Hybrid Algorithm for Node Deployment with the Guarantee of Connectivity in Wireless Sensor Networks

Hicham Deghbouch[1] and Fatima Debbat[2]
[1,2]Department of Computer Science, University Mustapha Stambouli of Mascara, Algeria
E-mails: hichaam.deghbouch@gmail.com, debbat_fati@yahoo.fr

Keywords: Bees Algorithm, Coverage, Connectivity, Metaheuristics, CSA, Deployment, Optimization, Simulation

Received: November 21, 2020

In this paper, the problem of deployment in wireless sensor networks is investigated. The authors propose a Hybrid Modified Crow Search Bee Algorithm (HMCSBA) for coverage maximization with the guarantee of connectivity between the deployed sensors. Firstly, a Modified Crow Search Algorithm (MCSA) is proposed based on the basic CSA algorithm to form a connected network after initial random deployment. The position equation of the original CSA was updated by introducing a linear flight length that increases throughout iterations to force the sensors to join the network. Then, the Bees Algorithm (BA) is applied to optimize the network coverage without losing connectivity between the deployed sensors. Simulations and comparative studies were carried out to prove the relevance of the proposed algorithm. Results demonstrate that the proposed algorithm can optimize the coverage and guarantee network connectivity.

Povzetek: Razvit je bil hibridni algoritem HMCSBA za zagotovljeno povezljivost brezžičnih senzorjev v omrežju.

1 Introduction

The Internet of Things (IoT) can be defined as the network that connects all types of physical objects such as cell phones, home automation systems, sensors and enables these objects to collect and exchange the gathered data [1]. Wireless Sensor Networks (WSNs) as IoT data gathering systems [2] are a widely used technology that has the capability to provide reliable monitoring for many real-world applications such as traffic surveillance, security monitoring, health care, machine failure diagnoses, and environmental monitoring [3]. WSN consists of several small devices with limited resources called sensors. These sensors are distributed in a geographical region and organized into a cooperative network [4]. Sensors keep track of the events that take place in their surroundings and communicate with each other to report the gathered data to the sink for processing [5].

Despite the numerous advantages of wireless sensor networks that make them the dominant in remote monitoring, several issues that affect the detection capability must be addressed while implementing them. Among these issues, coverage and connectivity are the most critical problems to be considered during the deployment of the network [6].

Deployment schemes are categorized into two types: deterministic deployment and random deployment. In deterministic deployment, the sensors are precisely placed in exact locations to achieve the required coverage and connectivity [7]. This type of deployment becomes an infeasible option in some applications where the region of interest is inaccessible [8]. In such applications, sensors are randomly deployed in the target field, which results in areas of varying density [9]. Therefore, maximizing network coverage becomes an essential requirement. Besides, the random deployment results in disjoint groups of sensors that prevent them from transferring the data to the sink. Therefore, a proper deployment scheme is essential to achieve the required sensing coverage and ensure that the collected information is successfully transferred to the sink directly or through a multi-hop path.

The optimal deployment of WSNs was defined as an NP-hard optimization problem in most works in the literature [10, 11]. Exact techniques are not preferable when solving these types of problems where the computational time taken by the algorithms increases exponentially with the problem dimension [12]. As an alternative, metaheuristic algorithms have been used for obtaining the optimal solutions of various real engineering design optimization problems [13]. Therefore, this paper proposes a Modified Crow Search Algorithm (MCSA) hybridized with the Bees Algorithm (BA) for optimizing the network coverage while ensuring the connectivity between the deployed sensors. The proposed algorithm namely Hybrid Modified Crow Search Bee Algorithm (HMCSBA), consists of two stages. In the first stage, a modified crow search algorithm for forming a connected network after initial random deployment is proposed. In the second stage, the bees algorithm is applied for maximizing the network coverage while maintaining network connectivity. Our contribution is summarized as follows:

- Proposal of a modified CSA algorithm to form a connected network after the initial random deployment of sensors.

- The modified CSA is hybridized with the BA to solve the deployment problem with the guarantee of connectivity.
- The performance of the proposed algorithm is evaluated in terms of coverage and connectivity with different deployment settings.
- The proposed hybrid algorithm is compared with other state-of-the-art algorithms to demonstrate its effectiveness in solving the deployment problem.

The rest of this paper is organized as follows: Section 2 presents related works on sensor deployment. Section 3 presents network assumptions, sensor detection model, and connectivity model. Section 4 describes the proposed HMCSBA deployment algorithm. Simulation results are presented in section 5. Finally, section 6 concludes the proposed work.

2 Related work

Metaheuristic optimization algorithms are at the top of the preferred algorithms in optimizing different aspects related to WSNs. Finding approximate solutions to the deployment problem using metaheuristics is considered a very active search field [14]. The literature provides a huge number of methods based on metaheuristics to optimize the deployment of WSNs. For instance, the coverage maximization with the least number of deployed sensors is investigated in [15]. The authors proposed an Improved Dynamic Deployment Technique for WSNs based on Genetic Algorithm called IDDT-GA. A variable-length encoding is used in IDDT-GA to represent the set of deployed sensors in each chromosome. The main goal of this improvement is to reduce the number of used sensors. Consequently, it reduces the overlapping area between the sensors in the network. In addition, the connectivity is ensured by employing a penalty to the objective function. The work in [16] proposes a hybrid algorithm for maximizing the coverage in WSNs. In this proposition, the particle swarm optimization algorithm is applied to the deployment problem and hybridized with Hooke–Jeeves method as a local search algorithm. The local search algorithm is used to overcome the stagnation problem of standard Particle Swarm Optimization (PSO). If the PSO fails to achieve any improvement in the coverage after a predefined number of cycles, the Hooke–Jeeves method is applied to the global best of the particles in order to improve its fitness. The work in [17] presents two flower pollination algorithms for heterogeneous WSNs deployment with the presence of obstacles. The first proposition is a single-objective improved flower pollination algorithm aims at maximizing the network coverage. The improved algorithm introduces a chaotic map with a nonlinear convergence factor to deal with the slow convergence of the Flower Pollination Algorithm (FPA). The second proposition is a multi-objective algorithm based on non-dominated sorting designed to tackle the problem of deployment in a forest environment. The algorithm aims at maximizing the coverage, minimizing radiation overflow rate, and minimizing the energy consumption of the sensor nodes while maintaining connectivity. The problem of coverage and connectivity of a set of target points is investigated in [18]. The authors highlighted the two shortcomings of the Artificial Bee Colony (ABC) algorithm when using the traditional roulette wheel selection mechanism: the entrapment in local optima and the fast convergence. To overcome the shortcomings, a modified ABC is proposed where the roulette wheel selection mechanism of the follower bees is replaced with the free search algorithm pheromone sensitivity model. The work in [19] proposes an improved whale group algorithm based on a probabilistic coverage model to solve the problem of deployment in WSNs. The linear weighted sum method is adopted to derive a single-objective function that optimizes the coverage and energy consumption of nodes. In [20], a dynamic deployment algorithm for optimizing area coverage in WSNs is suggested. The authors used the whale optimization algorithm to update the positions of sensors after initial random deployment. A fitness interval is used to evaluate the coverage of the sensors. If a sensor node achieves a coverage value in the predefined coverage interval, it is marked as an optimum sensor. Therefore, it will maintain its current location in the following iterations. In [21], three different methods for solving the problem of deployment with a minimum number of sensors are presented. The first method is based on the integer linear programming where a set of constraints are introduced to guarantee the connectivity between the nodes. The two other methods are heuristics based on a local search and a genetic algorithm. The local search heuristic deterministically deploys the set of sensors while maintaining the connectivity. In their second method, the steps of the classical GA are implemented with a new chromosome generation algorithm.

Table 1 shows a summary of numerical results provided by the various related work algorithms.

The HMCSBA has the following advantages over related works:

- Unlike most works in the literature, the HMCSBA algorithm has the advantage of maintaining full connectivity during the coverage maximization procedure and not only at the end of it.
- Using HMCSBA, the connectivity is ensured regardless of the communication range value.
- The HMCSBA utilizes the strength of two metaheuristics to maximize the coverage without losing connectivity.

Algorithm	Number of sensors	Sensing range	Coverage	Connectivity
[15]	50	5	98.28%	100%
[16]	50	5	96.64%	Not ensured
[17]	50	5-7	98.25%	Not ensured
[18]	50	20	97%	Not always ensured
[19]	50	5	95%	Not ensured
[20]	60	7	79%	Not ensured
[21]	50	5	96-98%	Not always ensured

Table 1: Numerical results provided by related work algorithms.

3 System model

3.1 Network assumptions

The initial network assumptions of the proposed model are:

- The network is composed of n homogeneous sensor nodes with the same characteristics.
- All sensor nodes are mobile and have the ability to change their positions during deployment.
- The sensing region is a two-dimensional $M \times N$ grid where the distance separating two adjacent points of the grid is equal to 1 unit.
- The sink node is placed in a predetermined position in the sensing region.
- The nodes can communicate with each other directly or via a multi-hop communication links.

3.2 Coverage model

The coverage model describes how well the sensor nodes can monitor the events in the sensing region [22]. In WSN, the unit disk model is generally used to define the sensing zone of a sensor [23]. In the present work, the binary detection model is chosen as the coverage model. The sensor is capable of detecting and reporting the information about any event that takes place in its sensing zone defined by the sensing range R_s. In order to monitor a given grid point in the sensing region, the distance between a sensor s_i located at (x_i, y_i) and the grid point p located at (x, y) should be less than or equal to the sensing range R_s. The binary detection model can be expressed as follows [24]:

$$P(s_i, x, y) = \begin{cases} 1, & if\ d(s_i, p) < R_s \\ 0, & otherwise \end{cases} \tag{1}$$

where $d(s_i, p)$ is the Euclidean distance between sensor s_i and grid point p.

Since the region of interest is composed of a finite number of grid points, the deployed sensor nodes have to ensure full coverage of all the grid points. Therefore, given a set S of sensors, the probability that a grid point p located at (x, y) is covered by the set of sensors S can be written as [24]:

$$PC(S, x, y) = 1 - \prod_{i=1}^{K}(1 - P(s_i, x, y)) \tag{2}$$

where K is the number of sensors in the set S.
The coverage is expressed by the total grid points covered by the set of sensors. Therefore, the coverage rate of the area is given by [24]:

$$CR = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N} PC(S, x, y)}{M \times N} \tag{3}$$

3.3 Network connectivity

The connectivity in WSNs is necessary to guarantee the transfer of the gathered data to the sink for processing [25]. Usually, the connectivity is defined as the ability of each sensor to find a direct or a multi-hop path to reach the sink [26]. In multi-hop communication, the sensors exchange data with their direct neighbors until it reaches the sink. In order for two sensors s_i and s_j to be connected, the Euclidean distance separating them must satisfy the following condition:

$$Com(s_i, s_j) = \begin{cases} 1, & if\ d(s_i, s_j) \leq R_c \\ 0, & otherwise \end{cases} \tag{4}$$

where R_c is the communication radius.

4 Hybrid modified crow search bee algorithm (HMCSBA) for WSNs deployment

4.1 Crow search algorithm (CSA)

The Crow Search Algorithm (CSA) is a metaheuristic optimization technique inspired from the intelligent behavior of crows in nature [27]. Crows are clever birds that observe other birds, steal their food, and move it into hidden locations away from other predators. The hidden locations are memorized by crows using sophisticated ways. The crow can recall the location of its hidden food even after several months from its last visit to that location [28].
The main inspiration of the CSA is the robbery between a flock of crows. Each crow hides the information about its

food location, and the other crows follow it and try to steal its food [29]. In the CSA, each artificial crow has a memory to store its best position found so far. In each iteration, the crows change their locations by flying in the search space using other crows' positions.
The CSA algorithm starts with a number of crows flying in a d- dimensional space. At each iteration, each crow updates its position using the following equation:

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 \times fl \times (m_j^t - x_i^t) & if\ r_2 \geq AP \\ random\ position & otherwise \end{cases} \quad (5)$$

where m_j^t is the food position of a randomly chosen crow j at $t-th$ iteration, x_i^t is the position of the crow i at $t-th$ iteration, AP is the awareness probability of crow j. r_1 and r_2 are random numbers in the range $[0\ 1]$.
If the crow lands in a position fitter than its memorized one, the crow is required to update its memory to this latest position. The position updating process of crows will continue until a stopping criterion is satisfied. The pseudo-code of the CSA algorithm is presented in Figure 1.

The CSA algorithm

Randomly initialize the position of a flock of *N* crows in the search space
Evaluate the position of the crows
Initialize the memory of each crow
While *t < maximum number of iterations* **do**
    For i = 1 : *N* (all N crows of the flock)
        Randomly choose one of the crows to follow (for example j)
        Define the awareness probability
        If $r_2 > AP$
            $x_i^{t+1} = x_i^t + r_1 \times fl \times (m_j^t - x_i^t)$
        Else
            $x_i^{t+1} = random\ position$
        End if
    End for
Check the feasibility of new positions
Evaluate the new position of crows
Update the memory of crows
End while

Figure 1: Pseudo-code of the CSA algorithm.

The first goal of the proposed hybrid algorithm is to form a connected network after the random deployment of sensor nodes in the sensing region. For this purpose, each sensor node including the sink is treated as a crow in the CSA algorithm. The movements of sensors using CSA are illustrated in Figure 2. The figure shows the possible positions of sensor s_i that follows sensor s_j with different values of the flight length (fl). Sensor s_i can go to every position on the dashed line.

Initially, the sink node as an individual crow is deployed in a predetermined position in the sensing region, and it is added to a list $ConnList$ used to store the connected nodes. We assume that the sink is static and never changes its position. Then, the crows (sensors) are scattered randomly in the search space and marked as non-connected nodes. At each iteration, only the non-connected nodes are allowed to change their positions. Consequently, the connected nodes will maintain their current positions during the next iterations. Each sensor s_i will choose a randomly connected sensor s_j from $ConnList$ to follow it. After the position updating by CSA, the new positions of sensors are evaluated. If a non-connected sensor node s_i becomes connected to other sensor node s_j (in the communication range of s_j), the sensor node s_i will be added to the list of connected nodes $ConnList$, and the two sensors are marked as neighbors. After joining the network, the sensor s_i will not change its current position in the following iterations. These steps are repeated until a complete connected network is obtained.

With the algorithm proposed so far, the sensor nodes are allowed to change their positions according to the steps of the CSA algorithm to form a connected network. However, due to the position updating used by the CSA algorithm, especially the random choice of positions. There is no guarantee that all the sensors will connect to the network eventually. Therefore, in the next section, a modified version of the CSA algorithm will be introduced to deal with this issue.

4.2 Modified crow search algorithm (MCSA)

In the CSA algorithm, two different elements are responsible for the movement of crows: the flight length (fl) and the awareness probability (AP).

Firstly, to see the effect of the flight length (fl) on the movement of sensors, simulation with different values of the flight length is performed. Figure 3 shows the impact of the flight length on the position updating mechanism of the CSA algorithm. It should be noted that the random variable r_1 in equation (5) is neglected in this simulation ($r_1 = 1$).

As can be seen from Figure 3, different positions located between the current position of sensor s_i and the connected sensor s_j can be achieved by adjusting the value of the flight length. It is observed that sensor s_i is approaching more and more from sensor s_j as the value of the flight length is increasing until the sensors overlap when the value of the flight length reaches 1. Therefore, in our proposal, the flight length (fl) will linearly increase from 0.1 to 1 throughout iterations. This improvement will force the sensors to join the network in the last iterations if they failed to connect in the early iterations. Thus, the connectivity is guaranteed regardless of the communication range value. The flight length (fl) is updated as follows:

$$fl = 0.1 - iter \times (\frac{0.1 - 1}{MaxIter}) \quad (6)$$

where $iter$ is the iteration counter, and $MaxIter$ is the maximum number of iterations.

The randomness is an important characteristic used by most metaheuristics to improve their exploration capabilities and avoiding the entrapment in local solutions [13].

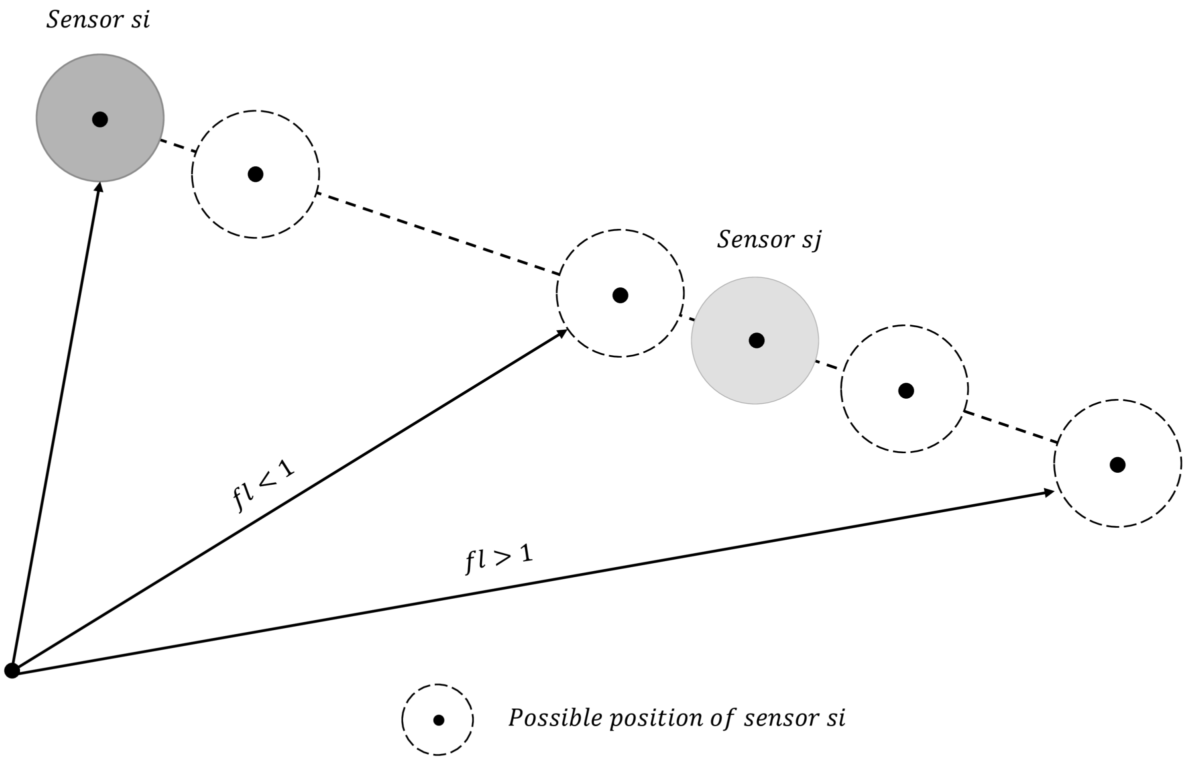


Figure 2: Sensor position update diagram using CSA.

The CSA algorithm also utilizes randomness when updating the positions of crows. The random movement in the CSA algorithm is controlled by the Awareness Probability (AP) variable [30], as shown in equation (5). This random choice of positions is not favorable in our situation because it has a great influence on the convergence of the algorithm. Besides, the use of random numbers such as r_1 could prevent the sensors from joining the network, especially in the last iterations, when our algorithm forces the isolated sensors to connect to the network. Therefore, in the modified version of the CSA algorithm, the AP and r_1 values are taken as 0 and 1, respectively. This modification guarantees that all the sensors will eventually connect to the network. Finally, the new position update equation of crows is written as:

$$x_i^{t+1} = x_i^t + fl \times (m_j^t - x_i^t) \tag{7}$$

where fl is linearly increasing from 0.1 to 1.

To sum up, the algorithm starts by deploying a set of sensors randomly in the deployment region. Only the sink node is considered to be connected. Therefore, it will be added to the list $ConnList$ that contains the set of connected nodes. After that, each non-connected node updates its position according to the position of a selected node from $ConnList$ using equation (7). When a sensor node joins the network, it will be added to the list $ConnList$ and never change its position during the next iterations. As the iteration counter increases, the size of the $ConnList$ will increase gradually as more nodes are joining the network. Consequently, the probability of a non-connected node to join the network will increase too. In addition, the sensors that failed to connect will be forced to join the network when the value of the flight length approaches or reaches 1 in the last iterations. These sensors will update their positions to the positions of the selected nodes, as shown in Figure 3. Therefore, the connectivity between the deployed sensors is guaranteed. The steps of the MCSA for network connectivity are summarized in Figure 4.

In addition to forming a connected network, a list of neighbors called $Neigh$ is returned by the MCSA algorithm for each deployed sensor node. The purpose of using the list $Neigh$ will be explained in the next section.

4.3 The bees algorithm for coverage maximization

4.3.1 The bees algorithm (BA)

The social interactions between the individuals of a colony of bees in nature are considered the main source of inspiration for many optimization algorithms. The Bees Algorithm (BA) proposed by Pham and Castellani [31] is one of the most powerful and successful swarm-based optimization algorithms. The BA takes its inspiration from the cooperative foraging behavior of a group of honeybees. In the bee colony, a small part of the population called scout bees is selected to search randomly the surroundings for promis-

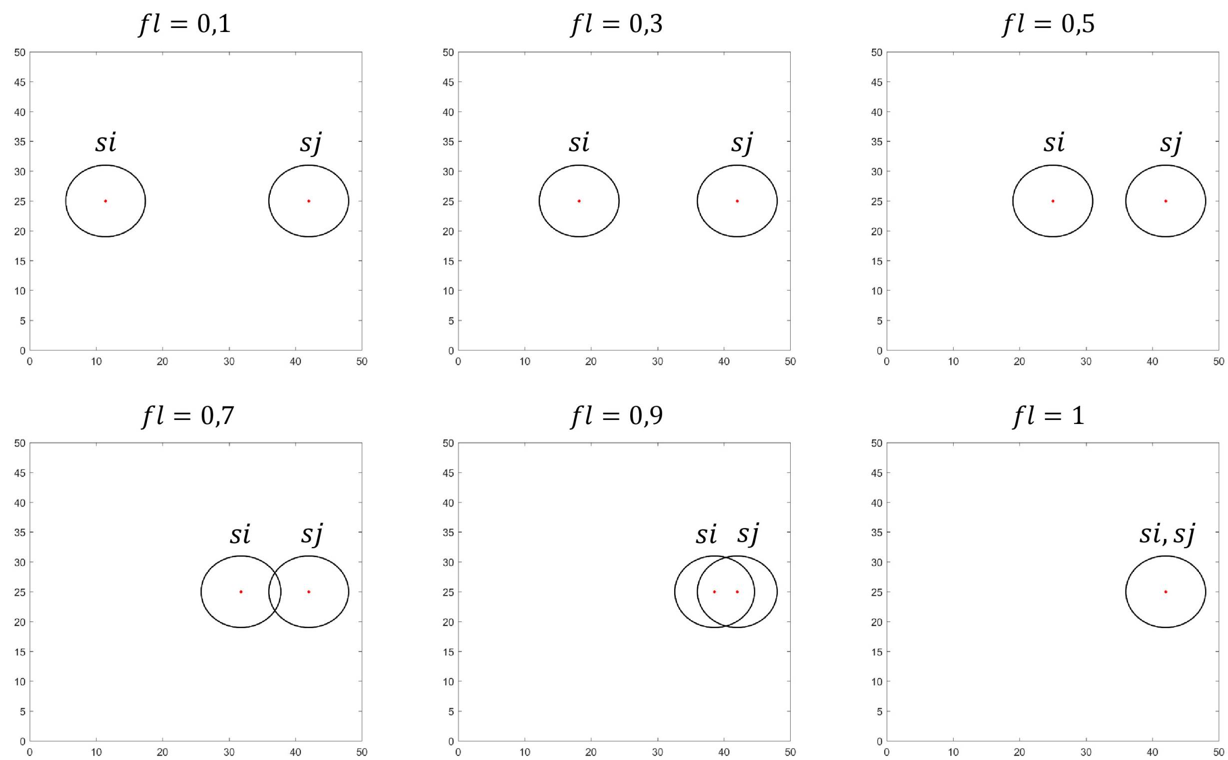


Figure 3: The behavior of the sensor s_i when varying fl.

ing flower patches to collect nectar. Once a scout bee discover a rich food source, it will memorize the information about the discovered food source and head back to the hive.

When the scout returns to the hive, it will deposit the collected nectar and heads towards a special place in the hive where other bees are waiting [32]. The scout bee starts moving by performing a dance routine to inform the other bees about the location and the quality of the discovered food source. After the scout finishes the special dance routine that is known as the waggle dance, it will recruit a number of foragers and goes back to the discovered food source with its follower bees to collect the nectar. When a forager gets back to the hive, it may waggle dance to recruit other bees to the selected food source [33].

4.3.2 Objective function derivation

With the algorithm proposed so far, the connectivity between the sensor nodes is guaranteed.

Consequently, all the gathered data is transferred and reported to the sink node with good quality. However, the sensor deployment scheme in WSNs aims at maximizing the coverage of the network by correctly positioning sensor nodes in appropriate locations [34]. The choice of optimal positions for the sensors will increase the probability of detecting most of the events that take place in the sensing region. In order to maximize the coverage while maintaining the connectivity, each sensor node is free to choose any position such that the distance separating him from its neighbors does not exceed the communication range. Usually, the neighbors of a sensor are all the sensors that lie in its communication range. However, having many neighbors will restrict the movement of the sensor preventing the algorithm from maximizing the coverage. Therefore, in the proposed approach, the neighbors of a sensor s_i are only the set of sensors in the list $Neigh(i)$ returned by the MCSA algorithm. The list contains the set of sensors that have connected to the network through sensor s_i combined with the sensor from which s_i has joined the network. Hence, each sensor node s_i will have at least one neighbor or a minimum set of neighbors to ensure at least 1-connectivity and gives him more flexibility in the movement to enhance the network coverage. Therefore, an objective function consisting of both coverage and the connectivity between the nodes can be written as:

- Maximize the coverage ratio while ensuring the connectivity between each sensor and its neighbors.

The objective function is formalized as follows:

$$Maximize\ (CR) \tag{8}$$

Subject to:

$$\sum_{s_j\ \in Neigh(i)} Com(s_i, s_j) = M_i, \forall\ s_i \in S \tag{9}$$

where $Neigh(i)$ denotes the set of neighbors of sensor s_i and M_i denotes the number of neighbors of sensor s_i.

The MCSA algorithm

Input:
The total number of sensor nodes Ns, the position of the sink node.
Output:
A connected network with new positions of sensor nodes.
A list of neighbors of each sensor node *si Neigh (i)*.
ConnList = ∅ // the set of connected sensors.
Initialize the position of the sink
ConnList = *ConnList* ∪ sink
Randomly initialize the positions of sensors in the search space
Initialize the memory of each crow
While *t* < *maximum number of iterations* **do**
Update flight length (*fl*) using equation (6)
For each non-connected sensor *si*
Randomly choose one of the crows to follow from *ConnList* (for example j)
$$x_i^{t+1} = x_i^t + fl \times \left(m_j^t - x_i^t\right)$$
If $Com(s_i, s_j) == 1$
ConnList = *ConnList* ∪ *si*.
Update the memory of crow i
Add *sj* to *Neigh (i)*.
Add *si* to *Neigh (j)*.
Else
If there is a sensor sj* such that sj* ∈ *ConnList* & $Com(s_i, s_{j*}) == 1$
ConnList = *ConnList* ∪ *si*.
Update the memory of crow i
Add *sj** to *Neigh (i)*.
Add *si* to *Neigh (j*)*.
End if
End if
End for
End while

Figure 4: Pseudo-code of the proposed MCSA algorithm.

4.3.3 Solution Representation

In BA, the position of each food source represents a feasible solution for the problem being optimized. Therefore, given $S = \{s_1, s_2, ..., s_i, s_n\}$ wireless sensor nodes to be deployed in the sensing region, each position of a food source X_i represents a deployment sequence expressed as an n-dimensional vector $X_i = (x_{s_1}, y_{s_1}, x_{s_2}, y_{s_2}, ..., x_{s_n}, y_{s_n})$ contains the position coordinates of the sensors as described in Figure 5. where x_{s_i} and y_{s_i} represent the x-coordinate and y-coordinate of the $i - th$ sensor node, respectively.

x_{s1}	y_{s1}	x_{s2}	y_{s2}	x_{s3}	y_{s3}	...	...	x_{sn}	y_{sn}

Figure 5: Solution Representation.

4.3.4 Initial population

At the beginning of the optimization task, a fixed number of scout bees is employed by the BA to search the surroundings for food sources. Each scout bee is assigned to one food source where the initial positions of the food sources are generated randomly in the search space. Since a solution is a vector that contains the positions of sensors, the random initialization will result in a disconnected network where disjoint segments of sensors are formed in the sensing region. Moreover, in some cases, many sensors could be isolated, which prevents them from transferring the gathered data to the sink. Therefore, random initialization is not feasible to guarantee connectivity between the nodes. To solve this issue, the random initialization of the solutions in the basic BA is replaced with the steps of the MCSA algorithm presented in this paper. The MCSA algorithm aims to connect the set of sensors and ensure data delivery between the sensors and the sink node. Thus, the initial solutions in the BA algorithm are guaranteed to be a set of connected networks.

4.3.5 The algorithm

In the artificial BA, the solutions are ranked based on the fitness values and categorized into two sets namely: selected sites and non-selected sites [35]. The selected sites are the solutions with the highest fitness, they are chosen for neighborhood search. Among the selected sites, the top-rated sites are called elite sites, they recruit the largest number of foragers nre. The remaining selected sites recruit nrb foragers ($nrb < nre$) for neighborhood search.
The neighborhood search is based on a random distribution of bees in a predefined range. Thus, an n-dimensional hyper-box neighborhood of size (ngh) is created around the selected sites. Then, at each iteration the scouts that located the solutions of highest fitness perform the waggle dance and recruit a number of foragers to conduct a local search in hope of finding a better solution. The solutions for the foragers are generated by changing the position of one of the sensors as follows:

$$v_{ij} = x_{ij} + r \tag{10}$$

where x_i is the selected solution, v_i is the new produced solution, r is a random number in the range $[-ngh, +ngh]$, and j is the randomly selected mobile sensor's position.

If one of the foragers finds a better solution than the scout, it becomes the new scout and participates in the waggle dance in the next generation. On the other hand, the scout bees that have found bad quality food sources (non-selected sites) are assigned in the solution space using MCSA algorithm scouting for new promising food sources.

To enhance the local search of the BA, two procedures named neighborhood shrinking and site abandonment are adopted in this work.

4.3.6 Neighborhood shrinking procedure

The initial size of a neighborhood (ngh) is kept unchanged as long as the local search in the neighborhood can find better solutions. If the local search failed in improving the fitness of a selected solution after a predefined number of iterations, the neighborhood size is decreased gradually according to the following formula [31]:

$$ngh(t+1) = 0.8 \times ngh(t) \tag{11}$$

This strategy aims at making the local search more exploitative by narrowing local search space and searching more densely the area around the optimum.

4.3.7 Site abandonment

If the shrinking procedure brings no improvement in the fitness value in a given neighborhood after a predefined number of iterations. It is assumed that the local optimum has been reached [36]. Therefore, the solution is replaced with a new one using the MCSA algorithm. If the abandoned site corresponds to the best global solution, it will be saved so that if the algorithm fails to achieve any enhancement, the stored solution is taken as the final one.

4.3.8 Bee population update

At the end of each iteration, the selected bee (best solution) from each patch combined with the remaining scout bees assigned randomly to conduct the global search will form the new population in the next iteration.
The pseudo-code of the HMCSBA algorithm is presented in Figure 6.

5 Performance evaluation

This section presents the obtained results from the simulations of the HMCSBA algorithm on the deployment problem in WSNs. In the first part of this section, the behavior

HMCSBA deployment algorithm

Input: size of area of interest, number of mobile sensors, number of scout bees, neighborhood size, stagnation *limit*, and maximum number of iterations *MaxIter*.

Output: an array contains the optimal positions of the mobile sensors.

- Initialize population by deploying *S* sensors for each food source Xi using **MCSA algorithm.**

t = 0;
While t ≤ *MaxIter*

1. Evaluate the fitness of the population.
2. Sort the solutions based on their fitness.
3. Select ***nb*** solutions with the highest fitness for neighborhood search.
4. Recruit ***nre*** foragers for each of the ***ne*** elite sites selected among the ***nb*** best sites.
5. Recruit ***nrb*** foragers for each of the remaining best sites.
6. Produce new solutions for the foragers in the neighborhood of the selected sites using equation (10).
7. Check the feasibility of produced solutions.
8. Select the fittest solution (bee) from each patch.
9. Perform the shrinking procedure using equation (11) when the neighborhood search of the retained solutions does not yield any progress.
10. Abandon sites that keeps falling into stagnation after a *limit* number of iterations.
11. Record the abandon solution if it corresponds to the best global solution.
12. Allocate the rest of the bees (non-selected sites) for global search using **MCSA algorithm.**
13. Form the new population.
14. Memorize the best solution achieved thus far.
15. t = t + 1.

End.

Figure 6: Pseudo-code of the proposed HMCSBA.

of the proposed HMCSBA in solving the coverage maximization problem with the guarantee of connectivity between the deployed sensors is analyzed with different sensing and communication ranges. The second part presents a set of comparative studies used to evaluate the performance of the HMCSBA algorithm in optimizing the network coverage compared with two recent and powerful deployment algorithms in the literature. For all the simulations, the deployed sensor nodes are considered homogeneous with the same capabilities. Furthermore, the sink is a special node placed in the center [37] of a $50m \times 50m$ terrain with the same communication capability as the other sensors.

Parameter	Value
Scout bees(n)	12
Best sites(nb)	10
Elite sites(ne)	4
Recruited bees of elite(nre)	5
Recruited bees of best(nrb)	1
Initial Neighbourhood size	6 ($ngh = R_s$)
Shrinking factor	0.8
Number of iterations	1000

Table 2: Parameter settings of BA algorithm.

5.1 Forming connected networks

As mentioned earlier, the first goal of the HMCSBA algorithm is to form a connected network after the random scattering of sensors in the deployment region. To observe this behavior, the algorithm is simulated with different number of sensors ($N = 10, N = 35$) in $50m \times 50m$ terrain with a sensing radius of $6m$. Two scenarios are simulated to demonstrate the effectiveness of the HMCSBA algorithm. In the first scenario, both the communication radius and the sensing radius are considered equal ($R_c = R_s$). In the second scenario, the communication radius is taken as twice the sensing radius ($R_c = 2 \times R_s$). The simulation results are shown in Figures 7 and 8.

Figures 7 and 8 show the initial random deployment of N sensors ($N = 10, N = 35$) with their corresponding results after applying our algorithm for both simulated scenarios. As can be seen from the figures, the random deployment creates disjoint groups of sensors scattered in the sensing region, with some nodes isolated completely from the network. Our algorithm was successful in forming strongly connected networks for all the different sensor densities. Based on the communication radius value, all the deployed sensors are organized to guarantee at least 1-connectivity for each sensor to transfer the collected data.

5.2 Coverage maximization with the guarantee of connectivity

The next goal of the proposed scheme is to maximize the network coverage by positioning the sensors in optimal locations with the consideration of maintaining the connectivity between the neighboring nodes. For observing the performance of the HMCSBA in optimizing the network coverage, the collected results from the simulations with the different number of sensors ($N = 10, N = 35$) are reported in Table 3. The parameter settings for the BA used in this simulation are illustrated in Table 2.

Figures 9 and 10 show the final deployment positions returned by the HMCSBA algorithm for the different sensor densities. For the first scenario, the proposed algorithm achieved a good sensor distribution of all the simulated number of sensors. The results of Table 3 shows that the HMCSBA algorithm performs well when $R_c = R_s$ by achieving coverage values of 29% and 79.16% for 10 and 35 sensors, respectively.

For the second scenario where the communication radius is twice the sensing radius, Figure 10 shows that the HMCSBA algorithm provides high-quality solutions by deploying the set of sensors in optimal locations to maximize the network coverage. The performance of the HMCSBA algorithm is confirmed by the high coverage values obtained, as shown in Table 3. Furthermore, the HMCSBA algorithm covers almost the entire area effectively for 35 sensors, where the coverage provided by the proposed algorithm reaches 99.24%.

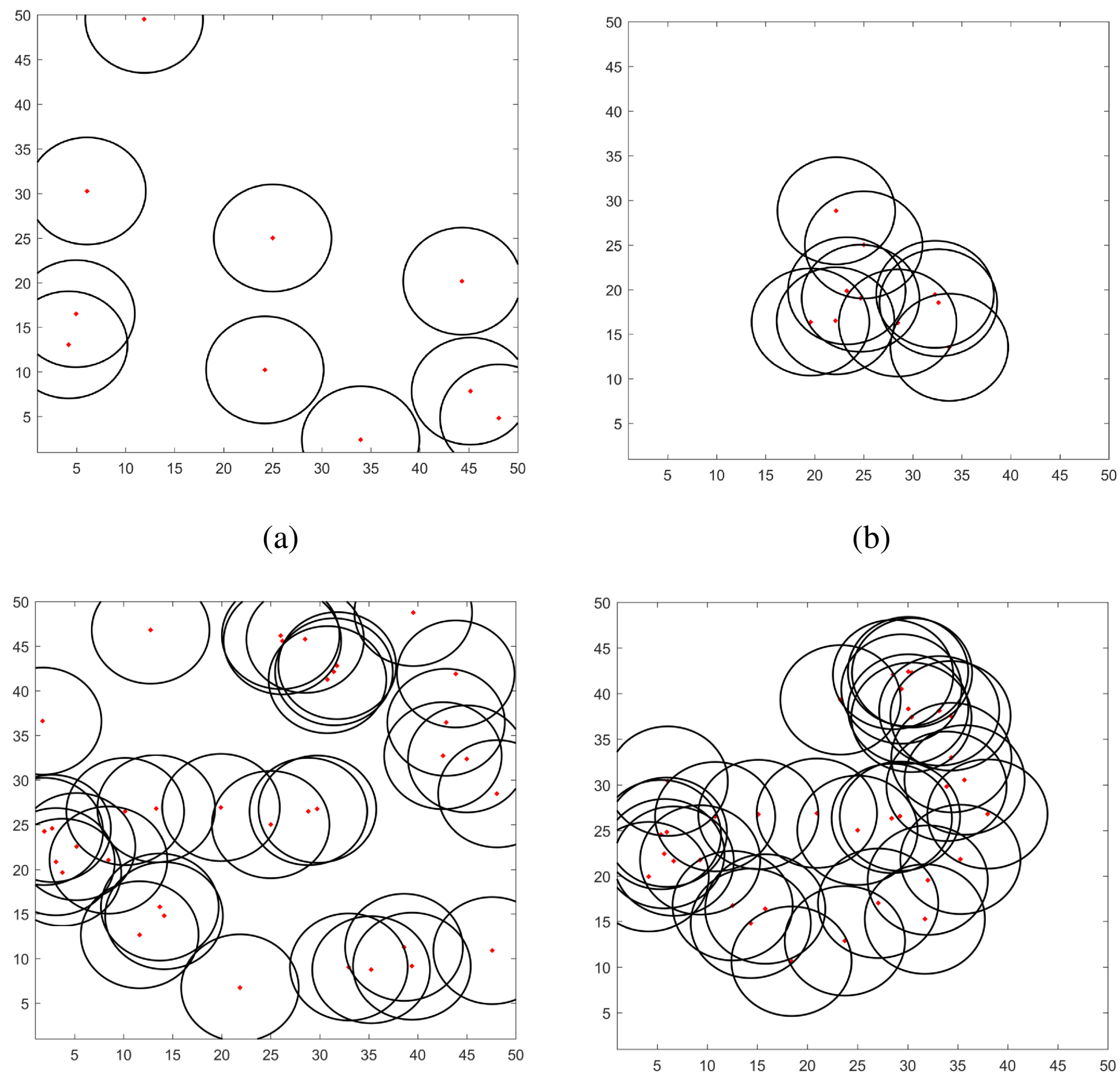
In addition, a full network connectivity is ensured by the HMCSBA algorithm while maximizing the coverage where each node has at least one neighbor directly connected to it. As can be seen in Table 3, the algorithm was successful in ensuring 100% network connectivity in the two simulated scenarios regardless of the number of deployed sensors.

In general, the HMCSBA ensures full connectivity while maximizing the coverage. Because at each step of the optimization task, the HMCSBA changes the position of one sensor rather than changing the positions of multiple sensors. This will give the HMCSBA full control over the movement of sensors, which allows it to choose the next candidate position of a sensor in a way that maximizes the coverage and maintains the connectivity with its neighbors. Furthermore, the shrinking of neighborhood size gradually reduces the movement of sensors. Consequently, this will reduce the probability of damaging the network connectivity during the movement of sensors.

5.3 Comparative study

This part investigates the effectiveness of the HMCSBA algorithm in solving the deployment problem in WSNs. For this purpose, a set of experiments are conducted using HMCSBA for different sensor densities. The obtained results are compared with the results of two recent deployment algorithms that are also based on metaheuristics namely WOA [20] and MCHSA [16]. The comparison is performed essentially on network coverage.

The performance of the HMCSBA in solving the deployment problem is assessed by comparing the coverage values and the standard deviation of the three algorithms. A total



(a) (b)

(c) (d)

Figure 7: (a) and (c) Initial deployment of 10 and 35 sensors, (b) and (d) Results after applying the MCSA algorithm when $R_c = R_s$.

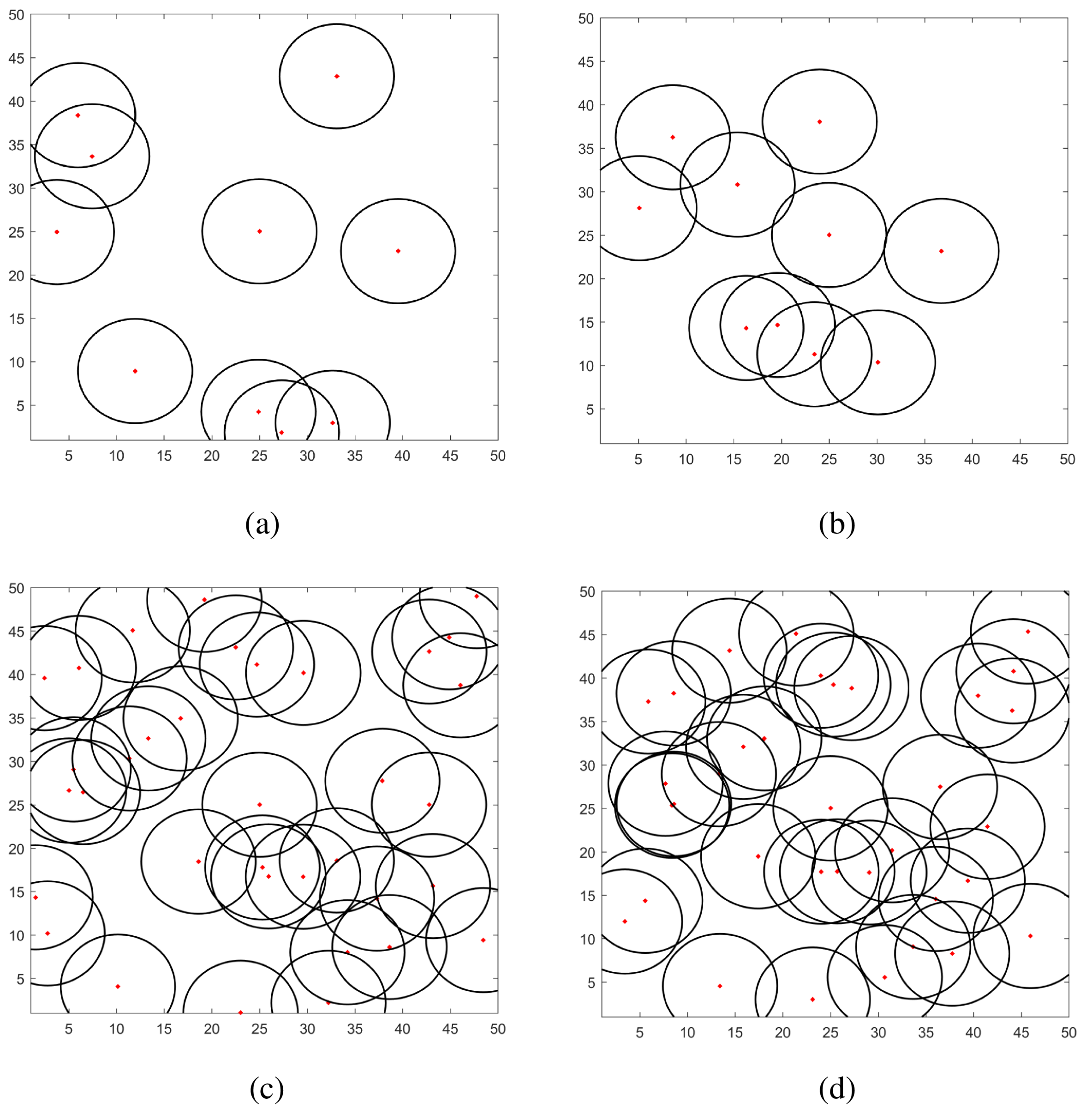
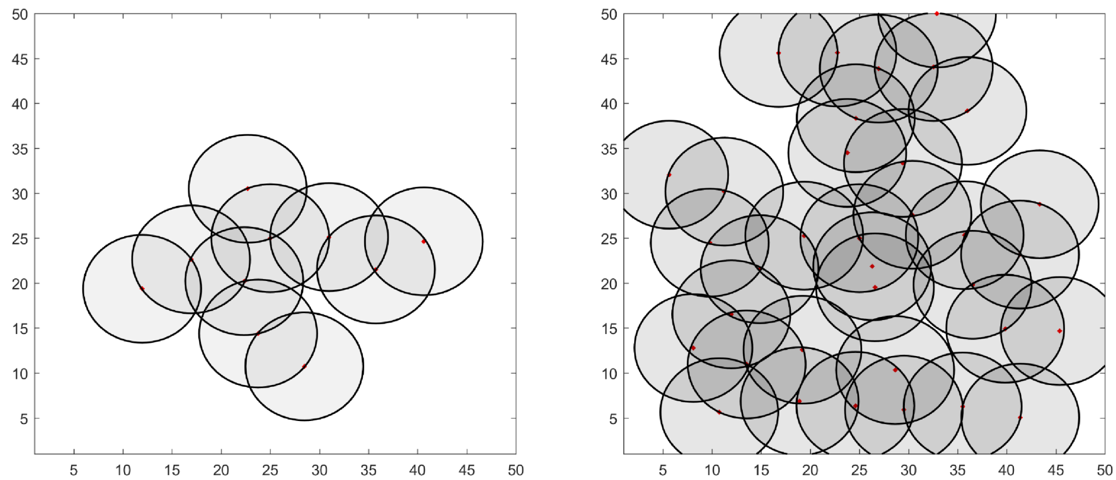
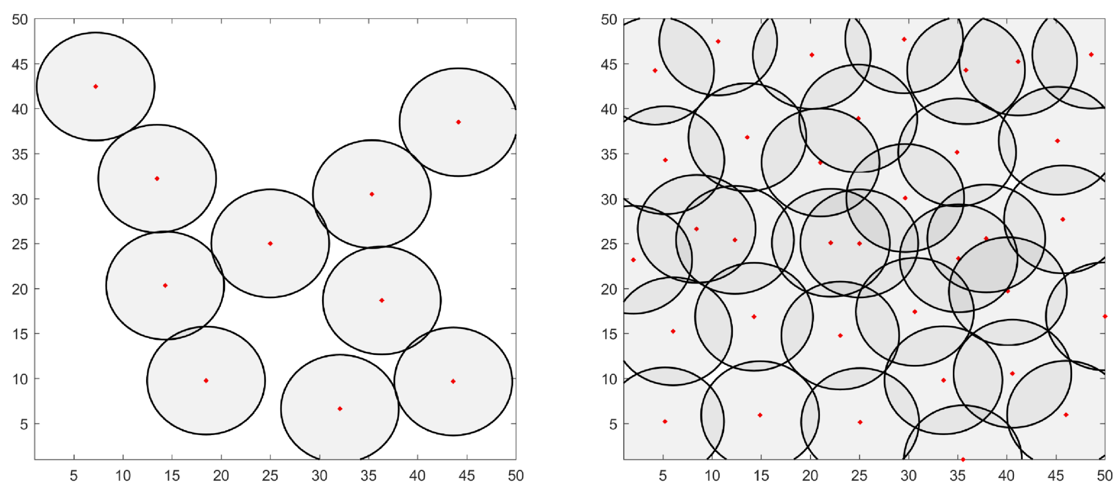


Figure 8: (a) and (c) Initial deployment of 10 and 35 sensors, (b) and (d) Results after applying the MCSA algorithm when $R_c = 2 \times R_s$.



(a) (b)

Figure 9: (a) final deployment of 10 sensors by HMCSBA, (b) final deployment of 35 sensors by HMCSBA when $R_c = R_s$.



(a) (b)

Figure 10: (a) final deployment of 10 sensors by HMCSBA, (b) final deployment of 35 sensors by HMCSBA when $R_c = 2 \times R_s$.

		HMCSBA	
Relation between R_c and R_s	Number of sensors	Coverage (%)	Connectivity (%)
$R_c = R_s$	10	29	100
	35	79.16	100
$R_c = 2 \times R_s$	10	45.84	100
	35	99.24	100

Table 3: Results of the HMCSBA algorithm.

of 15 simulations are performed by each algorithm. For this set of experiments, the number of sensors is varying between 10 and 50, the sensing radius is set to be $5m$, and the communication radius is taken as twice the sensing radius. The parameter settings of the HMCSBA used in this part are the same as the previous one provided in Table 2. The collected quantitative results are shown in Table 4.

According to the results of Table 4, the HMCSBA algorithm was successful in achieving the maximum network coverage in all the test cases. The HMCSBA algorithm provides significantly better results than that achieved by the WOA and MCHSA algorithms for 10 and 30 deployed sensors, where it covers 32.24% and 85.36% of the region of interest, respectively. Furthermore, the superiority of the proposed algorithm is observed when deploying 50 sensors where it reaches 98.84% of coverage higher than WOA and MCHSA by 4.36% and 2.2%, respectively, covering almost the entire region of interest.

Another key feature of the HMCSBA algorithm is its stability, where it provides small standard deviation values compared with the other competitors in two of three test cases except when the number of deployed nodes is 30, where the MCHSA provides the smallest value. This indicates that the HMCSBA algorithm is more effective than the two other algorithms in the majority of the experiments. The superior results can be observed on the mean values as well, proving that the HMCSBA algorithm outperforms WOA and MCHSA in all the employed metrics.

In addition, as can be seen in Figure 11, the overall coverage performance of the HMCSBA algorithm is verified as the number of deployed sensors increases. This figure indicates that the HMCSBA algorithm provides higher network coverage and outperforms the WOA and MCHSA algorithms regardless of the number of deployed sensors. Therefore, it can be said that the superiority of the HMCSBA algorithm is validated when solving the problem of coverage maximization.

Moreover, Table 5 shows that HMCSBA provides superior deployment results compared with those of related works presented in Table 1. To be more specific, the HMCSBA has ranked first, outperforming the other competitors where it provides the highest coverage with 100% network connectivity. Besides, as can be seen from Table 5, only one work has ensured full network connectivity between the sensors. Nevertheless, it is beaten by HMCSBA in terms of coverage with a difference of 0.56%.

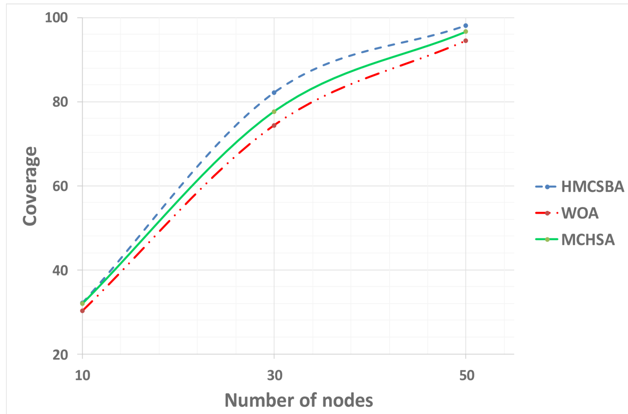


Figure 11: Comparison in terms of coverage rates.

6 Conclusion

This paper introduced a Hybrid Modified Crow Search Bee Algorithm (HMCSBA) for maximizing the network coverage while ensuring the connectivity between the deployed sensors. Firstly, a Modified Crow Search Algorithm (MCSA) is proposed where the flight length of crows is linearly increased to force the sensors to join the network in the last iterations. Besides, the random placement of crows is not considered to control the movement of sensors and guarantee connectivity. Secondly, the Bees Algorithm (BA) is applied to maximize the network coverage without losing the connectivity between the neighboring nodes. Finally, experimental and comparative studies with different sensor densities prove the superiority of the HMCSBA algorithm in forming a strongly connected network and optimizing the network coverage. In addition, the HMCSBA has the advantage of maintaining full connectivity during the coverage maximization procedure using different relations between the sensing radius and the communication radius. For future work, we will study the deployment of heterogeneous wireless sensor networks with a probabilistic detection model.

References

[1] Kumar, K. A., & Ramudu, K. (2019). Precision Agriculture using Internet of Things and Wireless sensor Networks. *Precision Agriculture* , 7(03).

	HMCSBA			WOA			MCHSA		
Number of nodes	Best(%)	Mean(%)	SD	Best(%)	Mean(%)	SD	Best(%)	Mean(%)	SD
10	32.24	32.08	0.1197	30.24	29.02	0.7795	31.96	31.73	0.2473
30	85.36	83.81	1.5509	74.36	72.99	0.9983	77.68	76.72	0.8706
50	98.84	98.56	0.2439	94.48	92.47	1.3824	96.64	94.83	1.2898

Table 4: Deployment results comparison between 3 algorithms.

Algorithm	Number of sensors	Sensing range	Coverage	Connectivity
[15]	50	5	98.28%	100%
[16]	50	5	96.64%	Not ensured
[17]	50	5-7	98.25%	Not ensured
[18]	50	20	97%	Not always ensured
[19]	50	5	95%	Not ensured
[20]	60	7	79%	Not ensured
[21]	50	5	96-98%	Not always ensured
HMCSBA	50	5	98.84%	100%

Table 5: Deployment results comparison between HMCSBA and related work algorithms.

[2] Harb, H., Idrees, A. K., Jaber, A., Makhoul, A., Zahwe, O., & Taam, M. A. (2017). Wireless sensor networks: A big data source in Internet of Things. *International Journal of Sensors Wireless Communications and Control* , 7(2), 93-109. https://doi.org/10.2174/2210327907666170906144926

[3] Aponte-Luis, J., Gómez-Galán, J. A., Gómez-Bravo, F., Sánchez-Raya, M., Alcina-Espigado, J., & Teixido-Rovira, P. M. (2018). An efficient wireless sensor network for industrial monitoring and control. *Sensors* , 18(1), 182. https://doi.org/10.3390/s18010182

[4] Guhan, R., Hari, U., & Ramachandran, B. (2019). Enhancement of QOS Parameters in Cluster-Based Wireless Sensor Network Using Cooperative MIMO. *Wireless Communication Networks and Internet of Things*, 187-195. https://doi.org/10.1007/978-981-10-8663-2_20

[5] Jain, E. V., & Kumar, E. T. (2017). A review of node deployment techniques in wireless sensor network. *International Research Journal of Engineering and Technology*, 4(8), 1697-1702.

[6] Chen, Y. N., Lin, W. H., & Chen, C. (2020). An effective sensor deployment scheme that ensures multilevel coverage of wireless sensor networks with uncertain properties. *Sensors*, 20(7), 1831. https://doi.org/10.3390/s20071831

[7] Cheng, W., Li, Y., Jiang, Y., & Yin, X. (2016). Regular deployment of wireless sensors to achieve connectivity and information coverage. *Sensors*, 16(8), 1270. https://doi.org/10.3390/s16081270

[8] Tripathi, A., Gupta, H. P., Dutta, T., Mishra, R., Shukla, K. K., & Jit, S. (2018). Coverage and connectivity in WSNs: A survey, research issues and challenges. *IEEE Access*, 6, 26971-26992. https://doi.org/10.1109/access.2018.2833632

[9] Farsi, M., Elhosseini, M. A., Badawy, M., Ali, H. A., & Eldin, H. Z. (2019). Deployment techniques in wireless sensor networks, coverage and connectivity: a survey. *IEEE Access*, 7, 28940-28954. https://doi.org/10.1109/access.2019.2902072

[10] Cheng, X., Narahari, B., Simha, R., Cheng, M. X., & Liu, D. (2003). Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics. *IEEE Transactions on mobile computing*, 2(3), 248-256. https://doi.org/10.1109/tmc.2003.1233530

[11] Chang, J. H., & Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on networking*, 12(4), 609-619. https://doi.org/10.1109/TNET.2004.833122

[12] Lanza-Gutiérrez, J. M., Caballé, N., Gómez-Pulido, J. A., Crawford, B., & Soto, R. (2019). Toward a Robust Multi-Objective Metaheuristic for Solving the Relay Node Placement Problem in Wireless Sensor Networks. *Sensors*, 19(3), 677. https://doi.org/10.3390/s19030677

[13] Mirjalili, S. (2015). The ant lion optimizer. *Advances in engineering software*, 83, 80-98. https://doi.org/10.1016/j.advengsoft.2015.01.010

[14] Tsai, C. W., Tsai, P. W., Pan, J. S., & Chao, H. C. (2015). Metaheuristics for the deployment problem of

WSN: A review. *Microprocessors and Microsystems*, 39(8), 1305-1317. https://doi.org/10.1016/j.micpro.2015.07.003

[15] ZainEldin, H., Badawy, M., Elhosseini, M., Arafat, H., & Abraham, A. (2020). 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*, 1-18. https://doi.org/10.1007/s12652-020-01698-5

[16] Panag, T. S., & Dhillon, J. S. (2019). Maximal coverage hybrid search algorithm for deployment in wireless sensor networks. *Wireless Networks*, 25(2), 637-652. https://doi.org/10.1007/s11276-017-1581-3

[17] Wang, Z., Xie, H., He, D., & Chan, S. (2019). Wireless Sensor Network Deployment Optimization Based on Two Flower Pollination Algorithms. *IEEE Access*, 7, 180590-180608. https://doi.org/10.1109/access.2019.2959949

[18] Yue, Y., Cao, L., & Luo, Z. (2019). Hybrid Artificial Bee Colony Algorithm for Improving the Coverage and Connectivity of Wireless Sensor Networks. *Wireless Personal Communications*, 108(3), 1719-1732. https://doi.org/10.1007/s11277-019-06492-x

[19] Wang, L., Wu, W., Qi, J., & Jia, Z. (2018). Wireless sensor network coverage optimization based on whale group algorithm. *Computer Science and Information Systems*, 15(3), 569-583. https://doi.org/10.2298/csis180103023w

[20] ÖZDAĞ, R., & CANAYAZ, M. (2017). A new dynamic deployment approach based on whale optimization algorithm in the optimization of coverage rates of wireless sensor networks. *European Journal of Technic*, 7(2). https://doi.org/10.23884/ejt.2017.7.2.06

[21] Rebai, M., Snoussi, H., Hnaien, F., & Khoukhi, L. (2015). Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. *Computers & Operations Research*, 59, 11-21. https://doi.org/10.1016/j.cor.2014.11.002

[22] Deng, X., Jiang, Y., Yang, L. T., Lin, M., Yi, L., & Wang, M. (2019). Data fusion based coverage optimization in heterogeneous sensor networks: A survey. *Information Fusion*, 52, 90-105. https://doi.org/10.1016/j.inffus.2018.11.020

[23] Zhou, J., Zhang, Z., Tang, S., Huang, X., Mo, Y., & Du, D. Z. (2017). Fault-tolerant virtual backbone in heterogeneous wireless sensor network. *IEEE/Acm Transactions on Networking*, 25(6), 3487-3499. https://doi.org/10.1109/infocom.2015.7218481

[24] Abo-Zahhad, M., Sabor, N., Sasaki, S., & Ahmed, S. M. (2016). A centralized immune-Voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks. *Information Fusion*, 30, 36-51. https://doi.org/10.1016/j.inffus.2015.11.005

[25] Khoufi, I., Minet, P., Laouiti, A., & Mahfoudh, S. (2017). Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges. *International Journal of Autonomous and Adaptive Communications Systems*. 10(4), 341 – 390. https://doi.org/10.1504/ijaacs.2017.10009671

[26] Boukerche, A., & Sun, P. (2018). Connectivity and coverage based protocols for wireless sensor networks. *Ad Hoc Networks*, 80, 54-69. https://doi.org/10.1016/j.adhoc.2018.07.003

[27] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*, 169, 1-12. https://doi.org/10.1016/j.compstruc.2016.03.001

[28] Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2019). Feature selection via a novel chaotic crow search algorithm. *Neural computing and applications*, 31(1), 171-188. https://doi.org/10.1007/s00521-017-2988-6

[29] Han, X., Xu, Q., Yue, L., Dong, Y., Xie, G., & Xu, X. (2020). An Improved Crow Search Algorithm Based on Spiral Search Mechanism for Solving Numerical and Engineering Optimization Problems. *IEEE Access*. https://doi.org/10.1109/access.2020.2980300

[30] Díaz, P., Pérez-Cisneros, M., Cuevas, E., Avalos, O., Gálvez, J., Hinojosa, S., & Zaldivar, D. (2018). An improved crow search algorithm applied to energy problems. *Energies*, 11(3), 571. https://doi.org/10.3390/en11030571

[31] Pham, D. T., & Castellani, M. (2009). The bees' algorithm: modelling foraging behaviour to solve continuous optimization problems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(12), 2919-2938.

[32] Castellani, M., Pham, Q. T., & Pham, D. T. (2012). Dynamic optimisation by a modified bees algorithm. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 226(7), 956-971.

[33] Pham, D. T., & Castellani, M. (2015). A comparative study of the Bees Algorithm as a tool for function optimisation. *Cogent Engineering*, 2(1), 1091540. https://doi.org/10.1080/23311916.2015.1091540

[34] Al-Karaki, J. N., & Gawanmeh, A. (2017). The optimal deployment, coverage, and connectivity problems in wireless sensor networks: revisited. *IEEE Access*, 5, 18051-18065. https://doi.org/10.1109/access.2017.2740382

[35] Castellani, M., Otri, S., & Pham, D. T. (2019). Printed circuit board assembly time minimisation using a novel Bees Algorithm. *Computers & Industrial Engineering*, 133, 186-194. https://doi.org/10.1016/j.cie.2019.05.015

[36] Pham, D. T., Baronti, L., Zhang, B., & Castellani, M. (2018). Optimisation of Engineering Systems With the Bees Algorithm. *International Journal of Artificial Life Research (IJALR)* , 8(1), 1-15. https://doi.org/10.4018/IJALR.2018010101

[37] Akhlaq, M., Sheltami, T. R., & Shakshuki, E. M. (2014). C3: an energy-efficient protocol for coverage, connectivity and communication in WSNs. *Personal and Ubiquitous Computing*, 18(5), 1117-1133. https://doi.org/10.1007/s00779-013-0719-2