Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 1, 2023, 1-12.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A Bi-objective Virtual-force Local Search PSO Algorithm for Improving Sensing Deployment in Wireless Sensor Network

Vahid Kiani[1*] and Mahdi Imanparast[2]

1. Department of Computer Engineering, University of Bojnord, Bojnord, Iran.
2. Department of Computer Science, University of Bojnord, Bojnord, Iran.

Article Info

Article History:
Received 13 May 2022
Revised 14 November 2022
Accepted 11 January 2023

DOI:10.22044/jadm.2023.11917.2339

Keywords:
Maximum Coverage, Coverage Improvement, Particle Swarm Optimization, Virtual Force Operator, Coverage Rate.

**Corresponding author: v.kiani@ub.ac.ir(V. Kiani).*

Abstract

In this paper, we present a bi-objective virtual-force local search particle swarm optimization (BVFPSO) algorithm to improve the placement of sensors in wireless sensor networks, while it simultaneously increases the coverage rate and preserves the battery energy of the sensors. Mostly, sensor nodes in a wireless sensor network are first randomly deployed in the target area, and their deployment should then be modified such that some objective functions are obtained. In the proposed BVFPSO algorithm, PSO is used as the basic meta-heuristic algorithm, and the virtual-force operator is used as the local search. As far as we know, this is the first time that a bi-objective PSO algorithm has been combined with a virtual force operator to improve the coverage rate of sensors while preserving their battery energy. The results of the simulations on some initial random deployments with the different numbers of sensors show that the BVFPSO algorithm by combining two objectives and using virtual-force local search is enabled to achieve a more efficient deployment in comparison to the competitive algorithms PSO, GA, FRED, and VFA with providing simultaneously the maximum coverage rate and the minimum energy consumption.

1. Introduction

Mobile wireless sensor networks are used to track targets, monitor the environment, capture environmental data, and gather information about moving objects in an area [1, 2]. The success of the mobile sensor networks in these various applications is highly dependent on the location of the sensors, which is referred to the deployment of sensors [3]. Sensors in a mobile wireless sensor network are powered by a battery with a limited energy, and have the ability to perceive the environment, perform limited calculations, communicate, and move to a new location. In this way, a mobile wireless sensor network can modify its sensors arrangement to achieve a better coverage of the target area [4]. In many applications, the initial deployment in the target area can be performed only by random deployment [5]. After the initial deployment, the sensor network should improve the arrangement of its sensor nodes by relocating them. Moreover, when one or more sensor nodes are unable to operate due to low battery, the sensor network may modify its node deployment to repair holes. The higher coverage rate, the more likely the wireless sensor network will detect and track the targets. On the other hand, moving the sensors will waste some of their battery power.

This paper aims to propose an algorithm that improves the quality of an initial deployment, so that the final deployment covers the target area well and the relocation consumes the lowest energy to relocate the sensors [6]. In the following, we review several research works related to sensor deployment. Most researchers in this area only paid attention to the coverage and tried to achieve the maximum possible coverage rate [3, 7]. For example, Liang *et al*. [8] proposed four heuristic and approximate algorithms for maximizing target coverage in a wireless sensor network. Zain Eldin *et al.* [9] proposed a new genetic algorithm with a new cross-over operator to maximize the coverage area while using the least possible number of sensors. Binh *et al.* proposed a genetic algorithm and a particle swarm optimization for maximizing coverage area in heterogeneous sensor networks constituted by sensor nodes of several types [10]. Tarnaris *et al.* [11] evaluated the performance of the genetic algorithm and particle swarm optimization in maximizing area coverage and k-coverage for a wireless sensor network. Osmani *et al*. [12] proposed a fuzzy redeployment algorithm (FRED), and employed a Voronoi diagram to determine the optimal position of sensors. They designed a fuzzy inference system to decide whether a sensor should move to its new location or not in each step of the algorithm. The virtual force algorithm has been a successful meta-heuristic algorithm for increasing the coverage rate of wireless sensor networks, and many improved versions of it have been proposed by various researchers. Liu *et al*. [13] proposed an improved virtual molecular force algorithm (IVMFA) based on the traditional virtual force algorithm to maximize the coverage rate in the wireless sensor network by considering the force network between the sensors and direct the sensors to fill the holes. Their algorithm considers the force network between the sensors and directs the sensors to fill the holes. Wang *et al*. [14] proposed a gray wolf optimization (GWO) algorithm equipped with a levy flight mechanism and a virtual force operator to improve the sensors' deployment in the target area. Xie *et al*. [15] presented an improved virtual force algorithm based on area intensity (IVFAI) that uses the density of sensors around each sensor to determine the appropriate distance between neighboring sensors during the redeployment process. Deng *et al*. [16] proposed a virtual spring force algorithm based on the Newton force rules to maximize the coverage area. Their algorithm adjusts the forces between the sensors according to the laws of Newton force and external central force, maximizes the coverage area while filling the holes using a central external force. Song *et al*. [17] developed an improved virtual force biogeography-based optimization (VFBBO) algorithm to increase the coverage rate, which also uses a virtual force operator to guide the sensors to better locations. Reducing the energy consumed by the sensor nodes during the redeployment process is one of the secondary goals that have been given special attention by researchers [18]. Abo-Zahhad *et al*. [6] used a bi-objective artificial immune system (IS) algorithm to reduce sensor moved distances while increasing the coverage rate. They considered maximizing coverage rate as the first goal, and minimizing sensor moved distances as the second goal, and combined these two goals into one objective function. Their proposed method does not use a virtual force operator. In fact, they consider both objectives of increasing the coverage rate and decreasing the displacement in the artificial immune system algorithm with the limited displacement model. Zhang *et al*. [19] proposed a two-phase method for simultaneously optimizing coverage rate and energy consumption. It maximizes coverage in the first phase with differential evolution and minimizes displacement in the second phase with a heuristic. Aziz *et al*. [20] used a two-phase PSO algorithm to solve the problem, the first phase aims to increase coverage rate, and the second phase aims to reduce sensor displacement. Qu *et al*. [21] proposed a multi-objective genetic algorithm to improve the deployment of the sensors, where its first objective is to increase the coverage area, and the second objective is to reduce the average sensor moved distances (displacement). Tuba *et al.* [22] considered the maximum coverage in mobile WSN with minimal displacement, and used the firefly algorithm, for solving their defined multi-objective optimization problem. Bai *et al*. [23] considered the problem of k-coverage of target points, and proposed a PSO algorithm to optimize the k-coverage, which uses a limited circular mobility model to reduce energy consumption of the sensors while maximizing the coverage rate. Heo *et al*. [24] proposed three algorithms including a distributed self-expanding algorithm, a peer-to-peer clustering algorithm, and a distributed algorithm based on the Voronoi diagram to increase the coverage rate and reduce energy consumption during sensors placement. Pournazari *et al*. [25] proposed a distributed algorithm for setting the position and direction of sensors in a multi-media sensor network to maximize area coverage. Qin and Chen [26] improved differential evolution algorithm with variation, cross-over, and selection operators to maximize the coverage rate while using a compensating operator for balanced energy distribution. Sheikhi and Barkhoda [27] use the migration operator between sub-populations in the imperialist competitive algorithm (ICA) to maximize the coverage of target points with two constraints of *k*-coverage target points and *m*-connected neighboring sensors. Gupta and Jha

[28] propose a biogeography-based optimization (BBO) algorithm to complete coverage of target points with two constraints of *k*-coverage of target points and *m*-connected of neighboring sensors. Hosseinirad [29] focused on reducing energy consumption of WSN and proposed a dynamic multilayer hierarchy clustering approach using evolutionary algorithms for densely deployed WSN.

In this paper, a bi-objective virtual-force local search PSO (BVFPSO) algorithm is presented to reduce the battery energy consumption of sensors while improving the coverage rate. BVFPSO considers two objectives of coverage improvement and energy preservation, and also uses a virtual-force algorithm as the local search operator for faster convergence. As far as we know, this is the first time that a bi-objective PSO algorithm has been combined with the virtual force algorithm to reduce the movement of sensors while improving the network coverage rate. Moreover, we have considered the limited movement model for the sensors. The movement range of each sensor relative to its initial position is limited by a circle to prevent sensors from moving too much.

Our main contributions are summarized as follows:

- Using a virtual-force algorithm as the local search operator for faster convergence.
- Obtaining a better coverage rate and a minimum moved distance, simultaneously, in most cases in comparison with other existing methods.

Our paper is organized as what follows. In section 2 the problem that this research work intends to solve is defined. Section 3 details the proposed BVFPSO algorithm. Section 4 evaluates the performance of the BVFPSO algorithm, and compares it with the four competitive algorithms PSO, GA, VFA, and FRED. Finally, Section 5 is devoted to conclusions and suggestions for future research works.

2. Problem description and assumptions

The problem that we intend to solve in this work is to improve the deployment of the sensors in an initial location, so that we first achieve the maximum coverage rate of the area, and secondly minimize the amount of sensor movement. The energy conservation of the sensors helps to increase the lifespan of the sensor network, and to monitor the environment for a longer period of time. In this case, it is assumed that p sensors in an initial deployment in a two-dimensional space are randomly arranged, and their location should be modified so that the coverage rate increases as the most important target by reducing redundant overlaps and better distribution of sensors in space. At the same time, we intend to modify the sensor deployment so that the sensor movement is also minimal, and so that the battery energy of sensors is preserved and the lifespan of the network is increased.

The first goal is to maximize the coverage rate. We denote the coverage rate of the deployment S by $f_1(S)$, which indicates what percentage of the target area is perceived by deployment S of sensors. In this work, we use the binary sensor model. In the binary sensor model shown in Figure 1, each sensor is aware of its current position, has a perceptual radius r, and identifies with complete certainty any object that appears in the circular region of the radius r around itself. In order to estimate the coverage rate of the deployment S, we use a grid of points in the target area similar to Figure 2.

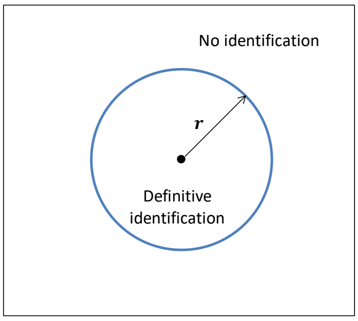

Figure 1. Binary model of a sensor node.

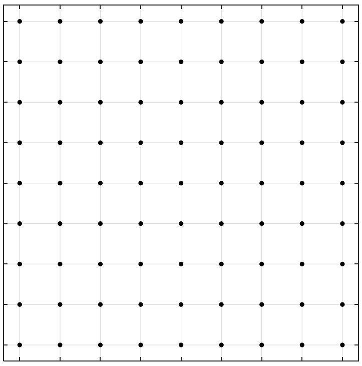

Figure 2. A gridded area with the help of a coarse grid.

For each grid point, as shown in Figure 3, it is checked that the point is covered by at least one sensor. The ratio of the number of target points covered to the total points in the target area shows the coverage rate of the deployment S:

$$\max\left(f_1(S) = \frac{N_c}{N} \right) \tag{1}$$

where N_c represents the number of covered points in the grid, and N represents the total number of grid points.

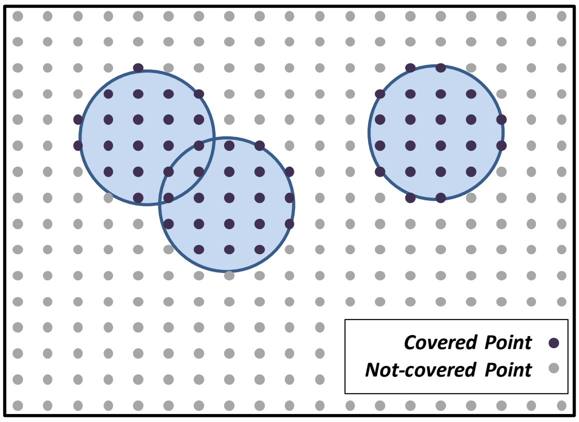

Figure 3. Coverage rate as the first goal of the proposed BVFPSO algorithm and its estimation using a grid of target points.

The second goal is to minimize battery energy consumption. Each sensor requires battery energy to move and relocate. Excluding acceleration, battery power consumption E_{mov} is linearly related to traveled distance as [25]:

$$E_{mov} = k_{mov} \cdot d_{mov} \tag{2}$$

where k_{mov} is a constant coefficient that indicates the rate of energy consumption, and d_{mov} is the displacement distance. Therefore, to reduce the energy consumption of the sensors, it is sufficient to minimize the sum of displacement distances of the sensors between the initial deployment and the final deployment. In this work, in order to prevent the sensors from moving too much, we have limited the maximum displacement of each sensor to 3r. In order to minimize the energy consumption of the sensors, it is sufficient to minimize the objective of the displacement distance as follows:

$$\min\left(f_2(S) = \frac{d_{rms}}{d_{max}} \right) \tag{3}$$

where d_{rms} shows the average displacement distances of the sensors to reach the deployment S from the initial deployment, and is calculated as follows:

$$d_{rms} = \sqrt{\frac{1}{p}\sum_{i=1}^{p} d_i^2} \tag{4}$$

where d_i shows the displacement of the ith sensor in the deployment S relative to the initial deployment. The moving distance of several sensors is visualized in Figure 4. The blue solid circles in this figure show the initial position of the sensors, and the red dashed circles show the final position of the sensors. Blue arrows in Figure 4 show the moved distance of the sensors.

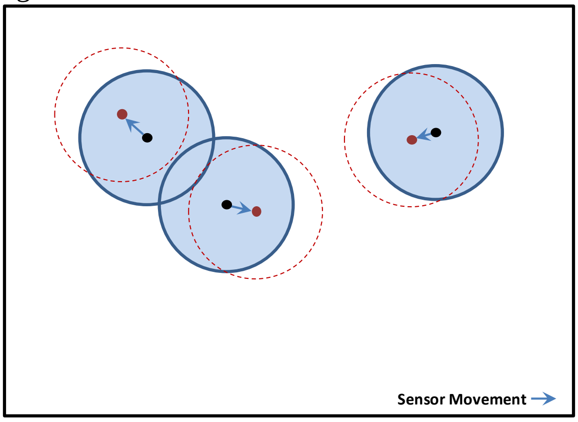

Figure 4. Moved distance of the sensors, which is minimized as the second goal of the proposed BVFPSO algorithm.

3. Proposed algorithm

In this section, we propose a BVFPSO algorithm for the mentioned problem. The proposed algorithm seeks to maximize the coverage rate while preserving the battery energy of the sensor nodes. Although these two goals conflict with each other, by combining them in one objective function, a good balance between these two goals can be obtained [6]. BVFPSO is a bi-objective memetic algorithm. In a memetic meta-heuristic algorithm, in addition to collective evolution mechanisms such as recombination and cross-over, individuals in a population have the opportunity to improve themselves with the local improvement mechanisms. A memetic meta-heuristic algorithm can be created by combining a population-based meta-heuristic algorithm with a local search operator [30]. In BVFPSO, the PSO algorithm is used as the basic meta-heuristic algorithm, and the virtual force as the local search operator. The virtual force operator is confined to the best particle of the population at the end of each iteration. BVFPSO should be executed in the cluster-head node, and calculates the new position of the sensor nodes based on their current position.

In the proposed BVFPSO, several movements are employed on particles to find the optimal solution in the search space. These particle movements are visualized in Figure 5. Similar to traditional PSO, in the BVFPSO, the velocity vector of each particle is constituted by a cognitive movement vector, a social movement vector, and an inertia movement vector. The cognitive movement is toward the personal best position of the current particle, the social movement vector is toward the global best position of the current population, and the inertia movement vector is in the same direction as the velocity vector of the particle in the previous time step. In addition to these traditional movement vectors, in BVFPSO, several local search moves would be carried out on the global best particle of the current population. These local search movements are determined in BVFPSO by the virtual-force operator. In addition, BVFPSO considers two goals, maximization of coverage and preservation of battery energy of the sensors during the redeployment process.

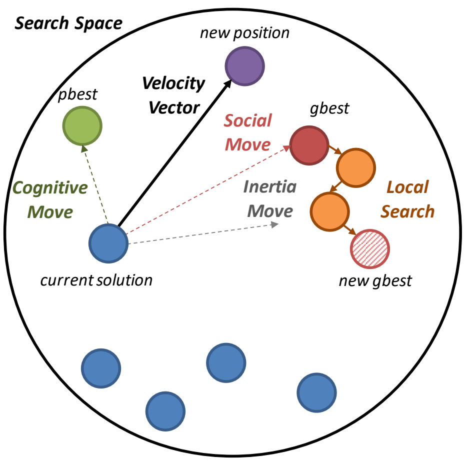

Figure 5. Visualization of solutions population and different particle movements in the proposed BVFPSO algorithm.

The flowchart of the proposed BVFPSO algorithm is shown in Figure 6. In BVFPSO, in order to generate the initial population of particles, random solutions around the initial placement are generated using a perturbation operator. A bi-objective fitness function that combines the coverage rate and moving distance of the sensors is then utilized to evaluate the quality of each solution. After that the personal best position of each particle and the best global position of the population is updated, velocity vectors are computed, and the position of particles is updated. At the end of each iteration, a random mutation operator and local search operator are applied. For random particle mutation, one of the sensor nodes in the current particle is randomly displaced to a close location in its local neighborhood. Local search is applied to the best particle of the population using the virtual-force operator. If the local search leads to a better deployment, the new deployment replaces the best solution. More details of the proposed algorithm are described below.

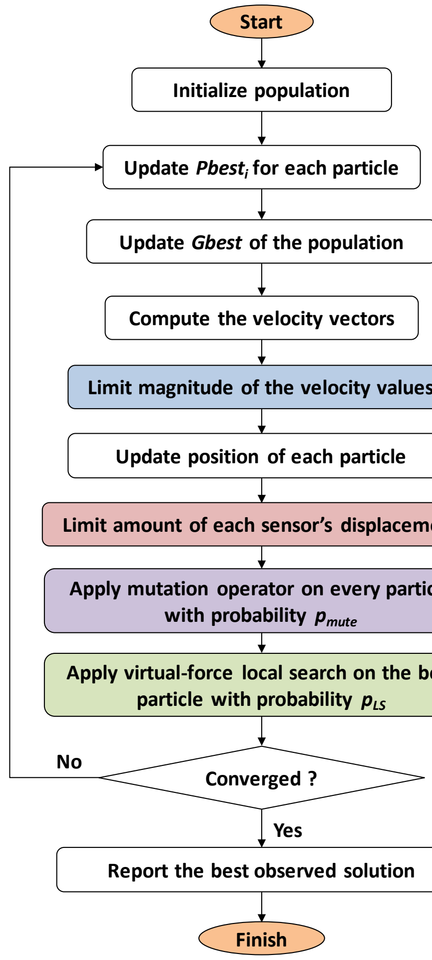

Figure 6. Flowchart of the proposed BVFPSO algorithm to improve the location of sensors in the wireless sensor networks.

3.1 Solution representation

In the proposed algorithm, each solution to the problem is a new deployment represented by a particle. A solution corresponds to the location of the p sensors, where the number of sensors is specified as the input of the algorithm. Each particle is represented by an array of length $2p$, where the position (x_i, y_i) of the sensor i is stored in cells $2i$-1 and $2i$ of the solution array, respectively. In this representation, the odd cells store the x values and the even cells store they values of the sensor locations. During the execution of the proposed algorithm, the values of x_i are limited to the range [x_{min}, x_{max}], and the values of y_i are limited to the range [y_{min}, y_{max}]. Figure 7 shows how to display the solution in the BVFPSO algorithm.

3.2 Population initialization

In order to generate the initial population, we use the perturbation operator on the initial deployment. In the proposed perturbation operator, each cell of the solution vector was calculated as follows:

$$S_d^i(1) = S_d^{init} + r_0 \frac{d_{max}}{2} \tag{5}$$

where S^i_d is dth dimension of ith solution in the population at the first iteration, r_0 is a random value in the interval [-1, +1], and S_d^{init} represents the d'th dimension of the solution vector corresponding to the initial deployment. In addition to the solutions generated by the perturbation operator, the initial deployment is also included as a particle in the initial population.

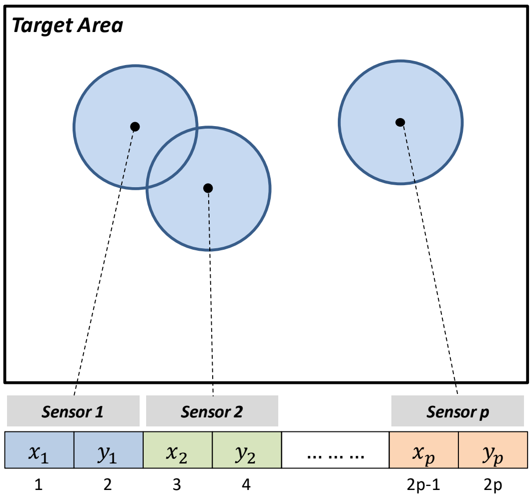

Figure 7. Numerical representation of sensor deployment in the BVFPSO algorithm.

3.3 Bi-objective optimization

In this research work, we aim to achieve two goals during sensor redeployment. The first goal is to improve the coverage rate of the sensor nodes by starting from initial deployment. The second goal is to minimize the amount of sensors' movement and preserve their battery energy. Therefore, the final goal of the proposed algorithm is to both maximize the coverage rate $f_1(S)$ and minimize the moved distance $f_2(S)$. We combine these two goals in the main cost function of the proposed algorithm as follows [19]:

$$\min\left(f\left(S^i\right) = w_{cost}\left(1 - f_1\left(S^i\right)\right) + \left(1 - w_{cost}\right) f_2\left(S^i\right)\right) \tag{6}$$

where the weight coefficient w_{cost} determines the relative importance of achieving a higher coverage rate versus lower energy consumption. By varying the weight w_{cost} from 0 to 1, different solutions can be obtained that balance these two goals. Since increasing the coverage rate is in conflict with preserving the battery energy, by increasing the value of w_{cost}, we expect the coverage rate is increased in the final solution, and consequently, the traveled distance by the sensors is increased.

3.3.1 Fitness evaluation

In meta-heuristic algorithms, the quality of a solution is known as its fitness, and the algorithm seeks to increase it. On the other hand, in the proposed algorithm, the value of the cost function $f(S)$ must be minimized. Therefore, to map the value of the cost function to the fitness of each solution, we use the following equation:

$$fit\left(S^i\right) = \frac{1}{1 + f\left(S^i\right)}$$

3.3.2 Particle position adjustment

In particle swarm optimization, each particle has a personal memory, denoted by $pbest_i$, that retains the best position the particle has ever been in. Moreover, the best solution ever discovered in the whole population is stored in an aggregative memory called *gbest*, which is shared among the particles. During the execution of the algorithm, each particle modifies its position based on its current position and its velocity vector is as follows:

$$x_{ik+1} = x_{ik} + v_{ik} \tag{8}$$

where the vector x_{ik} represents the position of the ith particle in the kth iteration, and the vector v_{ik} represents the velocity vector of the ith particle in the kth iteration. The velocity vector of each particle is also modified in each iteration. First, the velocity vector of each particle is filled with zero. Then in each iteration, the velocity vector in the current iteration is updated based on the velocity vector of the same particle in the previous iteration, the current position of that particle, its personal best position, and the global best position according to the following relation:

$$v_{ik+1} = w_k v_{ik} + c_1 r_1\left(pbest_{ik} - x_{ik}\right) + c_2 r_2\left(gbest_k - x_{ik}\right) \tag{9}$$

where r_1 and r_2 are two random numbers in the range [0,1], the coefficient w_k is called the inertia factor, the coefficient c_1 is called the cognitive learning factor, and the coefficient c_2 is called the social learning factor. Using a small inertia factor will cause the algorithm to focus on search space exploration and using a large value for the inertia coefficient will cause the algorithm to focus on local search and extract the solution in the current search area. Since the value of the velocity vector is calculated as a summation, the magnitude of the velocity values is usually controlled during execution so that does not exceed a certain limit.

3.3.3 Mutation operator

In order to increase the population diversity and better explore the search space, we use a mutation operator specific to the sensor deployment problem. The mutation operator is applied by probability P_{mute} to any solution in the population. When the mutation operator is applied to a solution, one of the current solution's sensors is randomly selected, and its position (*x*,*y*) is mutated, i.e. a random value is added to *x*, and another random value is added to *y*. The proposed mutation operator can be explained as follows:

$$x_{ik}^{2j-1} = x_{ik}^{2j-1} + r_3\left(\frac{r}{2}\right) \quad (10)$$

$$x_{ik}^{2j} = x_{ik}^{2j} + r_4\left(\frac{r}{2}\right) \quad (11)$$

where r_3 and r_4 are two random values in the range [-1, +1], and *j* is the index of the sensor randomly selected to perform the mutation operation. The mutation operation of a particle is visualized in Figure 8. The sensor at the top right corner of the target area is selected as the mutated sensor in Figure 8, and then moved randomly to a new location in its local neighborhood.

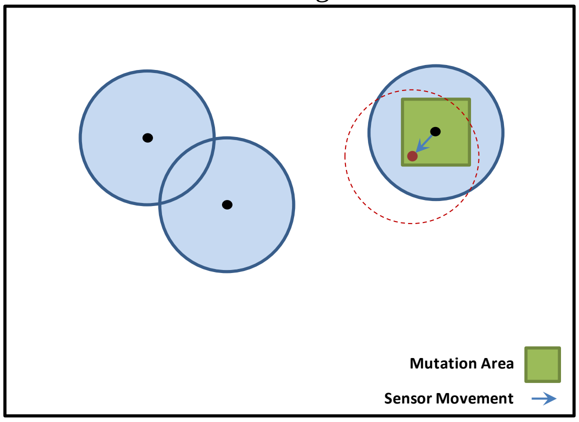

Figure 8. Random movement of one sensor in the mutation of sensor deployment.

3.4 Virtual-force local search

The proposed virtual-force operator in this paper modifies the position of the sensors by considering the attractive and repulsive forces between neighboring sensors so that the sensors approach an optimal distance from each other [5], [14], [15]. This avoids over-concentrating the sensors in a particular region or over-isolating the sensors in the target area. In order to improve an existing deployment, if two neighboring sensors become too close to each other, they must repel each other to avoid excessive overlap of the sensor nodes. On the other hand, if two neighboring sensors do not overlap but are relatively close to each other, they should absorb each other slightly to provide uniform coverage in the target area. Finally, when two sensors are far apart, they are not considered neighbors, and will not affect each other. As an example, in Figure 9 we intend to calculate the total force on the sensor s_1. Since the distance between sensor s_2 and sensor s_1 is greater than the optimal distance d_{th}, sensor s_2 will apply an attractive force to sensor s_1, which is represented by a dashed arrow between the two sensors. On the other hand, the distance from sensor s_3 to sensor s_1 is less than the optimal distance d_{th}, so sensor s_3 will apply a repulsive force to sensor s_1. The average vector of these two vectors is calculated as the total force applied to sensor s_1, which is shown by a black thick arrow in Figure 9.

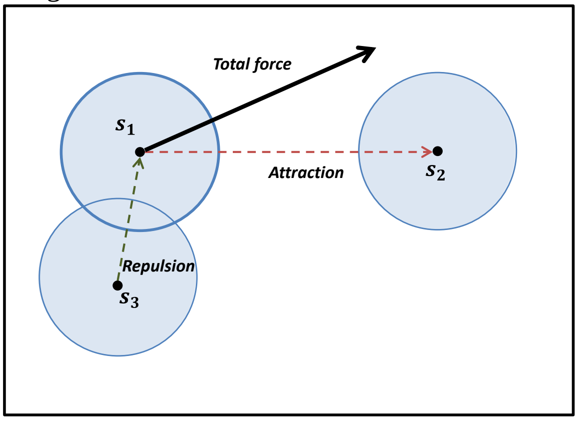

Figure 9. Attractive and repulsive forces between sensor nodes in the virtual-force local search operator.

More formally, suppose that we show the optimal distance of the sensors by d_{th}. The exerted force on the sensor *j* from sensor *j'* is represented by the symbol $F_{jj'}$, which is calculated according to the following equation in polar coordinates:

$$F_{jj'} = \begin{cases} \left(w_A\left(d_{jj'} - d_{th}\right), \alpha_{jj'}\right) & d_{th} < d_{jj'} < R \\ \left(w_R \dfrac{1}{d_{jj'}}, \alpha_{jj'} + \pi\right) & d_{jj'} < d_{th} \\ (0,0) & otherwise \end{cases} \quad (12)$$

where w_A is the attractive factor, and w_R is the repulsive factor, $d_{jj'}$ indicates the distance between two sensors, and d_{th} is the optimal distance between two sensors. The parameter *R* determines the maximum absorptive distance between two sensors, and the angle $\alpha_{jj'}$ is the linear angle that connects the location of sensor *j* to the location of sensor *j'*, and is calculated relative to the *y*-axis. If the distance between the sensors is less than the optimal distance d_{th}, a repulsive force is applied between them, the value of which is inversely proportional to the distance between two sensors. If the distance between the sensors is greater than the optimal distance and they are within the absorptive radius of each other, an attractive force is applied between them.After calculating the attractive and repulsive forces between the sensors, the total force exerted on each sensor by all other sensors must be calculated. For this purpose, forces are first mapped from polar coordinates to the Cartesian coordinates. Then the vector of total force applied to each sensor is calculated by computing the average of the x components of the applied forces and the average of the y components of the applied forces. Finally, the new location of each sensor is determined by its old location and the total force vector applied to the sensor.

In our proposed algorithm, the virtual force operator is considered the local search operator, and is applied to the best particle of the population with local search probability P_{LS}, in each iteration of the PSO algorithm. Each time the local search operator manipulates a solution, the location of 10 random sensors is modified in 10 iterations. A limited number of iterations is considered to make the proposed local search operator lightweight in terms of computational load.

4. Experimental Results

In order to verify the effectiveness of the proposed algorithm, we implemented BVFPSO in MATLAB 2021b environment, and also implemented the four competitive algorithms GA, PSO, VFA, and FRED. For several initial random placements, the quality of improved deployment by the BVFPSO is compared with genetic algorithm (GA), single-objective particle swarm optimization (PSO), virtual-force algorithm (VFA) [5], and fuzzy redeployment algorithm (FRED) [12]. The GA algorithm is implemented using a single-point cross-over operator and the mutation operator of the BVFPSO. In the single-objective PSO algorithm, only the coverage rate was used as the objective function.

The experiments of this section are performed on 5 random initial deployments with 30, 40, 50, and 60 sensors in a target area with dimensions of [-2,+2] × [-2,+2]. These tests are executed on a desktop computer with an **Intel Core i5-7400 3.00 GHz** processor and 4.00 GB of main memory. To avoid the error caused by changing the randomly initial deployment in each run, all algorithms are run on the same random locations.

4.1 Parameter setting

In all experiments, the parameters of Table 1 are used. The perceptive radius of each sensor is considered r = 0.40. In BVFPSO and VFA, the optimal distance between two sensors is adjusted based on the perceptive radius of each sensor and similar to most prior research works with relation $d_{th} = \sqrt{3}r$, which allows a complete coverage of the target area without any hole with an appropriate number of sensors [22]. Considering the limited displacement model, the maximum displacement distance of each sensor in the BVFPSO is limited to d_{max} = 3r, which prevents excessive displacement of each sensor. The weight factor w_{cost} in BVFPSO determines the importance of the coverage rate against energy preservation in the redeployment process. Changing this parameter between 0 and 1 causes different solutions to be generated as the final solution by the proposed method, each of which provides a different coverage rate for different moved distances. The value of this parameter for each sample problem is empirically selected from values between 0.85 to 0.95.

Considering GA, the probability of cross-over is set to 0.7, and the probability of mutation is set to 0.1. In BVFPSO and the traditional PSO, the velocity vector values are limited to the range [-0.2, +0.2] to prevent some dimensions from becoming too large in the velocity vectors.

Table 1. Value of the simulation parameters of the BVFPSO algorithm.

Parameter	Value
Target area	[-2,+2] × [-2,+2]
Population size	$N = 50$
Maximum number of iterations	$M = 100$
Sensor perceptive radius	$r = 0.4$
Maximum displacement	$d_{max} = 3r$
Inertia factor	$w_k = 0.04$
Cognitive learning factor	$c_1 = 0.1$
Social learning factor	$c_2 = 0.1$
Distance between grid points	$d_g = 0.1$
Optimal distance	$d_{th} = \sqrt{3}r$
Maximum attractive distance	$R = 3r$
Attractive force coefficient	$w_A = 0.01$
Repulsive force coefficient	$w_R = 0.10$
Chance of mutation	$P_{mute} = 0.10$
Chance of local search	$P_{LS} = 0.10$

4.2 Simulation results and analysis

Figure 10 illustrates the performance of the proposed BVFPSO algorithm in improving the deployment of 40 sensors. The initial random deployment of 40 sensors is shown in Figure 10 (a), and the final improved deployment of the sensors is shown in Figure 10 (b). The area covered by the sensors is significantly increased in Figure 10 (b) compared to Figure 10 (a). The sensors in the improved deployment are evenly distributed in the target area, and have a lower overlay in Figure 10 (b) compared with Figure 10 (a).

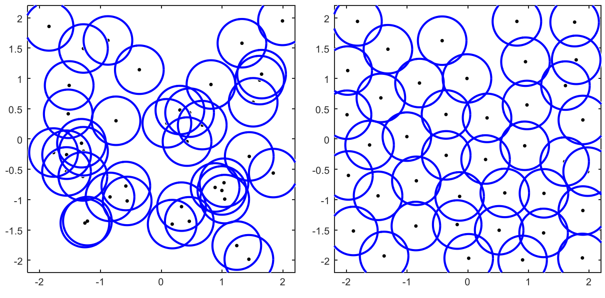

Figure 10. Redeployment of 40 sensors by the proposed BVFPSO algorithm considering coverage rate and battery energy consumption.

The results in terms of coverage rate and moved distance are summarized for several sample problems of 30, 40, 50, and 60 sensors in Table 2. For each sample instance, each redeployment algorithm was run 15 times. The reported coverages and moved distances are average of 15 runs.

Table 2. Comparison of five algorithms GA, PSO, VFA, FRED and BVFPSO in terms of coverage rate and moved distance for different number of sensors (p).

Algorithm	p	Initial Cover	Final Cover	Moved Distance
BVFPSO	30	0.63	**0.78**	0.32
GA	30	0.63	0.72	0.45
PSO	30	0.63	0.72	0.29
VFA	30	0.63	0.74	**0.28**
FRED	30	0.63	0.75	0.41
BVFPSO	40	0.67	**0.91**	0.44
GA	40	0.67	0.82	0.49
PSO	40	0.67	0.81	**0.33**
VFA	40	0.67	0.89	0.45
FRED	40	0.67	0.85	0.49
BVFPSO	50	0.82	**0.98**	0.29
GA	50	0.82	0.90	0.44
PSO	50	0.82	0.90	0.24
VFA	50	0.82	**0.98**	0.32
FRED	50	0.82	0.87	**0.21**
BVFPSO	60	0.85	**1.00**	**0.26**
GA	60	0.85	0.93	0.44
PSO	60	0.85	0.93	0.30
VFA	60	0.85	**1.00**	0.32
FRED	60	0.85	0.96	0.30

According to Table 2, our proposed BVFPSO algorithm achieved a higher coverage rate for all sample problems compared with GA, PSO, FRED, and VFA algorithms. Although GA and PSO, like BVFPSO, are population-based meta-heuristics, our proposed BVFPSO among them has obtained the highest coverage rate for all sample problems. This superiority is caused by the virtual-force local search operator in BVFPSO, which guides the sensor nodes to their best locations. In comparison with the single-solution meta-heuristic algorithms VFA and FRED, our BVFPSO algorithm has achieved an equal or higher coverage rate than the competing methods too. For most sample problems, in addition to BVFPSO, VFA has also achieved the highest coverage rates. This shows that the virtual-force operator plays an important role in achieving high coverage rates, and has been able to guide the sensors well to their optimal locations. In addition, in Table 2, the coverage rate of final deployment has increased with the increasing in the number of sensors. Since the coverage area is limited to [-2,+2] × [-2,+2], BVFPSO and VFA provided complete coverage of 100% in the target area as the number of sensors becomes 60 or more.

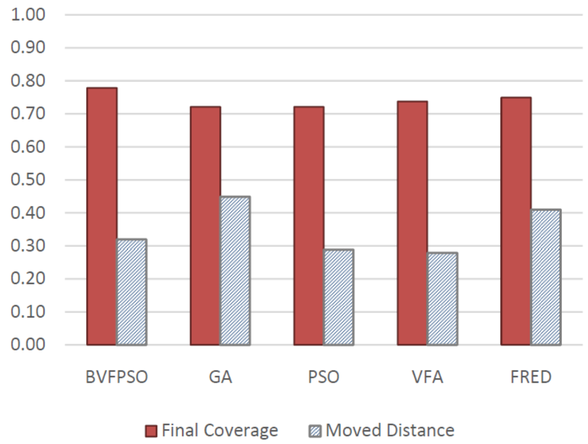

Figure 11. Comparison of five algorithms in terms of coverage rate and moved distance for p = 30 sensors.

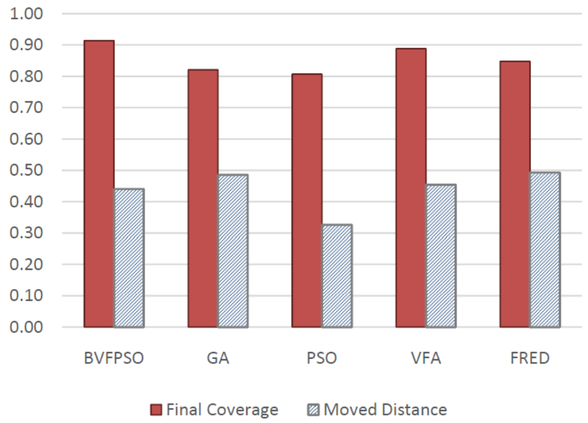

Figure 12. Comparison of five algorithms in terms of coverage rate and moved distance for p = 40 sensors.

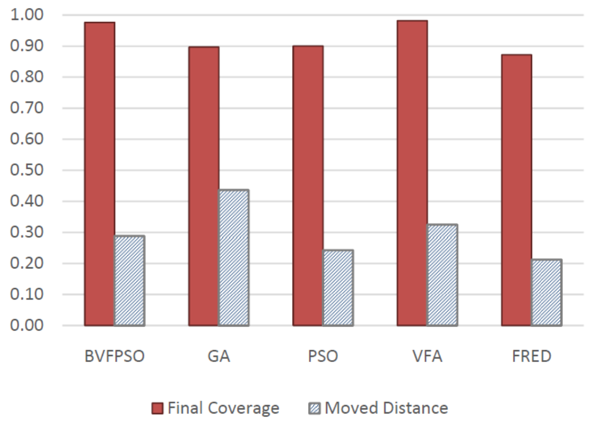

Figure 13. Comparison of five algorithms in terms of coverage rate and moved distance for p = 50 sensors.

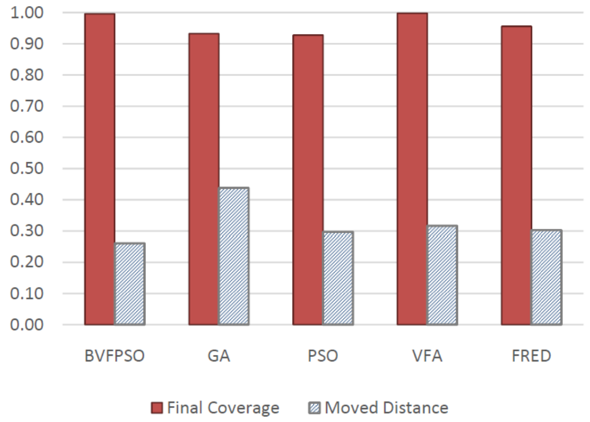

Figure 14. Comparison of five algorithms in terms of coverage rate and moved distance for p = 60 sensors.

Figures 11 to 14 present the results of evaluations in terms of the coverage rate and moved distance of the sensors to reach their final deployment from their initial deployment for different number of sensors (p). As it can be seen in these Figures, our proposed BVFPSO method has been able to achieve the highest coverage rate among other methods, and also moved the sensors less than most methods.

Figure 15 visually compares the computation time of the proposed BVFPSO for improving the deployment of different number of sensors.

The proposed BVFPSO algorithm uses a population-based search to achieve a higher coverage rate, and therefore, requires a little more time than VFA. This slight increasing in consumption time is tolerable due to the achievement of a higher coverage rate. Finally, the computation cost of the FRED algorithm has been very high and unacceptable in many applications. It requires calculating the Voronoi segmentation and the intersection of the circular coverage area of each sensor with its surrounding Voronoi partitions.

In summary, the results in terms of coverage rate, average moved distance, and computation time show that the proposed BVFPSO provides the highest coverage along with acceptable computation time and average moved distance. In the proposed BVFPSO algorithm, the local search operator based on virtual-forces can direct the sensors to better locations, and considering two goals at the same time would reduce the amount of sensor moved distances and preserve battery energy.

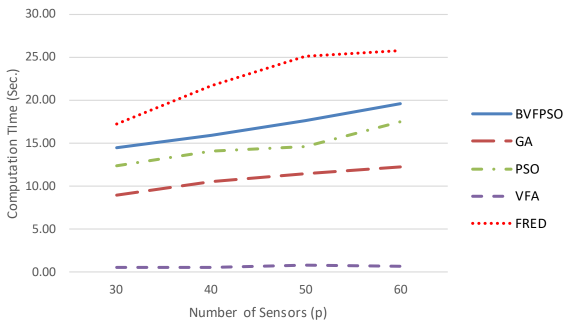

Figure 15. Comparison of computation time of the proposed BVFPSO algorithm with four algorithms GA, PSO, VFA, and FRED.

5. Conclusion

In this paper, a bi-objective virtual-force local search PSO (BVFPSO) algorithm was proposed to redeploy sensor nodes of the wireless sensor network monitoring a target area. It considers the two objectives of maximizing the coverage rate, and preserving the battery energy of the sensor nodes. The results of the simulations showed that the proposed algorithm by combining two objectives and using virtual-force local search could achieve a more efficient deployment in comparison to the competing algorithms. The use of local search helped BVFPSO to converge to higher coverage rates at a similar or lower average moved distance of sensors compared with the GA, PSO, VFA, and FRED algorithms. The extension of BVFPSO into more complex issues in wireless sensor networks could be the subject of the future research.

References

[1] R. Priyadarshi, B. Gupta, and A. Anurag, “Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues,” *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7333–7373,2020.

[2] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, “Applications of Wireless Sensor Networks: An Up-to-Date Survey,” *Applied System Innovation*, vol. 3, no. 1, 2020.

[3] A. Sangwan and R. P. Singh, “Survey on Coverage Problems in Wireless Sensor Networks,” *Wireless Personal Communications*, vol. 80, no. 4, pp. 1475–1500, 2015.

[4] H. I. Sweidan and T. C. Havens, “Coverage optimization in a terrain-aware wireless sensor network,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2016, pp. 3687–3694.

[5] Y. Zou and K. Chakrabarty, “Sensor Deployment and Target Localization in Distributed Sensor Networks,” *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, Feb. 2004.

[6] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, “Rearrangement of Mobile Wireless Sensor Nodes for Coverage Maximization Based on Immune Node Deployment Algorithm,” *Computers & Electrical Engineering*, vol. 43, no. C, pp. 76–89, Apr. 2015.

[7] B. Wang, “Coverage Problems in Sensor Networks: A Survey,” *ACM Computing Surveys*, vol. 43, no. 4, 2011.

[8] D. Liang, H. Shen, and L. Chen, “Maximum Target Coverage Problem in Mobile Wireless Sensor Networks,” *Sensors*, vol. 21, no. 1, p. 184, 2020.

[9] H. ZainEldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, “An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4177–4194, 2020.

[10] H. T. T. Binh, N. T. Hanh, L. V. Quan, N. D. Nghia, and N. Dey, “Metaheuristics for maximization of obstacles constrained area coverage in heterogeneous wireless sensor networks,” *Applied Soft Computing*, vol. 86, p. 105939, 2020.

[11] K. Tarnaris, I. Preka, D. Kandris, and A. Alexandridis, “Coverage and k-Coverage Optimization in Wireless Sensor Networks Using Computational Intelligence Methods: A Comparative Study,” *Electronics*, vol. 9, no. 4, Apr. 2020.

[12] A. Osmani, M. Dehghan, H. Pourakbar, and P. Emdadi, “Fuzzy-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks,” in *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 90–95, 2009.

[13] S. Liu, R. Zhang, and Y. Shi, “Design of coverage algorithm for mobile sensor networks based on virtual molecular force,” *Computer Communications*, vol. 150, pp. 269–277, 2020.

[14] S. Wang, X. Yang, X. Wang, and Z. Qian, “A Virtual Force Algorithm-Lévy-Embedded Grey Wolf Optimization Algorithm for Wireless Sensor Network Coverage Optimization,” *Sensors*, vol. 19, no. 12, 2019.

[15] J. Xie, D. Wei, S. Huang, and X. Bu, “A Sensor Deployment Approach Using Improved Virtual Force Algorithm Based on Area Intensity for Multisensor Networks,” *Mathematical Problems in Engineering*, vol. 2019, p. 8015309, 2019.

[16] X. Deng, Z. Yu, R. Tang, X. Qian, K. Yuan, and S. Liu, “An Optimized Node Deployment Solution Based on a Virtual Spring Force Algorithm for Wireless Sensor Network Applications,” *Sensors*, vol. 19, no. 8, p. 1817, 2019.

[17] M. Song, L. Yang, W. Li, and T. A. Gulliver, “Improving wireless sensor network coverage using the VF-BBO algorithm,” in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Aug. 2013, pp. 318–321.

[18] A. More and V. Raisinghani, “A survey on energy efficient coverage protocols in wireless sensor networks,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 428–448, 2017.

[19] Q. Zhang and M. P. Fok, “A Two-Phase Coverage-Enhancing Algorithm for Hybrid Wireless Sensor Networks,” *Sensors*, vol. 17, no. 1, 2017.

[20] N. A. A. Aziz, A. W. Mohemmed, M. Y. Alias, K. A. Aziz, and S. Syahali, “Coverage Maximization and Energy Conservation for Mobile Wireless Sensor Networks: A Two Phase Particle Swarm Optimization Algorithm,” *International Journal of Natural Computing Research (IJNCR)*, vol. 3, no. 2, pp. 43–63, 2012.

[21] Y. Qu and S. V. Georgakopoulos, “Relocation of wireless sensor network nodes using a genetic algorithm,” in *WAMICON 2011 Conference Proceedings*, Apr. 2011, pp. 1–5.

[22] E. Tuba, M. Tuba and M. Beko, “Mobile wireless sensor networks coverage maximization by firefly algorithm". *In Proceedings of 27th Conference Radioelektronik*a, pp. 1–5, 2017.

[23] X. Bai, S. Li, and J. Xu, “Mobile Sensor Deployment Optimization for k-Coverage in Wireless Sensor Networks with a Limited Mobility Model,” *IETE Technical Review*, vol. 27, no. 2, pp. 124–137, 2010.

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

[25] J. Pournazari, M. Alaei, and F. Yazdanpanah, “An Energy Efficient Autonomous Method for Coverage Optimization in Wireless Multimedia Sensor Networks,” *Wireless Personal Communications*, vol. 99, no. 2, pp. 717–736, 2018.

[26] N. Qin, and J. Chen, "An area coverage algorithm for wireless sensor networks based on differential evolution," *International Journal of Distributed Sensor Networks,* vol. 14, no. 8, 2018.

[27] H. Sheikhi, W. Barkhoda, "Solving the k-Coverage and m-Connected Problem in Wireless Sensor Networks through the Imperialist Competitive Algorithm," *Journal of Interconnec-tion Networks,* vol. 20, no. 1, p. 2050002, 2020.

[28] G. P. Gupta, and S. Jha, "Biogeography-based optimization scheme for solving the coverage and connected node placement problem for wireless sensor networks," *Wireless Networks,* vol. 25, no. 6, pp. 3167–3177, 2019.

[29] S. M. Hosseinirad, "Multi-layer Clustering Topology Design in Densely Deployed Wireless Sensor Network using Evolutionary Algorithms," *Journal of AI and Data Mining*, vol. 6, no. 2, pp. 297-311, 2018.

[30] R. Liu, C. Ma, W. Ma, and Y. Li, "A Multipopulation PSO Based Memetic Algorithm for Permutation Flow Shop Scheduling," *The Scientific World Journal*, vol. 2013, p. 387194, 2013.

یک الگوریتم بهینه‌سازی توده ذرات دو هدفه با جستجوی محلی و نیروی مجازی برای بهبود چیدمان حسگرها در شبکه‌های حسگر بی‌سیم

وحید کیانی[۱،*] و مهدی ایمان‌پرست[۲]

[۱] گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی،دانشگاه بجنورد، بجنورد، ایران.

[۲] گروه علوم کامپیوتر، دانشکده علوم پایه، دانشگاه بجنورد، بجنورد، ایران.

ارسال ۲۰۲۲/۰۵/۱۳؛ بازنگری ۲۰۲۲/۱۱/۱۴؛ پذیرش۲۰۲۳/۰۱/۱۱

چکیده:

در این مقاله یک الگوریتم توده ذرات دو هدفه با جستجوی محلی و نیروی مجازی (BVFPSO) برای بهبود چیدمان حسگرها در شبکه‌های حسگر بی‌سیم ارائه داده‌ایم، که به طور همزمان نرخ پوشش را افزایش داده و انرژی باتری حسگرها را حفظ می‌کند. در اکثر موارد، حسگرها در یک شبکه حسگر بی‌سیم ابتدا به صورت تصادفی در ناحیه هدف توزیع می‌شوند، و بعد چیدمان آن‌ها باید طوری اصلاح شود که برخی توابع هدف برآورده گردند. در الگوریتم BVFPSO پیشنهادی، از الگوریتم بهینه‌سازی توده ذرات (PSO) به عنوان الگوریتم فراابتکاری پایه، و از عملگر نیروی مجازی به عنوان جستجوی محلی استفاده شده است. تا جایی که ما اطلاع داریم، این اولین بار است که یک الگوریتم بهینه‌سازی توده ذرات دو هدفه با یک عملگر نیروی مجازی ترکیب شده است تا ضمن بهبود نرخ پوشش شبکه، انرژی باتری حسگرها را نیز حفظ کند. نتایج ارزیابی روی تعدادی چیدمان تصادفی اولیه با تعداد متفاوتی حسگر نشان می‌دهد که الگوریتم BVFPSO با ترکیب دو هدف و بکارگیری جستجوی محلی نیروی مجازی قادر است چیدمان کارآمدتری را نسبت به الگوریتم‌های PSO، GA، FRED و VFA تولید کند، به طوری که همزمان حداکثر نرخ پوشش و حداقل مصرف انرژی حاصل گردد.

کلمات کلیدی:پوشش حداکثری، بهبود پوشش، بهینه‌سازی توده ذرات، عملگر نیروی مجازی، نرخ پوشش.
