Solving Point Coverage Problem in Wireless Sensor Networks Using Whale Optimization Algorithm

Mahnaz Toloueiashtian, Mehdi Golsorkhtabaramiri*, and Seyed Yaser Bozorgi Rad
Department of Computer Engineering, Babol Branch, Islamic Azad University, Babol, Iran

Abstract: *Todays, dynamic power management methods that decrease the energy use of sensor networks after their design and deployment are of paramount importance. In Wireless Sensor Networks (WSN), coverage and detection quality are one aspect of service quality and power consumption reduction aspect. The aim of the coverage problem is to monitor at least one node at each point in the targeted area and is divided into three categories: border, area, and point coverage. In point coverage, which is our interest, the problem is to cover specific points of the environment scattered on the surface of the environment; their position is decided on and called the goal. In this paper, a new metaheuristic algorithm based on Whale Optimization Algorithm (WOA) is proposed. The proposed algorithm tries to find the Best Solution (BS) based on three operations exploration, spiral attack, and siege attack. Several scenarios, including medium, hard and complex problems, are designed to evaluate the proposed technique, and it is compared to Genetic Algorithm (GA) and Ant Colony Optimization (ACO) based on time complexity criteria in providing a suitable coverage, network lifetime, energy consumption. The simulation results show that the proposed algorithm performs better than the compared ones in most scenarios.*

Keywords: *Wireless sensor networks, point coverage, network lifetime, optimization, WOA.*

Received July 10, 2020; accept December 1, 2020
https://doi.org/10.34028/iajit/18/6/10

1. Introduction

Nowadays, the significance of Wireless Sensor Networks (WSN) has increased in parallel with their use in different areas [1, 13]. A set of sensors, called WSN, form high-power networks capable of sampling, processing local values, sending them to other sensors, and eventually to the main observer (client) [2].

Addressing these limitations requires the design of new telecommunication methods to raise the bandwidth required for each user and the design of powered protocols for efficient energy usage [10]. One of the issues in designing these protocols is the coverage and quality of WSN [7, 14].

The point coverage scenario has many military applications. In this scenario, several targets with known locations to be controlled are considered [11]. A large number of sensors are randomly propagated very close to the target and transmit the obtained information to the central processor node so that at any given moment, each target must be controlled by at least one sensor, presuming that each sensor is capable of controlling all the targets within its sensory range. Another way is to reduce the number of active sensors in the coverage area. One way to extend the lifetime of the sensor network by saving energy is to split the sensor set into several discrete sets. This division should be such that each set fully covers all the targets. These sets are activated sequentially so that only one set is active at a time [12].

On the assumption that each optimization method has its advantages in solving problems and considering the problem conditions, one can use the Whale Optimization Algorithm (WOA) to find the more optimal solution than other algorithms. This algorithm has an important and practical place among different optimization methods, so it is expected that the proposed technique will have better results in quality, stability, and energy usage balance and make better choices to prolong the lifespan in order that its results compete with that of other ones. The purpose of this study is to propose a protocol whose energy is efficient to extend the lifetime of a network, and we want to design a new cost function in order to optimize energy usage using WOA. The proposed protocol is designed to reduce the monitoring energy usage of a point coverage network where several targets are monitored and to prevent additional overhead in the transmission of information.

The structure of the paper is organized as follows: in section 2, related works are investigated. Section 3 is presented the architecture and network assumptions. The WOA and proposed algorithm are presented in section 4. Simulation results and conclusion is provided in sections 5 and 6, respectively.

2. Related Work

2.1. Point Coverage

The probability that an event in an area will be identified at a specific time is called network coverage power [8]. A very important issue in using WSN along with coverage is connectivity [21]. The points of interest are predefined, their locations are known, and they are called targets. In point coverage, even if it can cover all the environmental targets, no work has done unless there is a connection between sink and sensors to transmit information [6].

Since battery replacement is not suitable in many applications, low power usage is one of the essential requirements in this kind of network, and the lifespan of any sensor can be impressively reduced by optimizing power usage. One way to decrease energy usage is by reducing the number of sensors while maintaining target coverage [16]. In [20], a technique for saving power use is presented, in which the sensors are first divided into two groups, at any time, only one of them is active, and the process of activation and inactivity repeats alternately.

Wang *et al.* [17] considered the specific case of continuous coverage targets in which targets are prioritized. It is assumed that each target must be covered at least by a certain number of sensors, and finally, the problem is solved by GA. Xu and Yao [19] proposed reducing the number of sensors and reducing energy usage based on GA. This method is an advantage over other techniques. Wang *et al.* [18] presented a technique for achieving content coverage with random distribution, and sensor density is a variable parameter in the network. Optimal sensor density is obtained by defining a high boundary for the probability of point coverage and defining a relationship between the sensor density and the non-covered region's average. This results in a significant reduction in the amount of energy consumed.

Tian and Georganas [15] presented a procedure in which nodes become active and inactive in a distributed manner to obtain the desired coverage range and does not change.

Rostami *et al.* [9] solved the point coverage problem in WSN using the Gravitational Search Algorithm (GSA). The proposed algorithm was compared with the other three algorithms, and its superiority is confirmed over other ones and could reduce power consumption in the network.

3. Architecture and Network Assumptions

This section will give a detailed overview of the node-based WSN architecture based on the coverage point and problem formulation.

Figure 1 shows the network assumptions diagram an overview of the problem space of node-based WSN with a target point coverage approach. As shown in Figure 1, the problem space consists of a two-dimensional Euclidean space that includes N nodes identifying the environment marked in blue and K target points to be covered by the nodes, and U nodes known as sink nodes or super nodes. The main purpose of the problem is to cover the set of targets using nodes in the networks and send the investigated information to the super nodes and ultimately send the information of super nodes to the control center for further investigation. To this end, the network has certain limitations and assumptions, will be discussed below.

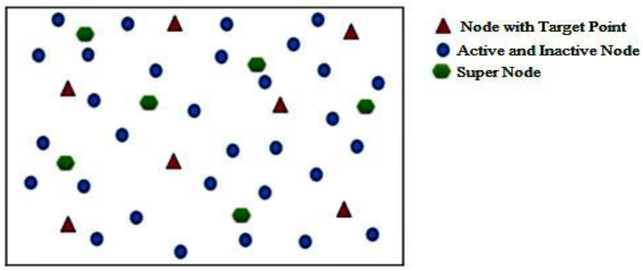

Figure 1. The network assumptions diagram of point coverage WSN.

Table 1 shows the assumptions and goals of the network.

Table 1. Assumptions and goals of the network.

Network Assumptions	• The network has a two-dimensional space X and Y. • The distance between points or nodes is based on Euclidean distance
Target Points Assumptions	• Each point is at the coordinate (X, Y) the problem space. • The number of target points is predetermined. • Each point in the problem space is fixed
Super Node Assumptions	• The network has K super node • The coordinate of each super node is predetermined and fix and is in a two-dimensional space X and Y. • Each super node has j joule energy at first. • Each node has a RC neighborhood radius • Each node has RT transmission distance limitation. • There is at least one connection between the super nodes in the network
Nodes Assumptions	• The network has K super node. • The coordinate of each super node is predetermined and fix and is in a two-dimensional space X and Y. • Each node has j joule energy at first. • Each node has a RC neighborhood radius • Each node has RT transmission distance limitation. • Each node can be active or inactive; if inactive, no energy is consumed. • Each node can be a coverage node or an intermediate node.
Problem Limitations	• The whole set of nodes must be covered by at least one node. • There must be at least one connection (direct or indirect) between the active node and the super node.
The Goal of the Problem	• All points must be covered. • Energy usage should be reduced during the information transmission. • Network lifetime should be raised.

4. Whale Optimization Algorithm (WOA)

WOA was proposed by Mirjalili and Lewis [5] for optimizing numerical problems. The algorithm simulates the intelligence hunting behavior of humpback whales. This foraging behavior is called the bubble-net feeding method that is only be observed in humpback whales. The whales create the typical bubbles along a circular path while encircling prey during hunting. In order to perform optimization, the mathematical model for spiral bubble-net feeding behavior is given as follows:

4.1. Encircling Prey

Current best search agent position to be the target prey or close to the optimum point, and other search agents will try to update their position towards the best search agent. This behavior is formulated as the following equations:

$$D = |\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)|, \quad (1)$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D}, \quad (2)$$

Where t indicates the current iteration, X^* is the position vector of the best solution has been obtained such a far iteration t, $\vec{X}$ is the position vector of each agent, | | is the absolute value, and is an element-by-element multiplication. The coefficient vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a}.r - \vec{a} \quad (3)$$

$$\vec{C} = 2r, \quad (4)$$

Where $\vec{a}$ is linearly decreased from 2 to 0 throughout the iteration, and r is *a* random number [0,1].

4.2. Bubble-net Attacking Method

The Bubble-net strategy is a hybrid of combined two approaches that can be mathematically modeled as follows:

a) *Shrinking Encircling Mechanism*

This behavior of Whales simulated by decreasing the value of $\vec{a}$ in Equation (3). Note that the fluctuation range of $\vec{A}$ is also decreased by $\vec{a}$. In other words, $\vec{A}$ is a random value in the interval [-a, a] where a is decreased from 2 to 0 over the course of iterations. Setting random values for $\vec{A}$ in [−1, 1], the new position of a search agent can be defined anywhere between the original position of the agent and the position of the current best agent.

b) *Spiral Updating Position*

In this approach, *a* spiral equation is created between the position of whale and prey to simulate the helix-shaped movement of humpback whales as follows:

$$\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \vec{X}(t)| \quad (5)$$

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos 2\pi l + \overrightarrow{X^*}(t) \quad (6)$$

Where $\overrightarrow{D}'$ is the distance between the whale and prey, b is constant defines the logarithmic shape, l is random in [−1,1] and is an element-by-element multiplication.

Indeed, humpback whales swim along *a* spiral-shaped path and at the same time within the shrinking circle. Assuming a probability of 50%, choosing either the shrinking encircling movement or the spiral model movement is simulated during iterations of the algorithm. It means that:

$$X(t+1) = \begin{cases} X^*(t) - A.D & if\ p < 0.5 \\ D.\acute{e}^{bl}.\cos 2\pi l + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \quad (7)$$

Where p is a random number in [0, 1].

4.3. Search for Prey

Almost all metaheuristic algorithms explore the optimum using random selection. In the bubble net method, the position of the optimal design is not known, so humpback whales search for prey randomly. In contrast to the exploitation phase with $\vec{A}$ in the interval [−1, 1] in this phase consider, $\vec{A}$ be a vector of the random values greater than 1 or less than −1. With this assumption, the search agent able to move far away from a reference whale. In return, the position of the search agent will be updated according to randomly chosen from the search agent, instead of the best search agent found so far. These two actions formulated as follows:

$$D = |C.X_{rand} - X|, \quad (8)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - A.D \quad (9)$$

Where X_{rand} is a random position vector.

4.4. The Proposed Algorithm

4.4.1. Parameter Initialization

In the WOA, to solve the point coverage problem in the WSN, there is a set of parameters that are set according to Table 2.

Table 2. Parameters of the WOA algorithm.

Variable Name	Value
N: the number of initial populations	Initialized according to the problem space
r: coefficient of search operators	Random initialization in (0,1)
a: coefficients of Search Operators	Random initialization in (0,1)
K: termination condition	The specified number of iterations according to the problem space
P: search type selection operator	Random initialization in (0,1)
RS: sensory range	Pixel-based initialization
RC: sensory range	Pixel-based initialization
$\mathbf{E_i}$	The primary energy of nodes (j)

4.4.2. Solution Display

One of the most important steps in a population-based metaheuristic algorithm is displaying solution coding. In the proposed algorithm, a one-dimensional array with the total number of nodes is used to represent the solution, each index of this array representing the sensor number, and the content of each entry of this array is 1 or 0, 1 representing active node and this node can do the coverage process and 0 representing the inactive node. Consider Figures 2 and 3 below for a better understanding.

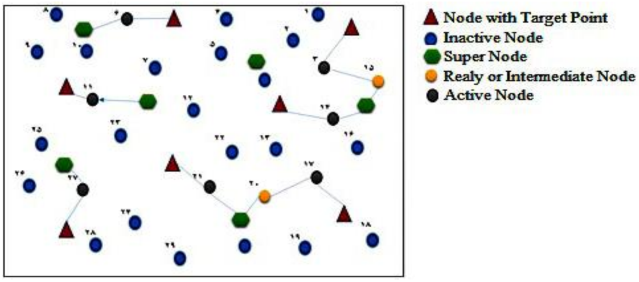

Figure 2. The overall view of the problem space containing 29 coverage points.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	0

Figure 3. A sample of the solution in the problem space of Figure 2.

As shown, sensors are numbered at first; the array is composed of the total number of sensors and the contents of each array dilled with 0 and 1 randomly. As specified, 7 of the 29 sensors are considered active sensors, and the rest as relay nodes. To understand how to create a solution, consider the following steps.

1) Network sensors (blue nodes) will be numbered individually.
2) An array (its length is equal to the number of numbered sensors) will be initialized.
3) A random value of 0 or 1 will be assigned to each array index.0 indicates inactive node, and 1 indicates active node.
4) Each active sensor will be connected to the nearest super node located within its neighborhood radius.
5) Other active nodes, if not connected to the super node, will be connected to the next closest node known as the intermediate node, and the intermediate node will be connected to the nearest super node.
6) Repeat steps 4 and 5 until all active nodes have a connection to the super node.

Given the above explanation for creating a solution, we will randomly create *n* solutions, and each solution represents a sample solution in the problem space, then we evaluate the fitness of each solution.

4.4.3. Fitness Evaluation

At this step, after creating a solution, the fitness of each solution is computed in the following steps:

1) The number of target points not covered by any node is considered an error, and its value is equal to *PN*.
2) The number of active nodes is *C*, in fact the value of *C* is equal to the sum of the number of active nodes and relay nodes.
3) The amount of energy consumed by all nodes is *SumE*. The amount of energy consumed to transmit each bit is computed from Equation (10).

Each node consumes E_s energy for transmitting L bit data with d distance from itself.

$$E_s = \begin{cases} L \times E_{elect} + L \times E_{fs} \times d^2 & d < d_{co} \\ L \times E_{elect} + L \times E_{mp} \times d^4 & d \geq d_{co} \end{cases} \tag{10}$$

Where, E_{elect} is the energy required to activate the electronic circuits of the transmitter (in nanoscale), d_{co} is a threshold. E_{mp} and E_{fs} are the activation energy of the amplifier when the distance is greater and lower than threshold.

Also, the amount of energy consumed for receiving L bits per receiver is equal to E_r, obtained from Equation (11).

$$E_r = L \times E_{elect} \tag{11}$$

Finally, *SumE* equals the sum of E_r and E_s of all nodes.

Then the fitness of each solution is obtained from Equation (12) based on a weighting method.

$$\text{Fitness}_i = \text{W1} * \text{SumE} + \text{W2} * \text{PN} + \text{W3} * \text{C} \tag{12}$$

*W*1, *W*2, and *W*3 are weighing in (0, 1) so that the sum of all of them equals 1. According to the fitness formula, a solution with the least fitness value is selected as the optimal solution.

4.4.4. Initial Population Generation

At this point, a population of n solutions is generated randomly, and then the fitness of each solution is calculated.

4.4.5. Selecting the Best Solution

After calculating the fitness, the *BS* with the least fitness function is selected as the candidate or *X**, after creating the initial population, the algorithm starts searching in the problem space and updating the solutions. Each of these processes is explained below.

4.4.6. Applying Siege Attack Operator

To apply this operator, the algorithm tries to move to the candidate solution to improve the position of the current solution. To better understand this method, operator, and how-to discretization, consider Figure 4 below.

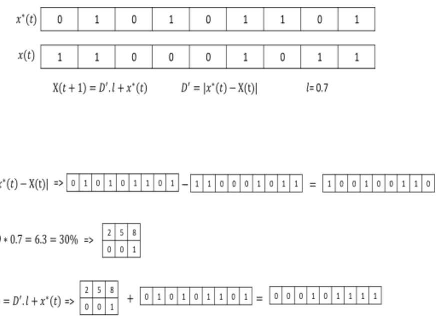

Figure 4. How does work siege operator.

As shown in Figure 4, suppose we have two solutions: candidate solution *X** and current solution

X(t) at iteration *t* and the situation of the current solution should be updated. There are three steps for updating using siege operator which is discretized according to the siege formula. These steps are as follow:

- *Step* 1. The difference of two solutions or two arrays must be obtained first. The difference between the two solutions, *XOR* of each index of the array is computed pairwise. In other words, if node 1 is active in the candidate solution and inactive in the current solution, the result of *XOR* is 1. We do this for all entries to obtain vector *D*.
- *Step* 2. At this step, a random real integer *L* ranging from 0 to 1 is multiplied by each iteration. It is assumed that the result is (9*7.0= 6.3) because the decimal value is 0.3, 30% of the array is chosen as candidate solution; here, nodes 2, 5, and 8 are chosen.
- *Step* 3. At this step, the vector obtained in the previous step will be applied to the candidate solution vector. In other words, cells 2, 5, 8 are filled with the values of the vector obtained in step 2, and the new position of the current solution is obtained.

After updating the situation of the current solution, the fitness of it is computed and is considered as a candidate solution if it is better than the previous candidate solution.

4.4.7. Exploration Operator

In this operator, we move toward a random solution in the population, for this we first select a random solution based on the roulette wheel selection method, and then the current solution of this random solution is updated. To better understand this method, consider Figure 5.

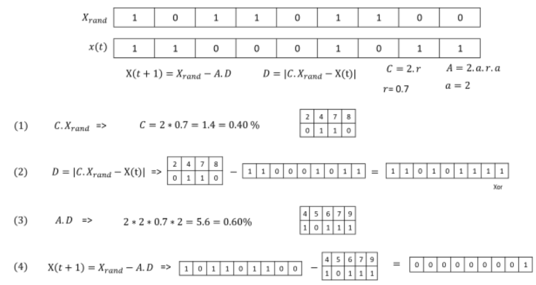

Figure 5. How does work exploration operator.

- *Step* 1. At this point, after selecting a random solution, a real number *C* will be multiplied by the random solution vector is selected. It is assumed here that the result is 1.4. Because the decimal value is 0.4, 40% of the random solution is chosen randomly (Figure 4).
- *Step* 2. Here, the difference between the vector obtained in step 1 and the current solution is computed the same way as before and based on the *XOR* function with respect to the entries of the solution vector.
- *Step* 3. At this point, we multiply the real numbers on vector *D*, which is the multiplication of a real number by a vector.
- *Step* 4. Here the difference between an obtained vector in step 3 and the random solution is computed, and the new situation of the current solution is obtained.

4.4.8. Spiral Attack Operator

This operator is similar to the exploration operator except that the solution's situation is obtained based on the candidate solution *X**. Figure 6 shows the overall view of this operator.

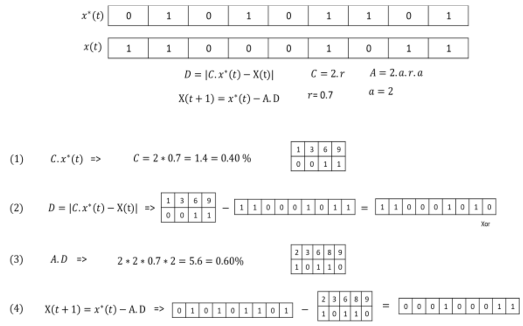

Figure 6. How does work spiral operator.

4.4.9. Termination Condition

A certain number of iterations is considered as a termination condition, and its value is determined by the size of the problem space. In other words, the greater the number of target points and sensors, the more space the problem will have, and after reaching a certain number of iterations, the algorithm terminates, and the *BS* is represented.

5. Simulation Results

After presenting the proposed algorithm and outlining its advantages in the previous section, it is necessary to evaluate the proposed procedure against the widely used metaheuristic algorithms in this area. To evaluate the proposed procedure, several valid test data are first generated based on different parameters in WSN for the point coverage problem. Then the proposed method is compared with ACO [4] and GA [3].

We investigate the algorithms based on different criteria, such as time complexity in providing proper coverage, network lifetime, power consumption, and the number of transmitted and received packets, coverage quality. Results show that the proposed method has always performed better than compared ones and gained acceptable results.

5.1. Test Data

Test data used for evaluating the proposed algorithm includes two 2-dimensional space 300×300 and 400×400, in which a set of sensors, including coverage points, target points, and super nodes with different parameters, are fixed.

Table 3 shows the parameters of each sensor in the network for two scenarios: 300×300 and 400×400. Then, for both scenarios, multiple networks with different nodes are considered. Each network is known as WSN_N_M, where *N* is the size of the network space between 300 and 400, *M* is the number of nodes in network space. Selected algorithms for comparison are performed for a certain number of iterations. These test data are shown in Table 4.

Table 3. Parameters of the sensors in the network.

Parameters	Value	Unit
Super Nodes Location	Random	-
Nodes Location	Random	-
Nodes Initial Energy	0.1 j	Jules
Super Node Initial Energy	1 j	Jules
The Size of Transmitted Packet	4000	Bit
Threshold of Transmitting Information (Sensor Transfer Radius)	[50-80] for the first scenario [90-100] for the second scenario	Meter
Neighborhood Radius	80 for the first scenario 100 for the second scenario	Meter
Energy Consumed at Distances Below Threshold	10×10^{-9}	Nano
Energy Consumed at Distances Greater Than Threshold	0.0013×10^{-9}	Nano
Energy Needed to Activate Circuits	50×10^{-12}	Pico

According to the test data provided, each of the compared algorithms should receive this test data and cover the network as best as possible and provide suitable solutions to improve the network lifetime, network quality in terms of the number of transmitted and received packets. Below we will introduce the compared algorithms and their parameters.

Table 4. Properties of test data designed for two spaces of the problem.

Problem Size	Network Name	Number of Nodes	Number of Super Nodes	Number of Target Nodes
300×300	WSN_300_50	50	5	20
300×300	WSN_300_100	100	10	30
300×300	WSN_300_150	150	12	50
300×300	WSN_300_200	200	20	60
300×300	WSN_300_250	250	25	70
300×300	WSN_300_300	300	28	80
400×400	WSN_400_50	50	5	10
400×400	WSN_400_100	100	10	20
400×400	WSN_400_150	150	15	30
400×400	WSN_400_200	200	20	40
400×400	WSN_400_250	250	22	50
400×400	WSN_400_300	300	30	80

5.2. Compared Algorithms

After designing test data, we introduce the parameters of the compared algorithms for testing each test data. These algorithms are ACO algorithm [4] and GA [3]. We choose these two algorithms because GA is widely used in most problems, and ACO could recently provide acceptable solutions. These two algorithms also belong to the family of population-based algorithms and have sufficient similarity to the proposed WOA. However, the two algorithms have different strategies for problem-solving. Table 5 shows the parameters of these algorithms.

Table 5. Parameters of the proposed and compared algorithms.

Parameters	GA [3]	ACO [4]	Proposed Algorithm (WOA)	Values
Size of the population	✓	✓	✓	100=(50,100,150) -300
Maximum number of iterations	✓	✓	✓	600-100
Termination condition	When the energy of the first nodes is left			

We implemented our proposed algorithm WOA using C# programming language to solve the problem and validate our proposed algorithm. To compare the proposed algorithm with similar algorithms we run NS2 network simulation application on a computer with an Intel (R) Pentium (R) 4 CPU of 3.00GHz. The simulated test was created based on time complexity, network lifetime, number of received bits, and coverage quality criterion. The simulation results is given in the following subsection.

5.2.1. Simulation Results Based on Network Lifetime

This criterion indicates the number of network rounds in the coverage and transmission phases. This criterion shows the number of network rounds, to achieve this criterion, each algorithm performs the coverage phase and the transmission phase until at least one sensor has left its energy, in other words, the lifetime of the network will be measured based on the first death of a sensor.

Table 6 shows the evaluation based on the lifetime of the network for a problem space of 300×300.

Table 6. Network lifetime in 300×300 space.

Problem Size	GA	ACO	WOA
WSN_300_50	76	19	99
WSN_300_100	59	12	83
WSN_300_150	49	11	79
WSN_300_200	35	10	58
WSN_300_250	36	8	47
WSN_300_300	18	7	48

Figure 7 shows the evaluation based on the lifetime of the network for a problem space of 300×300. The vertical axis represents the number of rounds, and the horizontal axis represents the set of data tests in space 300×300. As it is known, as the number of network nodes increases, the problem space becomes harder, and network lifetime is reduced.

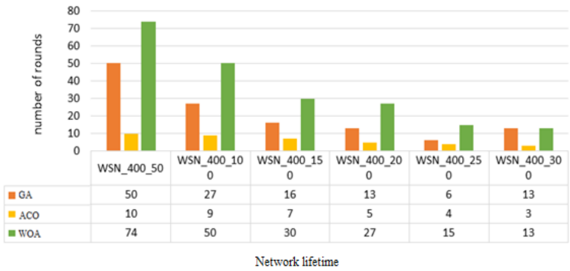

Figure 7. Network lifetime in 300×300 space.

Table 7 shows the evaluation based on the lifetime of the network for a problem space of 400×400.

Table 7. Network lifetime in 400×400 space.

Problem Size	GA	ACO	WOA
WSN_400_50	50	10	74
WSN_400_100	27	9	50
WSN_400_150	16	7	30
WSN_400_200	13	5	27
WSN_400_250	6	4	15
WSN_400_300	13	3	13

Figure 8 shows the network lifetime for a problem space of 400×400.

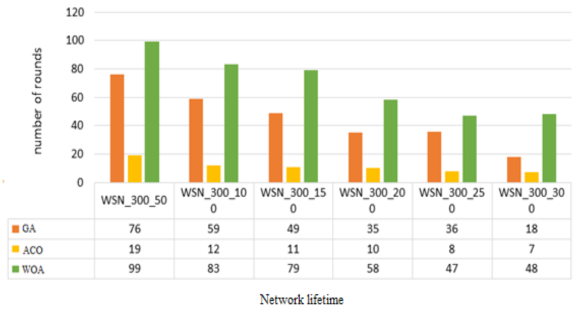

Figure 8. Network lifetime in 400×400 space.

Like previous problem space, here also proposed algorithm WOA performs better than compared algorithms. However, as it is evident, by increasing the dimensions of the problem space, the complexity of the problem increases, and network lifetime reduces.

5.2.2. Results Based on the Number of Received Bits

This criterion actually checks the volume of information received from the sensors in the problem space by the center unit. In other words, it is assumed that in each round, active sensors send the information to the manager, and then this information is sent to the center unit. The more information sent from sensors to the network is always best, and we can obtain more information from the target points, and the more information received from the environment indicates that the algorithm performs better, the coverage process and network lifetime are longer.

Table 8 shows the evaluation based on the lifetime of the network for a problem space of 300×300.

Table 8. The received bits at test data of 300×300.

Problem Size	GA	ACO	WOA
WSN_300_50	3049	684	4083
WSN_300_100	5013	946	7240
WSN_300_150	6325	983	10596
WSN_300_200	6089	1177	10509
WSN_300_250	7917	1491	10768
WSN_300_300	4720	1816	13525

Figure 9 shows the evaluation based on the number of received bits for the provided test data.

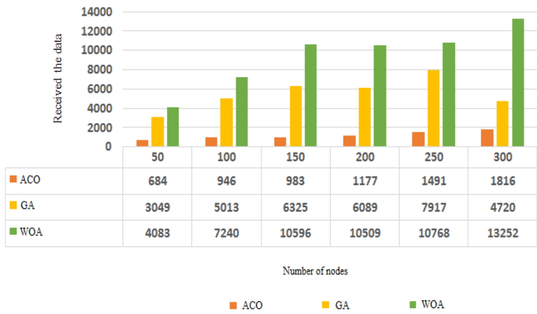

Figure 9. The received bits at test data of 300×300.

6. Conclusions

In this paper for solve the point coverage problem used WOA algorithm. In the proposed algorithm, by to discretizing the WOA in the problem space, the operators of the algorithm including exploration, Spiral Attack and Siege Attack are developed and an efficient and effective algorithm for point cover optimization problem is proposed. The proposed algorithm has been evaluated in three different scenarios against GA and ACO. Evaluation results based on the lifetime criterion showed that the proposed procedure has always performed better than the two compared algorithms in most scenarios. Since the more the number of sensors in the coverage problem, the more difficult the problem is and algorithm get stuck in a local optimum, so achieving a good coverage at low time requires a more efficient algorithm. Therefore, in the future it is suggested to combine the proposed technique with Local Search Algorithms (LSA) so that the local search in the problem space can always maintain the scatter of the solutions and also avoid local optimum. For future work, we can combine the proposed algorithm with another metaheuristic algorithm and present a new and more powerful algorithm to solve the point coverage problem in WSN. The above algorithm can also be used for problems such as regional and border coverage.

References

[1] Bozorgi S., Rostami A., Hosseinabadi A., and Balas V., "A New Clustering Protocol for Energy Harvesting-Wireless Sensor Networks," *Computers and Electrical Engineering*, vol. 64, pp. 233-247, 2017.

[2] Han T., Bozorgi S., Orang A., Hosseinabadi A., Sangaiah A., and Chen M., "A Hybrid Unequal Clustering Based on Density with Energy Conservation in Wireless Nodes," *Sustainability*, vol. 11, no. 3, pp. 1-26, 2019.

[3] Harizan S. and Kuila P., "Coverage and Connectivity Aware Energy Efficient Scheduling in Target Based Wireless Sensor Networks: an Improved Genetic Algorithm-Based Approach," *Wireless Networks*, vol. 25, pp. 1995-2011, 2019.

[4] Lee J. and Lee J., "Ant-Colony-Based Scheduling Algorithm for Energy-Efficient Coverage of WSN," *IEEE Sensors Journal*, vol. 12, no. 10, pp. 3036-3046, 2012.

[5] Mirjalili S. and Lewis A., "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.

[6] Nguyen T., Phan T., Nguyen H., Aimtongkham P., and So-In C., "An Efficient Distributed Algorithm for Target-Coverage Preservation in Wireless Sensor Networks," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 453-466, 2021.

[7] Rostami A., Badkoobe M., Mohanna F., Hosseinabadi A., and Balas V., "Imperialist Competition based Clustering Algorithm to Improve the Lifetime of Wireless Sensor Network," *in Proceedings of 7th International Workshop Soft Computing Applications*, Arad, pp. 189-202, 2016.

[8] Rostami A., Badkoobe M., Mohanna F., keshavarz H., Hosseinabadi A., and Sangaiah A., "Survey on Clustering in Heterogeneous and Homogeneous Wireless Sensor Networks," *The Journal of Supercomputing*, vol. 74, pp. 277-323, 2018.

[9] Rostami A., Bernety H., and Hosseinabadi A., "A Novel and Optimized Algorithm to Select Monitoring Sensors by GSA," *in Proceedings of 2nd International Conference on Control, Instrumentation and Automation*, Shiraz, pp. 829-834, 2011.

[10] Sangaiah A., Bian G., Bozorgi S., Suraki M., Hosseinabadi A., and Shareh M., "A Novel Quality of Service Aware Web Services Composition using Biogeography-based Optimization Algorithm," *Soft Computing*, vol. 24, pp. 8125-8137, 2020.

[11] Saemi B., Hosseinabadi A., Kardgar M., Balas V., and Ebadi H., "Nature Inspired Partitioning Clustering Algorithms: A Review and Analysis," *in Proceedings of 7th International Workshop Soft Computing Applications*, Arad, pp. 96-116, 2016.

[12] Sangaiah A., Sadeghilalimi M., Hosseinabadi A., and Zhang W., "Energy Consumption in Point-Coverage Wireless Sensor Networks via Bat Algorithm," *IEEE Access*, vol. 7, pp. 180258-180269, 2019.

[13] Samara G. and Aljaidi M., "Aware-Routing Protocol using Best First Search Algorithm in Wireless Sensor," *The International Arab Journal of Information Technology*, vol. 15, no. 34, pp. 592-598, 2018.

[14] Tavakkolai H., Yadollahi N., Yadollahi M., Hosseinabadi A., Rezaei P., and Kardgar M., "Sensor Selection Wireless Multimedia Sensor Network using Gravitational Search Algorithm," *Indian Journal of Science and Technology*, vol. 8, no. 14, pp. 1-6, 2015.

[15] Tian D. and Georganas N., "A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271-290, 2003.

[16] Wu H., Li Q., Zhu H., Han X., Li Y., and Yang B., "Directional Sensor Placement in Vegetable Greenhouse for Maximizing Target Coverage Without Occlusion," *Wireless Networks*, vol. 26, pp. 4677-4687, 2020.

[17] Wang X., Xing G., Zhang Y., Lu C., Pless R., and Gill C., "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," *in Proceedings of ACM Conference on Embedded Networked Sensor System*, New York, pp. 28-39, 2003.

[18] Wang W., Sinivasan V., Chua K., and Wang B., "Energy-efficient Coverage for Target Detection in Wireless Sensor Networks," *in Proceedings of 6th International Symposium on Information Processing in Sensor Networks*, Cambridge, pp. 313-322, 2007.

[19] Xu Y. and Yao X., "A GA Approach to the Optimal Placement of Sensors in Wireless Sensor Networks with Obstacles and Preferences," *in Proceedings of 3rd IEEE Consumer Communications and Networking Conference*, Las Vegas, pp. 127-131, 2006.

[20] Yarinezhad R. and Hashemi S., "A Sensor Deployment Approach for Target Coverage Problem in Wireless Sensor Networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-16, 2020.

[21] Zameni M., Rezaei A., and Farzinvash L., "Two-Phase Node Deployment for Target Coverage in Rechargeable WSNs Using Genetic Algorithm and Integer Linear Programming," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 4172-4200, 2021.

Mahnaz Toloueiashtian received the M.Sc. in Software Engineering, from Islamic Azad University Ayatollah Amoli, Iran, 2016. She is currently a Ph.D. Candidate in Computer Engineering from the Islamic Azad University, Babol Branch, Babol, Iran. She researches interests are in WSN, Cloud Computing, and Image Processing.

Mehdi Golsorkhtabaramiri received his M.Sc. in Computer Systems Architecture Engineering. He received his Ph.D. in Computer Engineering from the Science and Research Branch, Islamic Azad University, Tehran, Iran. From 2011 until now, he has been working as a faculty member at the Islamic Azad University, Babol Branch, Iran. His current research interests include Radio Frequency Identification (RFID) systems, Wireless Sensor Network (WSN), and Wireless Communications.

Seyed Yaser Bozorgi Rad was born in Iran in 1980. He graduated in PhD of Computer Science from University Teknology Malaysia in 2012. His degree in Master of Science is in Information Technology from UTM as well. Software Projects management, Advanced Networks Security, Computing Solutions and Distributed Systems are some of his interested research areas. Moreover, He has been a faculty member of IAU Babol branch since 2014.