05(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| WFG4     | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| WFG5     | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| WFG6     | 5.90E-01(=) | 7.85E-01(=) | 5.20E-01(=) | 6.74E-01(=) | 1.76E-02(-) | 3.02E-11(+) | 7.17E-01(-) | 6.63E-01(=) | 7.39E-01(=) | 7.39E-01(=) |
| WFG7     | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| WFG8     | 3.02E-11(+) | 7.85E-11(+) | 5.07E-10(+) | 3.02E-11(+) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| WFG9     | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| DTLZ1    | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(-) | 6.20E-01(=) | 3.02E-11(+) |
| DTLZ2    | 3.02E-11(+) | 6.77E-05(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) |
| DTLZ3    | 3.02E-11(+) | 5.57E-10(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) |
| DTLZ4    | 3.02E-11(+) | 2.44E-09(+)) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) | 3.02E-11(+) |
| DTLZ5    | 3.02E-11(-) | 1.11E-06(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) |
| DTLZ6    | 3.02E-11(+) | 2.15E-02(+)) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 1.26E-01(=) |
| DTLZ7    | 4.80E-07(-) | 2.39E-08(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 3.02E-11(-) | 5.57E-10(-) | 4.62E-10(-) |
| +\(|-\)  | 14\(\backslash\)5 | 14\(\backslash\)3| 18\(\backslash\)1 | 17\(\backslash\)2 | 14\(\backslash\)5 | 13\(\backslash\)6 | 12\(\backslash\)0| 10\(\backslash\)4| 13\(\backslash\)2| 13\(\backslash\)3|
Balancing the trade-off between cost and reliability for wireless sensor networks: a multi-objective optimized...

Fig. 6 The mean IGD versus iterations obtained by the CMOMPA with different $\theta$ values on ZDT6, DTLZ1, WFG7 over 30 runs

Table 9 Simulation setups

| Parameters                        | Values |
|-----------------------------------|--------|
| Communication range, $R_c$        | 10 m   |
| Measure of the uncertainty, $R_e$ | 2 m    |
| Number of types of SNs            | 3      |
| Deployment cost of SN of type 1, $C_{SN}(1)$ | 2      |
| Deployment cost of SN of type 2, $C_{SN}(2)$ | 5      |
| Deployment cost of SN of type 3, $C_{SN}(3)$ | 10     |
| Sensing range of SN of type 1, $R_s(1)$ | 5 m    |
| Sensing range of SN of type 2, $R_s(2)$ | 10 m   |
| Sensing range of SN of type 3, $R_s(3)$ | 15 m   |
| Number of target points to be sensed | 300    |
| Parameter used in (7), $\lambda_1$ | 0.5    |
| Parameter used in (7), $\lambda_2$ | 1      |
| Threshold used in (8), $\delta$   | 0.8    |
| Relax threshold used in (25), $\varepsilon$ | 0.1    |
| Maximum number of iterations      | 2000   |

Fig. 7 Deployment scenario of a WSN for simulations

Table 10 Probabilities of SN failure

| Network lifetime (day) | Node death probability |
|------------------------|------------------------|
| 0-30                   | 0.018                  |
| 31-60                  | 0.047                  |
| 61-90                  | 0.119                  |
| 91-120                 | 0.500                  |
| 121-150                | 0.880                  |
| 151-180                | 0.980                  |

location ($C_{loc}(i), i = \{1, 2, \ldots, I_{site}\}$) is assigned with a random integer in $[1, 5]$, and only one SN can be deployed in a candidate location. It should be noted that in all experiments, the same deployment costs of locations are used. Unless otherwise stated, the parameter settings are listed in Table 9.

To evaluate the performance of the CMOMPA based optimal SN deployment method, we compared the coverage, connectivity, and deployment costs per day under different $K$-coverage and $C$-connectivity combinations. We assumed that each SN fails with a predefined probability as the working time increases. To be more realistic, the probability of SN failure also increases with the increase of working time, as shown in Table 10. Every 12 hours, we judge whether any surviving node is damaged according to the corresponding probability. We performed 10 independent repetitive experiments for each combination of $K$ and $C$ (See C3 and C4 in (25)).

Firstly, we investigate the effect of $K$ upon the coverage. Figure 8 shows how the average coverage varies over network working time under different $C$ and $K$ combinations. The vertical dash lines indicate the number of days the full coverage of the monitoring are can be maintained. It can be seen that the time period for the network to maintain full coverage is improved with the increase of $K$. However, when $C = 3$, the improvement is not as significant as the scenarios of $C = 1$ and 2. This is because when $C = 3$, there are already enough SNs in the network, which satisfies the connectivity of the network and meanwhile improve the network coverage.

Next, we further study how $C$ affects the network connectivity. Figure 9 depicts the duration of network
connectivity under various $C$ and $K$ combinations. We can read from it that, for a specific $K$, the duration of network connectivity increases with $C$. Similarly, for a specific $C$, the duration of network connectivity increases with $K$. The reason is straightforward since a greater $K$ or $C$ requires more SNs and thus enhance the network connectivity.

Finally, we investigate the daily cost under various $C$ and $K$ combinations. The daily cost of a WSN can be obtained by calculating the total deployment cost divided by its lifetime which is defined as follows:

**Definition 6** Lifetime of a WSN.
The lifetime of a WSN is defined as the duration (in days) that both the full coverage of the monitoring area (constraint $C_1$ in (25)) and network connectivity (constraint $C_2$ in (25)) are satisfied.

The results are shown in Fig. 10. It can be found that the lowest daily cost can be obtained in the case of $K = 1$ and $C = 2$, but not $K = 1$ and $C = 1$. The reason is that the low coverage and connectivity requirements lead to shorter lifetime of the WSN (see Figs. 8 and 9), which results in a higher daily cost. In scenarios with large $K$ and $C$, although the longer network lifetime can be achieved, the daily cost is considerably higher due to much higher total deployment cost.

We further compared with the PFs obtained by CMOMPA and four other top five algorithms listed in Table 7, namely CCMO, NSGA-III, MOMPA, DCNSGA-III. The results are shown in Fig. 11. We can see that the PF of CMOMPA dominates other algorithms, which indicates that the CMOMPA can find solutions with lower cost and higher reliability. It is worth noting that CCMO and DCNSGA-III failed to obtain PFs. Besides, NSGA-III only found one solution, rather than a Pareto surface. This is probably due to the huge dimension of constraints (there are thousands of constraints) in (25). The large-scale constraints limit their global search ability and make them fall into local infeasible regions.

The CMOMPA based deployment method can obtain a set of approximated Pareto optimal solutions, which helps the decision-makers choose the most appropriate one and balance the trade-off among deployment cost, sensing reliability and network reliability based on the actual application requirements. The followings are some examples.

**Example 1** An application needs to deploy a wireless sensor network to monitor the environment over a short period of time. Their budget is limited, but they need to achieve full

---

**Fig. 8** Changes of coverage rates as the working time increases under various $C$ and $K$ combinations

**Fig. 9** Durations of network connectivity under various $C$ and $K$ combinations

**Fig. 10** Daily costs under various $C$ and $K$ combinations
Balancing the trade-off between cost and reliability for wireless sensor networks: a multi-objective optimized...

6 Conclusion

This paper balances the trade-off among the deployment cost, connectivity and coverage for heterogeneous WSNs under the constraints of specific coverage and network connectivity requirements. Different from the existing work, this paper fully considers the heterogeneity of SNs and 3-D environment, which is more suitable for practical WSNs deployments. In view that this is a very complicated multi-objective optimization problem, we propose a novel multi-objective swarm intelligence optimization algorithm, named CMOMPA, to solve it. CMOMPA is a multi-objective version of the MPA. It takes the reference points based method to determine elite individuals and leverages the Gaussian elitist perturbation strategy and competition mechanism-based learning strategy to produce offspring with stronger diversity and distribution. A large number of comparative computational studies over 20 benchmark functions show that CMOMPA has strong competitiveness. Based on CMOMPA, we propose the optimal SN deployment method for 3-D heterogeneous WSNs. Simulation results show that the proposed method can effectively obtain the approximated Pareto optimal solutions from which decision makers can choose the most appropriate solution and balance the trade-off among deployment cost, sensing reliability and network reliability based on the actual requirements of the application. It is worth noting that the CMOMPA proposed in this paper can also be easily used to handle other complex multi-objective optimization problems in engineering and scientific researches, only with necessary modifications on fitness functions, constraints and representations of individuals.

In future research, we will extend our work in the following aspects.

1) Firstly, we will consider the heterogeneity of communication ranges caused by various obstacles within the deployment areas. As known to all, the strengths of radio signals degrade when penetrating obstacles. So each SN may have a different communication range. Considering the heterogeneity of communication range can make our model more consistent with the actual situation.

2) Secondly, although CMOMPA proposed in this paper shows satisfactory performance in solving multi-modal multi-objective optimization problems, it is not effective enough for multi-objective optimization problems with irregular PFs or disconnected PFs. Therefore, improving CMOMPA so that it can deal with such multi-objective optimization problems is another focus of the follow-up research.

3) Thirdly, we are going to apply the proposed optimal deployment method to a real-world WSN applications, which is a large WSN to be deployed in a local large factory.

Acknowledgements This work was supported in part by the Natural Science Foundation of Zhejiang Province, China, under Grant LZ20F010008, and in part by the Xinmiao Talent Program of Zhejiang Province under Grant 2021R429025.
References

1. Ramson SRJino, Moni DJ (2017) Applications of wireless sensor networks—a survey. In: 2017 international conference on innovations in electrical, electronics, instrumentation and media technology (ICIEEMT). IEEE, pp 325–329

2. Faramarzi A, Heidarnejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377

3. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697

4. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN’95 international conference on neural networks, vol 4. IEEE, pp 1942–1948

5. Liu P, Hu Q, Jin K, Yu G, Tang Z (2021) Toward the energy-saving optimization of WLAN deployment in real 3-D environment: a hybrid swarm intelligent method. IEEE Syst J 16(2):2425–2436

6. Chen H, Li S, Heidari AA, Wang P, Li J, Yang Y, Wang M, Huang C (2020) Efficient multi-population outpost fruit fly-driven optimizers: Framework and advances in support vector machines. Expert Syst Appl 142:112999

7. Fan J, Hu Q, Tang Z (2018) Predicting vacant parking space availability: an SVM method with fruit fly optimisation. IET Intell Transp Syst 12(10):1414–1420

8. Zhou M, Lin F, Hu Q, Tang Z, Jin C (2020) AI-enabled diagnosis of spontaneous rupture of ovarian endometriomas: A PSO enhanced random forest approach. IEEE Access 8:132253–132264

9. Negi G, Kumar A, Pant S, Ram M (2021) Optimization of complex system reliability using hybrid grey wolf optimizer. Decision Making: Applications in Management and Engineering 4(2):241–256

10. Das M, Roy A, Maity S, Kar S, Sengupta S (2021) Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm. Decision Making: Applications in Management and Engineering

11. Dulebenets MA (2017) Application of evolutionary computation for berth scheduling at marine container terminals: parameter tuning versus parameter control. IEEE Trans Intell Transp Syst 19(1):25–37

12. Sahebjaninia N, Fathollahi-Fard AM, Hajighahee-Kesheti M (2018) Sustainable tire closed-loop supply chain network design: hybrid metaheuristic algorithms for large-scale networks. J Clean Product 196:273–290

13. Jia H, Miao H, Tian G, Zhou M, Feng Y, Li Z, Li J (2019) Multiobjective bike repositioning in bike-sharing systems via a modified artificial bee colony algorithm. IEEE Trans Autom Sci Eng 17(2):909–920

14. Dhiman G, Singh KK, Soni M, Nagar A, Dehghani M, Sowik A, Kaur A, Sharma A, Houssein EH, Cengiz K (2021) MOSOA: a new multi-objective seagull optimization algorithm. Expert Syst Appl 167:114150

15. Wang L, Zheng X (2018) A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skil resource constrained project scheduling problem. Swarm Evol Comput 38:54–63

16. Cai X, Chen A, Chen L, Tang Z (2021) Joint optimal multi-connectivity enabled user association and power allocation in mmWave networks. In: 2021 IEEE Wireless communications and networking conference (WCNC). IEEE, pp 1–6

17. Liu H, Li Y, Duan Z, Chen C (2020) A review on multi-objective optimization framework in wind energy forecasting techniques and applications. Energy Convers Manag 224:113324

18. Karasu S, Altan A, Bekiros S, Ahmad W (2020) A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy 212:118750

19. Zhang J, Huang Y, Wang Y, Ma G (2020) Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. Constr Build Mater 253:119208

20. Tang Z, Hu Q, Yu G (2016) Energy-efficient multi-objective power allocation for multi-user AF cooperative networks. In: 2016 IEEE Wireless communications and networking conference. IEEE, pp 1–6

21. Ganguly S (2020) Multi-objective distributed generation penetration planning with load model using particle swarm optimization. Decision Making: Applications in Management and Engineering 3(1):30–42

22. Abdel-Basset M, Mohamed R, Elhoseny M, Chakrabortty RK, Ryan M (2020) A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy. IEEE Access 8:79521–79540

23. Ebeed M, Alhejji A, Kamel S, Jurado F (2020) Solving the optimal reactive power dispatch using marine predators algorithm considering the uncertainties in load and wind-solar generation systems. Energies 13(17):4316

24. Abd Elaziz M, Shehabelddeen TA, Elsheikhh AH, Zhou J, Ewees AA, Al-qaness MohammadAA (2020) Utilization of random vector functional link integrated with marine predators algorithm for tensile behavior prediction of dissimilar friction stir welded aluminum alloy joints. J Mater Res Technol 9(5):11370–11381

25. Cheng R, Jin Y (2014) A competitive swarm optimizer for large scale optimization. IEEE Trans Cybern 45(2):191–204

26. Deb K, Jain H (2013) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601

27. Zitler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195

28. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506

29. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization. Springer, pp 105–145

30. Özdağ R, Karci A (2015) Sensor node deployment based on electromagnetism-like algorithm in mobile wireless sensor networks. Int J Distrib Sensor Netw 11(2):507967

31. Zhang Y, Wang L (2010) A distributed sensor deployment algorithm of mobile sensor network. In: 2010 8th World congress on intelligent control and automation. IEEE, pp 6963–6968

32. Guo J, Jafarkhani H (2019) Movement-efficient sensor deployment in wireless sensor networks with limited communication range. IEEE Trans Wirel Commun 18(7):3469–3484

33.-Yoon, Kim Y-H (2013) An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. IEEE Trans Cybern 43(5):1473–1483

34. Liu X, He D (2014) Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks. J Netw Comput Appl 39:310–318

35. Guo J, Jafarkhani H (2016) Sensor deployment with limited communication range in homogeneous and heterogeneous wireless sensor networks. IEEE Trans Wirel Commun 15(10):6771–6784

36. Chen C-P, Mukhopadhyay SC, Chuang C-L, Lin T-S, Liao M-S, Wang Y-C, Jiang J-A (2014) A hybrid memetic framework for
Balancing the trade-off between cost and reliability for wireless sensor networks: a multi-objective optimized...

37. Kashino Z, Nejat G, Benhabib B (2018) A hybrid strategy for target search using static and mobile sensors. IEEE Trans Cybern 45(10):2309–2322
38. Vilela J, Kashino Z, Ly R, Nejat G, Benhabib B (2016) A dynamic approach to sensor network deployment for mobile-target detection in unstructured, expanding search areas. IEEE Sensors J 16(11):4405–4417
39. Zorlu O, Sahingoz OK (2016) Increasing the coverage of homogeneous wireless sensor network by genetic algorithm based deployment. In: 2016 Sixth international conference on digital information and communication technology and its applications (DICTAP). IEEE, pp 109–114
40. Mekikis P-V, Kartaschi E, Antonopoulos A, Alonso L, Verikoukis C (2018) Connectivity analysis in clustered wireless sensor networks powered by solar energy. IEEE Trans Wirel Commun 17(4):2389–2401
41. Piltyay S, Bulasenko A, Demchenko I (2020) Wireless sensor network connectivity in heterogeneous 5G mobile systems. In: 2020 IEEE International conference on problems of infocommunications, science and technology (PIC S&T). IEEE, pp 625–630
42. Dagdeviren O, Akram VK (2017) PACK: path coloring based k-connectivity detection algorithm for wireless sensor networks. Ad Hoc Netw 64:41–52
43. Saeed N, Celik A, Al-Naffouri TY, Alouini M-S (2018) Connectivity analysis of underwater optical wireless sensor networks: a graph theoretic approach. In: 2018 IEEE International conference on communications workshops (ICC Workshops). IEEE, pp 1–6
44. Mazumder A, Zhou C, Das A, Sen A (2016) Budget constrained relay node placement problem for maximal “connectedness”. In: MILCOM 2016-2016 IEEE military communications conference. IEEE, pp 849–854
45. Harizan S, Kuila P (2020) A novel NSGA-II for coverage and connectivity aware sensor node scheduling in industrial wireless sensor networks. Digit Signal Process 105:102753
46. Akram VK, Dagdeviren O, Tavl B (2020) Distributed k-connectivity restoration for fault tolerant wireless sensor and actuator networks: algorithm design and experimental evaluations. IEEE Trans Reliab
47. Konstantinidis A, Yang K (2011) Multi-objective k-connected deployment and power assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition. Comput Commun 34(1):83–98
48. Sheikhi H, Hoseini M, Sabaei M (2021) k-connected in heterogeneous wireless sensor networks. Wirel Pers Commun 120(4):3277–3292
49. Mostafaei H, Montieri A, Persico V, Pescapé A (2017) A sleep scheduling approach based on learning automata for WSN partialcoverage. J Netw Comput Appl 80:67–78
50. Natarajan P, Parthiban L (2020) k-coverage m-connected node placement using shuffled frog leaping: Nelder–Mead algorithm in WSN. Journal of Ambient Intelligence and Humanized Computing, 1–16
51. Gupta SK, Kuila P, Jana PK (2016) Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks. Comput Electr Eng 56:544–556
52. Sheikhi H, Barkhoda W (2020) Solving the k-coverage and m-connected problem in wireless sensor networks through the imperialistic competitive algorithm. J Interconn Netw 20(01):2050002
53. Hechmi JM, Zrelli A, Khbida M, Khlaifi H, Ezzedine T (2018) Coverage and connectivity of WSN models for health open-pit mines monitoring. In: 2018 14th International wireless communications & mobile computing conference (IWCMC). IEEE, pp 310–315
54. Ganesan T, Rajarajeswari P (2019) Genetic algorithm approach improved by 2D lifting scheme for sensor node placement in optimal position. In: 2019 International conference on intelligent sustainable systems (ICISS). IEEE, pp 104–109
55. Huang C-F, Tseng Y-C, Wu H-L (2007) Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network. ACM Trans Sensor Netw (TOSN) 3(1):5–es
56. Yue Y, Cao L, Luo Z (2019) Hybrid artificial bee colony algorithm for improving the coverage and connectivity of wireless sensor networks. Wirel Pers Commun 108(3):1719–1732
57. Yan F, Ma W, Shen F, Xia W, Shen L (2020) Connectivity based k-coverage hole detection in wireless sensor networks. Mob Netw Applic 25(2):783–793
58. Njoya AN, Ari AAA, Awa MN, Titiouna C, Labraoui N, Effa YJ, Abdou W, Guerroui A (2020) Hybrid wireless sensors deployment scheme with connectivity and coverage maintaining in wireless sensor networks. Wirel Pers Commun 112(3):1893–1917
59. Harizan S, Kuila P (2020) Nature-inspired algorithms for k-coverage and m-connectivity problems in wireless sensor networks. In: Design frameworks for wireless networks. Springer, pp 281–301
60. Chawra VK, Gupta GP (2021) Memetic algorithm based energy efficient wake-up scheduling scheme for maximizing the lifetime, coverage and connectivity in three-dimensional wireless sensor networks. Wirel Pers Commun, 1–16
61. Boualem A, Dahmani Y, Runz CD, Ayaida M (2019) Spiderweb strategy: application for area coverage with mobile sensor nodes in 3D wireless sensor network. Int J Sensor Netw 29(2):121–133
62. Hassan MH, Yousri D, Kamel S, Rahmann C (2022) A modified marine predators algorithm for solving single-and multi-objective combined economic emission dispatch problems. Comput Industr Eng 164:107906
63. Abdel-Basset M, Mohamed R, Mirjalili S, Chakrabortty RK, Ryan M (2021) An efficient marine predators algorithm for solving multi-objective optimization problems: analysis and validations. IEEE Access 9:42817–42844
64. Al Harthi M, Ghoneim S, Elsayed A, El-Sehiemy R, Shaheen A, Ginidi A (2021) A multi-objective marine predator optimizer for optimal techno-economic operation of AC/DC grids. Stud Inform Control 30:89–99
65. Yousri D, Ousama A, Fathy A, Babu TS, Allam D et al (2022) Managing the exchange of energy between microgrid elements based on multi-objective enhanced marine predators algorithm. Alexandria Eng J 61(11):8487–8505
66. Zhong K, Zhou G, Deng W, Zhou Y, Luo Q (2021) MOMPA: multi-objective marine predator algorithm. Comput Methods Appl Mech Eng 385:114029
67. Jangir P, Buch H, Mirjalili S, Manoharan P (2021) MOMPA: multi-objective marine predator algorithm for solving multi-objective optimization problems. Evol Intel, 1–27
68. Chen L, Cui X, Jin K, Tang Z (2021) MOMPA: a high performance multi-objective optimizer based on marine predator algorithm. In: Proceedings of the genetic and evolutionary computation conference companion, pp 177–178
69. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
70. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
71. Tian G, Ren Y, Zhou M (2016) Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution
and particle swarm optimization combined algorithm. IEEE Trans Intell Transp Syst 17(11):3009–3021.

72. Abdel-Basset M, Mohamed R, Mirjalili S (2021) A novel whale optimization algorithm integrated with nelder–mead simplex for multi-objective optimization problems. Knowl-Based Syst 212:106619

73. Fathollahi-Fard AM, Ahmadi A, Karimi B (2021) Multi-objective optimization of home healthcare with working-time balancing and care continuity. Sustainability 13(22):12431

74. Pasha J, Nwodu AL, Fathollahi-Fard AM, Tian G, Li Z, Wang H, Dulebenets MA (2022) Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings. Adv Eng Inform 52:101623

75. Seydanlou P, Jalili F, Tavakkoli-Moghaddam R, Fathollahi-Fard AM (2022) A multi-objective optimization framework for a sustainable closed-loop supply chain network in the olive industry: hybrid meta-heuristic algorithms. Expert Syst Appl, 117566

76. Yuan G, Yang Y, Tian G, Fathollahi-Fard AM (2022) Capacitated multi-objective disassembly scheduling with fuzzy processing time via a fruit fly optimization algorithm. Environ Sci Pollut Res, 1–18

77. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

78. Zou Y, Chakrabarty K (2003) Target localization based on energy considerations in distributed sensor networks. In: Proceedings of the first IEEE international workshop on sensor network protocols and applications, 2003. IEEE, pp 51–58

79. Li M, Yang S, Liu X (2013) Shift-based density estimation for Pareto-based algorithms in many-objective optimization. IEEE Trans Evol Comput 18(3):348–365

80. Deb K, Goyal M et al (1996) A combined genetic adaptive search (geneas) for engineering design. Comput Sci Inf 26:30–45

81. Corte DW, Jerram NR, Knowles JD, Oates MJ (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd annual conference on genetic and evolutionary computation, pp 283–290

82. Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2018) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. Inf Sci 427:63–76

83. Chen B, Zeng W, Lin Y, Zhang D (2014) A new local search-based multiobjective optimization algorithm. IEEE Trans Evol Comput 19(1):50–73

84. Tian Y, Zhang T, Xiao J, Zhang X, Jin Y (2020) A coevolutionary framework for constrained multiobjective optimization problems. IEEE Trans Evol Comput 25(1):102–116

85. Tian Y, Zhang T, Su Y, Zhang X, Tan KC, Jin Y (2021) Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. IEEE Transactions on Cybernetics

86. Jiao R, Zeng S, Li C, Yang S, Ong Y-S (2020) Handling constrained many-objective optimization problems via problem transformation. IEEE Transactions on Cybernetics

87. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87

88. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132

89. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271

90. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C (2022) An evolutionary multitasking optimization framework for constrained multiobjective optimization problems. IEEE Trans Evol Comput 26(2):263–277

91. Sun Z, Ren H, Yen GG, Chen T, Wu J, An H, Yang J (2022) An evolutionary algorithm with constraint relaxation strategy for highly constrained multiobjective optimization. IEEE Transactions on Cybernetics

92. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inform Sci 180(10):2044–2064

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

Long Chen received the B.S. degree in the information engineering from Zhengzhou University of Light Industry, Henan, China, in 2018. He is currently a master student in Computer Science and Technology from Wenzhou University, Zhejiang, China. His research interests include sensor networks and multi-objective evolutionary algorithms.

Yingying Xu received the Ph.D. degree in communication and information system from Beijing Jiaotong University, Beijing, China, in 2014. She is currently a lecturer at the College of Computer Science & Artificial Intelligence, Wenzhou University, Wenzhou, China. Her research interests include cybersecurity, swarm intelligence and machine learning.

Fangyi Xu received the B.S. degree in the network engineering from Wenzhou University, Zhejiang, China, in 2022. Her research interests include artificial intelligence, network optimization and evolutionary algorithms.
Qian Hu received the B.S. degree in communication engineering from the Nanjing Institute of Posts and Telecommunications, Nanjing, China, in 2001, and the M.S. degree in circuits and systems from Zhejiang University, Hangzhou, China, in 2005.

She is currently an Associate Professor with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China. Her research interests include machine learning, intelligent systems, and wireless networks.

Zhenzhou Tang (Senior Member, IEEE) received the B.S. and M.S. degrees in communication engineering from Zhejiang University, Hangzhou, China, in 2001 and 2004, respectively, and the Ph.D. degree in communication engineering from Dalian University of Technology, Dalian, China, in 2015.

In 2004, he joined Wenzhou University, Wenzhou, China, where he is currently a full professor with the College of Computer Science and Artificial Intelligence, and the Director of Innovation Research Center for Intelligent Networking. From February 2015 to August 2015, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA. From 2015 to 2018, he was a Postdoctoral Research Fellow with Zhejiang University. His current research interests include machine learning, intelligent systems, and wireless networks.