



A Hybrid Bees Algorithm with Grasshopper Optimization Algorithm for Optimal Deployment of Wireless Sensor Networks

Hicham Deghbouch^[1, A], Fatima Debbat^[1, B]

^[1]Department of Computer Science, University Mustapha Stambouli of Mascara, Algeria

^[A]hichaam.deghbouch@gmail.com, ^[B]debbat_fati@yahoo.fr

Abstract This work addresses the deployment problem in Wireless Sensor Networks (WSNs) by hybridizing two metaheuristics, namely the Bees Algorithm (BA) and the Grasshopper Optimization Algorithm (GOA). The BA is an optimization algorithm that demonstrated promising results in solving many engineering problems. However, the local search process of BA lacks efficient exploitation due to the random assignment of search agents inside the neighborhoods, which weakens the algorithm's accuracy and results in slow convergence especially when solving higher dimension problems. To alleviate this shortcoming, this paper proposes a hybrid algorithm that utilizes the strength of the GOA to enhance the exploitation phase of the BA. To prove the effectiveness of the proposed algorithm, it is applied for WSNs deployment optimization with various deployment settings. Results demonstrate that the proposed hybrid algorithm can optimize the deployment of WSN and outperforms the state-of-the-art algorithms in terms of coverage, overlapping area, average moving distance, and energy consumption.

Keywords: Deployment optimization, Metaheuristics, Coverage, Energy consumption, Bio-inspired computing, Overlapping area, Swarm intelligence.

1 Introduction

Deployment optimization is a crucial issue that must be taken into consideration while designing an efficient Wireless Sensor Network (WSN) [1]. Usually, a WSN comprises a number of sensors that are dispersed in a geographical area [2]. Depends on the application, WSNs are deployed to achieve one or more objectives. In most real-life applications, the main purpose of any WSN is data collection [3]. To achieve this purpose, the sensors are required to sense all the events and report the maximum amount of information about them to the end-user [4]. In other words, the positions of sensors must be chosen carefully to maximize the area coverage. This makes the choice of the appropriate deployment strategy very critical.

The deployment strategy is determined according to the application and the nature of the environment [5]. When deploying the WSN in a friendly environment, the sensors are placed at accurate coordinates relying on a predefined deterministic pattern [6]. However, in unreachable or risky environments, the only available choice is random deployment, in which the sensors are scattered from the air over the sensing area [7]. Clearly, when the sensors are deployed at random, they will not ensure the expected performance, because there is a high chance that they end up placed above each other where their coverage areas overlap [8]. This results in poor coverage preventing the WSN from achieving the desired goals. In

the case of random deployment, most applications utilize mobile sensors, which gives them the ability to improve the coverage by moving the sensors to new locations after the initial random scattering.

In the last few decades, a significant number of optimization algorithms are proposed to solve real-life problems [9]. Metaheuristics as optimization algorithms have proven their efficiency in solving several challenging problems [10]. However, many of them face difficulties during optimizing a number of problems [11]. For instance, when the problem dimension is high (as in the case of WSN deployment) they fail in producing high-quality solutions, which prevents them from achieving approximate solutions of the global optimum. The Bees Algorithm (BA) [12] is a metaheuristic algorithm based on the behavior of honeybees when seeking food, it has been successfully applied to project scheduling [13], multiple disc clutch problem [14], printed circuit board assembly minimization [15], etc. However, BA has some disadvantages, mainly, in the exploitation phase where the repetitive unguided random search performed by the foragers results in slow convergence and low precision. Therefore, aiming at improving the exploitation of the BA, this paper proposes a hybrid algorithm based on the Bees Algorithm (BA) and the Grasshopper Optimization Algorithm (GOA) named as BAGOA. The GOA is a modern optimization algorithm inspired from the swarming behavior of grasshoppers. The strength of GOA lies in its high level of exploitation guided by the social interactions between all the agents in the swarm [16], which makes it ideal for hybridization with the BA. In this paper, BAGOA is applied for optimizing the deployment of WSNs after an initial random scattering of sensors. Our contribution is summarized as follows:

- We propose a hybrid algorithm called BAGOA that utilizes the GOA to strengthen the exploitation of the BA.
- The BAGOA is applied to solve the problem of deployment optimization in WSNs.
- The performance of BAGOA is evaluated in terms of coverage, overlapping area, moving distance, and energy consumption with different deployment settings.
- The BAGOA is compared with four 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 a literature review on sensor deployment problem. Sections 3 and 4 present an overview on the BA and GOA algorithms. Section 5 describes the proposed BAGOA algorithm. Section 6 presents the proposed deployment method. Simulation results are presented in section 7. Finally, section 8 concludes the proposed work.

2 Literature Review

Similar to various problems related to the design of WSNs, deployment optimization got its share of attention from the research community. In the literature, several studies have proposed plenty of solutions that differ according to the optimization objective, the method, and the application where the WSN is implemented. Regarding the method, metaheuristic algorithms are at the top of the solution list used for deployment optimization. The reason for using metaheuristics lies in their advantages over exact methods that require a lot of computational time to achieve a satisfactory solution.

An Improved Grey Wolf Optimizer (IGWO) for coverage optimization is presented in [17]. Firstly, a nonlinear convergence factor is introduced to balance exploration and exploitation. Secondly, the position updating equation of the GWO is improved by an enhanced weighting strategy. Finally, a dynamic variation strategy is used to maintain the diversity of the population and avoids entrapment in local solutions. The presented results show that IGWO delivers high-quality solutions in optimizing the coverage. However, despite the improvements introduced to the GWO, IGWO still suffers from slow evolution especially when the problem dimension is high. In [18], a dynamic deployment method based on the whale optimization metaheuristic for optimizing area coverage in WSNs is proposed. In this method, a coverage interval is defined for fitness evaluation. That is, 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. Although this mechanism makes the algorithm run fast and requires less computational time to finish the optimization task, it leads to a premature convergence, which prevents the algorithm from optimizing the coverage. The quick Artificial Bee Colony algorithm

(qABC), which is a variant of the standard ABC algorithm was introduced in [19] to solve the problem of deployment in WSNs. The authors considered both the Boolean and the probabilistic perception models. The deployment optimization is taken as a minimization problem where the aim is to minimize the uncovered area, thereby maximizing the coverage of the WSN. The qABC proved that it is effective in optimizing the coverage, however, it needs more CPU to perform optimization. The work in [20] proposes a Maximal Coverage Hybrid Search Algorithm (MCHSA) for maximizing area coverage using multiple types of sensors. The MCHSA is constructed based on the framework of the Particle Swarm Optimization (PSO) algorithm and the Hooke Jeeves method. The latter is applied to overcome the stagnation problem of the PSO algorithm. That is, if the PSO search stagnates, the Hooke Jeeves method is called to improve the fitness of the global best. MCHSA has the advantage of maximizing the coverage using multiple types of sensors. However, it takes a longer computational time because of the repetitive calling of the Hooke Jeeves method. In [21], a Biogeography Based Optimization (BBO) algorithm is applied to the dynamic deployment of WSNs including both mobile and static sensors. The algorithm starts by deploying a number of stationary and mobile sensors randomly in the sensing area. Then, the migration and mutation operators of the BBO are used to move the mobile sensors to optimal locations that optimize the network coverage. Experiments show that the BBO outputs good coverage results. However, it does not consider the energy consumption of sensors during movement. In [22], Social Spider Optimization (SSO) algorithm is implemented to solve the problem of deployment in WSNs. The proposed deployment scheme takes advantage of the information exchange model using vibration utilized by spiders to update the positions of sensors. According to the gender, each spider updates its position either by attraction or by repulsion to optimize the network coverage. Although the SSO provides good results in coverage maximization, it is not efficient when the number of sensors is high.

Aiming to enhance the search ability and the convergence of the Salp Swarm Algorithm (SSA), a Weighted SSA is applied in [23] to tackle the sensor deployment problem in WSNs. The location update equation of salps is improved by a new strategy called weighted distance position update. Experiments show that WSSA provides good coverage values and performs well in minimizing energy consumption. However, in their simulations, authors consider only a WSN with a small number of sensors in a deployment area of size 20m by 20m, which is not the case in real-world applications. The work in [24] proposes a multi-objective algorithm based on the firefly algorithm for area coverage optimization. The authors used the weight sum method to derive an objective function that maximizes coverage and minimizes the energy consumed during sensor mobility. Despite the good performance of the proposed algorithm in optimizing the energy, it is not efficient in maximizing coverage due to the lack of exploration phase. Authors in [25] implemented the moth-flame optimization algorithm for target coverage in WSNs. The proposed algorithm starts by deploying the sensors randomly, and then it utilizes the movement of moths to cover a set of targets fixed in prior. This strategy improves the coverage of targets, but it does not consider energy consumption during displacement.

3 Overview of the Bees Algorithm

The Bees Algorithm (BA) was proposed by Pham and Castellani in 2006 [12]. It mimics the behavior of a swarm of honeybees when searching for food sources. In BA, the bees are classified into two categories; scout bees and foragers. The scouts are responsible for discovering new food sources to collect nectar. They start exploring randomly the surroundings of the hive in search for promising flower patches. When a scout finds a good food source, it remembers the position information of the discovered food source and returns to the hive. The scout shares the information about its findings with the foragers that are waiting in the hive. The scout utilizes a special dance called the waggle dance to inform the foragers about the discovered food source [26]. The highly-rated patches that contain rich and easily available nectar (with higher fitness) attract the largest number of foragers. Finally, the scout bee goes back to the flower patch followed by the recruited foragers to collect the nectar. When a forager bee comes back to the hive, it may in turn waggle dance to direct other bees to the flower patch [27].

The artificial BA starts with ns scout bees being placed randomly in the search space. Then, the position of each scout is evaluated using a fitness function. At each iteration, the nb scouts that discovered the solutions with highest fitness perform the waggle dance to recruit a number of foragers. The ne top-rated solutions recruit the largest number of foragers (nre) to the discovered flower patch. The remaining

scout bees recruit ($nrb < nre$) foragers. This mechanism allows a larger number of bees to exploit the most promising areas in the search space.

The foragers are placed randomly on solutions in the discovered flower patch. The flower patch represents the visited food source and its vicinity expressed by the neighborhood size (ngh). If within the neighborhood a forager lands on a solution better than the solution advertised by the scout, this forager becomes the new scout and replaces the old scout in performing the waggle dance in the next iteration [28]. The remaining scout bees ($ns - nb$) that have found poor food sources are assigned randomly in the solution space scouting for new promising flower patches. The above steps are repeated until a satisfactory solution is found or a maximum number of iterations is reached. The pseudo code of the BA is presented in Figure 1.

Pseudo-code of the BA

Randomly initialize the position of the scouts

While $t \leq \text{MaxIter}$

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. Randomly produce new solutions for the foragers in the neighborhood of the selected sites
7. Check the feasibility of produced solutions
8. Select the fittest solution (bee) from each patch
9. Allocate the rest of the scouts randomly scouting for new solutions
10. Form the new population
11. Memorize the best solution achieved thus far
12. increment t by 1

End

Figure 1: Pseudo code of the BA algorithm.

4 Framework of GOA

Grasshoppers are voracious insects that eat almost all types of plants that come in their path. These small insects are considered a nightmare for farmers due to the huge damage that they inflict on agricultural crops. Usually, the grasshoppers live and eat individually in nature but most of the time millions of these insects meet and form one of the largest swarm seen in nature. Similar to other insects, the grasshopper life cycle is divided into two stages: larval and adulthood. The swarming of grasshoppers is seen in both stages but with different behaviors. In the larval stage, the grasshoppers move slowly in the ground with very small steps because they have no wings. In contrast, adult grasshoppers form a swarm in the air where they move abruptly with larger steps [29, 30].

The Grasshopper Optimization Algorithm (GOA) is a modern nature-inspired algorithm presented by Saremi et al., 2017 [31] for solving optimization problems. The characteristics of the two swarming behaviors are the main motivation of the GOA. The movement of grasshoppers in a swarm is mathematically formulated as follows [31]:

$$X_i = S_i + G_i + A_i \quad (1)$$

Where X_i denotes the position of i -th grasshopper, S_i is the social interaction, G_i is the gravity force, and A_i represents the wind advection.

Among the three components in equation 1, the social interaction is the most important factor in the movement of grasshoppers, which can be expressed as follows:

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(d_{ij}) \widehat{d}_{ij} \quad (2)$$

$$d_{ij} = |x_j - x_i| \quad (3)$$

$$\widehat{d}_{ij} = (x_j - x_i)/d_{ij} \quad (4)$$

Where d_{ij} is the distance between the i -th and the j -th grasshopper, \widehat{d}_{ij} is a unit vector from the i -th to the j -th grasshopper, N is the number of grasshoppers, and finally the $s()$ function represents the strength of social forces:

$$s(r) = fe^{-\frac{d}{l}} - e^{-d} \quad (5)$$

Where f and l represent the intensity of attraction and the attractive length scale, respectively.

Based on the distance, the space between two grasshoppers is divided into three zones: repulsion zone, attraction zone, and comfort zone (where there is neither attraction nor repulsion). Each grasshopper updates its position either by attraction or repulsion taking into consideration the positions of all grasshoppers in a swarm.

The gravity force and wind advection of the i -th grasshopper are calculated according to the following equations:

$$G_i = -g\widehat{e}_g \quad (6)$$

$$A_i = u\widehat{e}_w \quad (7)$$

Where g and u are constants, \widehat{e}_g and \widehat{e}_w represent the unity vector towards the center of the earth and the direction of the wind, respectively.

The mathematical model proposed so far cannot be utilized directly to perform optimization because the swarm does not converge to a particular point, a modified version of equation 1 is proposed in [31] to solve optimization problems as follows:

$$X_i^d = c \left(\sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right) + T_d \quad (8)$$

Where ub_d and lb_d denote the upper and the lower bounds of the d -th dimension, T_d represents the value of the d -th dimension in the best solution. The outer c is a decreasing coefficient that balance exploration and exploitation around the best solution, the inner c is utilized to shrink the comfort zone, repulsion zone, and attraction zone with respect to the number of iterations. The parameter c is computed by the following equation:

$$c = c_{max} - l \frac{c_{max} - c_{min}}{L} \quad (9)$$

Where l is the current iteration and L is the maximum number of iterations. Generally, the values of c_{max} and c_{min} are taken as 1 and 0.00001, respectively.

In the modified equation, the gravity force has not been considered and the wind direction has been assumed towards the target T_d . The pseudo code of the GOA algorithm is illustrated in Figure 2.

5 Hybrid Bees Algorithm with Grasshopper Optimization Algorithm (BAGOA)

Randomness is an important characteristic employed by metaheuristics mainly in the exploration phase where the search agents are subject to abrupt movements to explore different areas of the search space. In addition, it is also utilized to avoid entrapment in local solutions during optimization. However, in the exploitation phase, the search should be directed and concentrated near the best solutions to improve the solution quality and increase the chances of obtaining a good approximation of the global optimum [32].

The foremost matter of concern is that the BA relies on randomness in all the phases of optimization. When exploring the search space, the scouts that have found low-quality solutions are distributed uniformly at random in the search space scouting for new promising solutions. Similar to exploration, in the exploitation phase, the foragers in the BA are placed on solutions that are also generated randomly in the neighborhood of the selected solutions. Essentially, both exploration and exploitation are mainly

Pseudo-code of GOA

Randomly initialize the position of the swarm of grasshoppers
Initialize c_{max} , c_{min} and maximum number of iterations
Evaluate the positions of grasshoppers
T = the best grasshopper
While (t < Maximum number of iterations) **do**
 Update c using **Equation (9)**
 For each search agent
 Normalize the distances between grasshoppers in [1, 4]
 Update the position of the current search agent by **Equation (8)**
 Bring the current search agent back if it goes outside the
 boundaries
 End for
 Update T if there is a better solution
 increment t by 1
End while

Figure 2: Pseudo code of the GOA algorithm.

performed using random mutations in current solutions. This random behavior is not efficient, especially during exploitation due to its influence on the convergence speed of the algorithm. The random and undirected search leads to slow convergence because the search agents were not allowed to exploit the neighborhood efficiently. Moreover, the repetitive random local search performed in the neighborhoods relatively weakens the algorithm's accuracy, stability, and success rate.

To overcome the aforementioned shortcomings, the BA is hybridized with GOA to enhance the exploitative strength of the algorithm. The purpose of proposing the hybrid algorithm is to minimize the effect of randomness in BA during exploitation to improve the algorithm's search capability and makes the search oriented toward the best solution in the neighborhood. As aforementioned, the GOA utilizes the information of all search agents to define the next position of each one of them. Therefore, instead of randomly searching around the solution advertised by the scout, the BAGOA updates the position of each forager based on its current position, the position of the target (best solution in the neighborhood), and the position of all other foragers in the corresponding neighborhood. Initially, the foragers are distributed at random in the neighborhoods, then the BAGOA selects the best experienced forager bee in each neighborhood as the target and adjusts the positions of the other foragers based on the social knowledge according to equation (8). This mechanism allows the foragers to exploit the neighborhood efficiently. Moreover, the directed search toward the best solution in each neighborhood increases the chances of achieving a good approximation of the global best, which in return leads toward superior results in terms of convergence and accuracy.

It should be noted that the parameter c in equation (8) reduces the movements of foragers around the best solution in the neighborhood, which is essential in exploitation. The foragers will converge towards the target as much as possible in the last steps of optimization seeking a more accurate target. The pseudo code of the proposed BAGOA algorithm is illustrated in Figure 3.

6 BAGOA for deployment optimization

6.1 Coverage Model Description

For coverage calculation, we utilize the Boolean perception model to define the detection capability of a sensor. In this model, the sensing area where the sensors are deployed is regarded as a grid of dimension $M \times N$. To optimize the coverage, the sensors are supposed to cover the maximum number of points in the grid. The coverage of each point in the grid is judged based on the Euclidean distance separating it

Pseudo-code of the BAGOA

Randomly initialize the position of the scouts
Initialize c_{max}, c_{min}
While $t \leq \text{MaxIter}$

1. Evaluate the fitness of the population
2. Sort the solutions based on their fitness
3. Update c using **Equation (9)**
4. Select **nb** solutions with the highest fitness for neighborhood search
5. Recruit **nre** foragers for each of the **ne** elite sites selected among the **nb** best sites
6. Recruit **nrnb** foragers for each of the remaining best sites
7. Randomly produce new solutions for the foragers in the neighborhood of the selected sites
8. Calculate the distances between foragers and normalize them in $[1, 4]$
9. Update the position of the foragers by **Equation (8)**
10. Check the feasibility of produced solutions
11. Select the fittest solution (bee) from each patch
12. Allocate the rest of the scouts randomly scouting for new solutions
13. Form the new population
14. Memorize the best solution achieved thus far
15. increment t by 1

End

Figure 3: Pseudo code of the proposed BAGOA algorithm.

from the set of sensors. For covering a grid point T located at (x_t, y_t) by a sensor s_i located at (x_i, y_i) in 2D-space, the distance separating them should be less than or equal to the perceptual radius of the sensor s_i as expressed in equation (10) [33].

$$P(x_i, y_i, x_t, y_t) = \begin{cases} 1, & \text{if } \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2} < R_s \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Where R_s is the perceptual radius of sensor s_i .

Based on equation (10), the coverage of the grid point T by all the sensors in the network can be determined according to the following equation [34]:

$$PC(S, x_t, y_t) = 1 - \prod_{i=1}^{sn} (1 - P(x_i, y_i, x_t, y_t)) \quad (11)$$

Where S is the set of sensors and sn is the total number of sensors.

At last, the coverage rate of all the grid points is computed using equation (12) [34].

$$CR = \frac{\sum_{x_t=1}^M \sum_{y_t=1}^N PC(S, x_t, y_t)}{M \times N} \quad (12)$$

6.2 Solution Representation

In BAGOA, the position of each food source represents a feasible solution for the problem being optimized. Assume we have $S = \{s_1, s_2, \dots, s_i, s_n\}$ wireless sensors to be deployed in the area of interest. Each solution $X_i (i = 1, \dots, m)$ is a vector $X_i = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)$ that contains the position coordinates of all the sensors. Where x_i and y_i represent the coordinates of the i -th sensor in 2D space.

6.3 Initialization

The BAGOA starts by employing a number of scout bees randomly in the search space looking for food sources. As aforementioned, the position of each food source can be identified as a vector of coordinates.

Therefore, given a deployment area of size $M \times N$, the initial coordinates are chosen randomly inside the bounds of the deployment area such that $0 \leq x \leq M$ and $0 \leq y \leq N$.

6.4 Deployment Optimization

The algorithm starts by randomly deploying S sensors for each solution (food source). Then, each scout bee is assigned to one solution, which means that the number of generated solutions is equal to the number of scouts. After that, the solutions are evaluated in terms of coverage to determine the best and the elite bees. In the next step, the selected scout bees recruit a number of foragers for neighborhood search. The recruited foragers are placed on random solutions in the vicinity of the solution advertised by the scout. At this stage, the best forager in each neighborhood is selected as a target and the positions of the foragers in the corresponding neighborhood are updated according to the position update equation of GOA (equation (8)). If one of the foragers finds a better network coverage than the scout, that forager becomes the new scout and participates in the waggle dance in the next generation. The global search of the BAGOA is the same as BA where the bees that ranked last are assigned randomly in the solution space scouting for new promising solutions. The above steps are repeated until a coverage value above a predefined threshold is met or a maximum number of iterations is reached. The flow chart of the BAGOA deployment algorithm is illustrated in Figure 4. The dotted rectangle shows the improvement introduced in this paper.

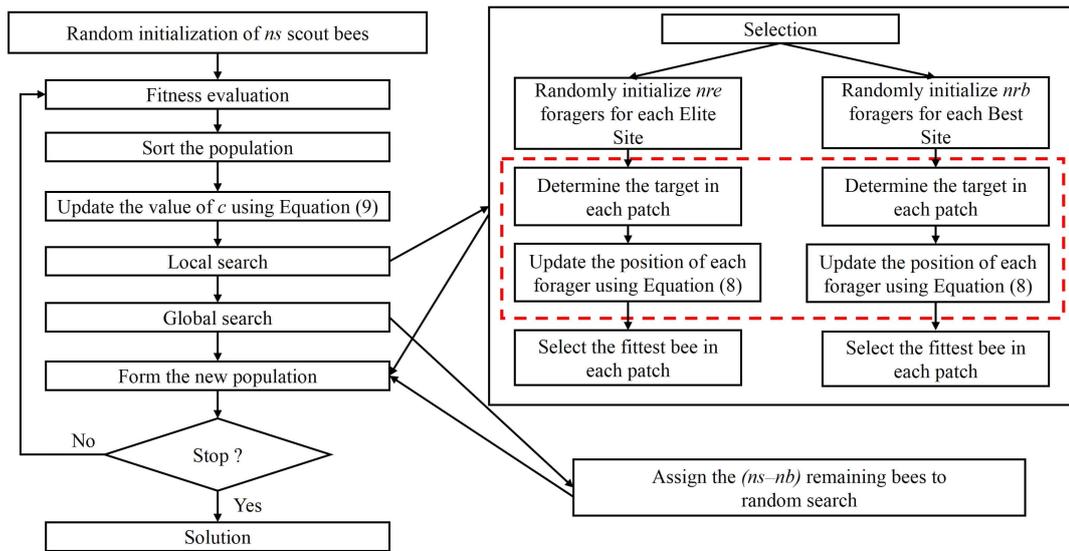


Figure 4: Flow chart of the proposed BAGOA algorithm.

7 Experimental Studies

This section is designed to prove the superiority and the effectiveness of the proposed hybrid algorithm in solving the problem of deployment in WSNs. Three groups of experiments are conducted with different area sizes and sensing range values. In the first experiment, the proposed algorithm is compared with two related work algorithms namely BA and IGWO in terms of search accuracy and convergence. In the second experiment, the deployment performance of the BAGOA in a large sensing area is assessed with a different number of mobile sensors. The obtained results are compared with those of the qABC algorithm. The last experiment simulates a wireless sensor network composed of both stationary and mobile sensors. The BBO is chosen as a comparison algorithm to evaluate the results of the hybrid algorithm presented in this paper.

7.1 Parameter Settings

Table 1 illustrates the information about the comparison algorithms used in the set of experiments. The parameter settings of the algorithms are the same as those in the corresponding publications. The size of the population of all the comparison algorithms is set to 30. Each algorithm is executed 10 times, and the best and average results are analyzed. In all the simulations, the initial positions of mobile sensors are generated randomly inside the sensing area. The parameter values of the BAGOA used for comparison with related works are presented in Table 2. It should be noted that the algorithms were implemented using Matlab R2017a software.

Table 1: The comparison algorithms.

Algorithm	Reference	Sensor type	Coverage type
BA	[35]	Mobile	Area coverage
IGWO	[17]	Mobile	Area coverage
qABC	[19]	Mobile	Area coverage
BBO	[21]	Static and Mobile	Area coverage

Table 2: Parameter settings of the BAGOA.

Parameter	Value
Scout bees (ns)	7
Best sites (nb)	5
Elite sites (ne)	3
Recruited bees of elite (nre)	7
Recruited bees of best (nrbs)	2
Neighbourhood size (ngh)	40

7.2 Evaluation Metrics

7.2.1 The overlapping area

Given the fact that the sensors are homogeneous, two sensors are said to be overlapped if the distance separating them is less than twice the sensing radius. The overlapping area between the two sensors can be expressed as the sum of common grid points covered by them [36]. Therefore, the overlapping area between the set of sensors can be computed as follows [37]:

$$Overlapp_{area} = \sum_{x=1}^M \sum_{y=1}^N overlapp(S, x, y) \quad (13)$$

Where

$$overlapp(S, x, y) = \begin{cases} 1, & \text{if the grid point } T(x, y) \text{ is covered by at least two sensors from } S \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

7.2.2 Energy consumption

Controlling energy consumption during deployment has a significant influence on the life span of the WSN [38]. After optimizing the coverage, the sensors are required to move from their initial random positions to their final positions optimized by the algorithms. Because sensors are constrained in terms of energy, the movements of sensors should be reduced as much as possible in order to conserve energy. In the present work, the total energy consumed by the sensing devices during movement can be computed as follows [39]:

$$E = \omega \times d_{all} \quad (15)$$

Where ω is the amount of energy depleted per meter of movement, d_{all} represents the average moving distance in the network, and it is expressed as follows [39]:

$$d_{all} = \frac{\sum_{i=1}^{sn} dist(initial_i, final_i)}{sn} \quad (16)$$

Where sn is the number of sensors, $dist$ is a function that calculates the Euclidean distance between the initial and final positions of sensors.

7.3 Experiments and results

In the first experiment, the BAGOA is simulated in an area of size $50\text{ m} \times 50\text{ m}$, the perceptual radius of the sensors is the same and fixed to 5 m , and the maximum number of iterations is 200. The collected quantitative results are compared with those obtained from the simulations of the standard Bees Algorithm (BA) and the Improved Grey Wolf Optimizer (IGWO). The comparison results are illustrated in Table 3. Please note that the values written in bold represent the best results.

Table 3: Deployment results comparison between the algorithms in an area of size 50 m by 50 m.

Number of sensors	BAGOA			BA			IGWO		
	Best(%)	Mean(%)	Std	Best(%)	Mean(%)	Std	Best(%)	Mean(%)	Std
30	87.56	86.45	0.680	84.56	83.90	0.385	86.52	84.91	0.861
40	97.52	96.74	0.420	94.88	94.30	0.426	95.84	94.29	1.089
50	99.72	99.39	0.190	99	98.38	0.343	98.88	97.94	0.756

Table 3 shows that BAGOA provides superior deployment results compared with the other algorithms in all the test cases. In particular, BAGOA was the most efficient algorithm in optimizing the coverage where it achieves 87.56% and 97.52% when deploying 30 and 40 nodes respectively. Furthermore, it covers almost the entire sensing area effectively where it reaches 99.72% of coverage with 50 deployed sensor nodes higher than those obtained using BA and the IGWO by 0.72% and 0.84%, respectively. Moreover, after BAGOA optimizes the deployment, the sensors are evenly distributed as illustrated in Figure 5 covering the sensing area effectively.

When comparing the coverage mean values of the three algorithms, Table 3 shows that the BAGOA provides significantly better results in all the test cases. For instance, in comparison with the mean values produced by IGWO, the values provided by BAGOA are improved up to 1.54%, 2.45%, and 1.45% when deploying 30, 40, and 50 sensors respectively. The standard deviation (Std) values in Table 3 also prove the superiority of the proposed algorithm where it achieves the smallest standard deviations in the majority of the test cases. This indicates that most of the time the BAGOA can optimize the network coverage and obtains a good approximation of the full coverage.

In further analysis, the convergence curves of the three algorithms are shown in Figure 6. It should be noted that the algorithms are executed with the same experimental parameters and the average of the solutions obtained in some iterations over 10 runs is compared for clarity. As may be seen in this figure, the BAGOA significantly outperforms the BA and the IGWO in optimizing the coverage where it reaches 99.39% on average. The IGWO is ranked last, where it provides a coverage value lower than that optimized using BA by 0.44%. In direct comparison with BA, the BAGOA tends to be accelerated faster as iteration increases. The reason why BAGOA converges fast is that the foragers search the neighborhood efficiently guided by the social interaction and oriented toward the best solution in the neighborhood instead of the repetitive blind search performed by the standard BA. In addition, we observe from the first iterations that the BAGOA coverage curve is always above the curves of the other competitors. This indicates that BAGOA has the fastest convergence speed, followed by the BA algorithm with a big difference from IGWO, which has the slowest speed. Therefore, Figure 6 verifies the validity of the improvement proposed in this paper in enhancing the convergence and the precision of the standard BA.

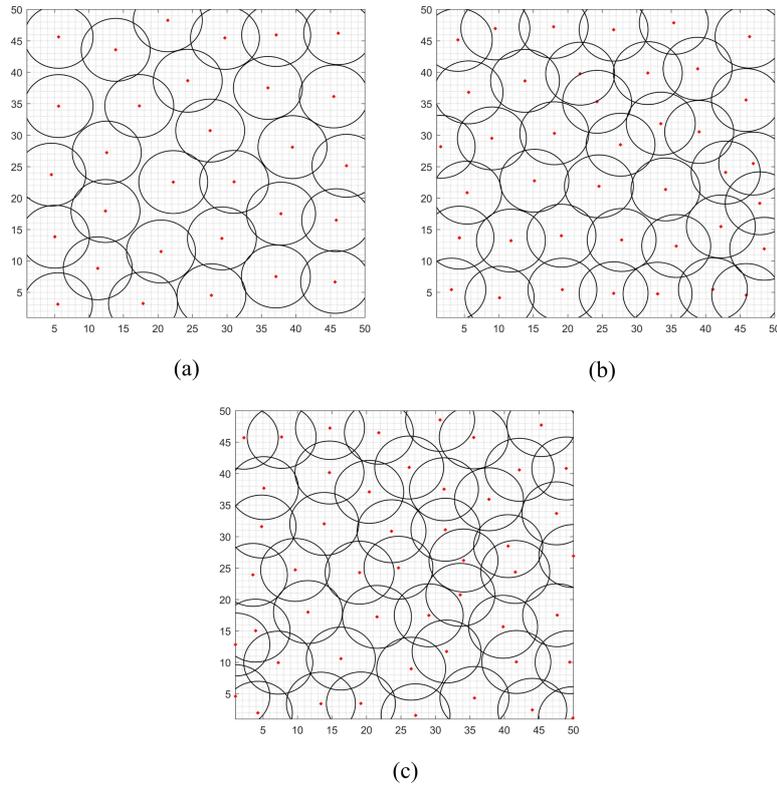


Figure 5: (a) final deployment solutions derived by BAGO: (a) for 30 sensors, (b) for 40 sensors, and (c) for 50 sensors.

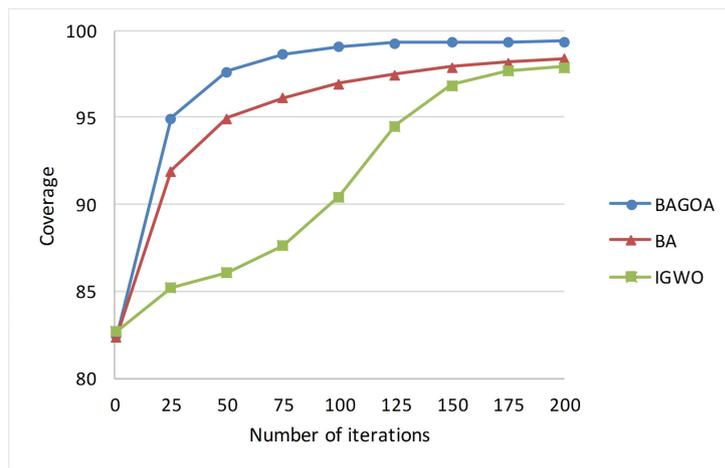


Figure 6: Convergence curves of BA, IGWO, and BAGO.

In addition to the above, the BAGO delivers significantly better results in terms of minimizing the average moving distance and energy consumption outperforming both BA and the IGWO. As can be seen from Figures 7 and 8, IGWO results in fast depletion of the energy of sensors where it fails to minimize the moving distance during deployment. On the other hand, both BAGO and BA achieved

better results than IGWO with varying numbers of nodes. However, BAGOA was able to minimize the moving distance of sensors where it increases the energy saving up to 3.20%, 3.72%, and 3.65% compared with BA when deploying 30, 40, and 50 sensors respectively. The main reason is that the movement of sensors is bounded by the neighborhood size. Unlike IGWO that move the sensors to any locations inside the entire sensing area, BAGOA avoids long distance movements by reducing the boundaries of the movement area. Therefore, each sensor will move only in the area marked by the neighborhood size. Furthermore, the shrinking of the comfort zone, repulsion zone, and attraction zone gradually reduces the movement of sensors. Consequently, this will decrease the moving distance, which in return leads to reducing the energy consumed during the displacement of sensors to their final positions. Besides, the results show that the energy consumption of BAGOA is almost stable even with the increase in the number of nodes.

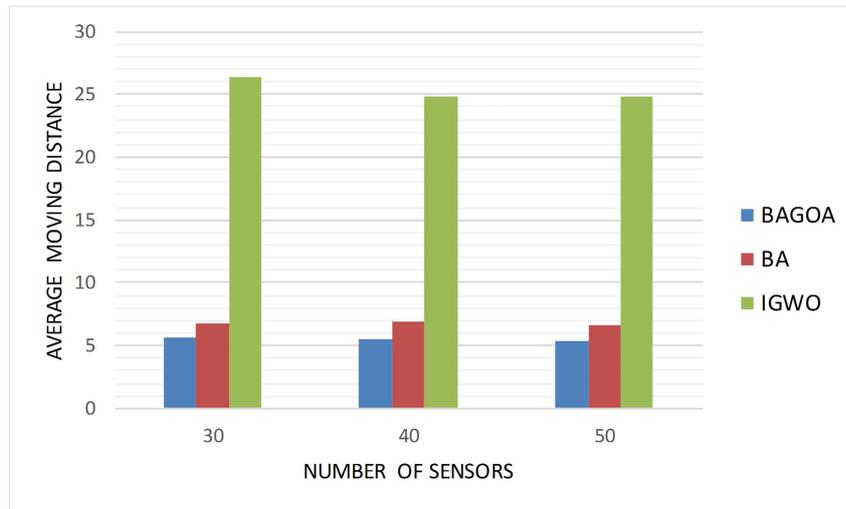


Figure 7: Average moving distance comparison between BA, IGWO, and BAGOA.

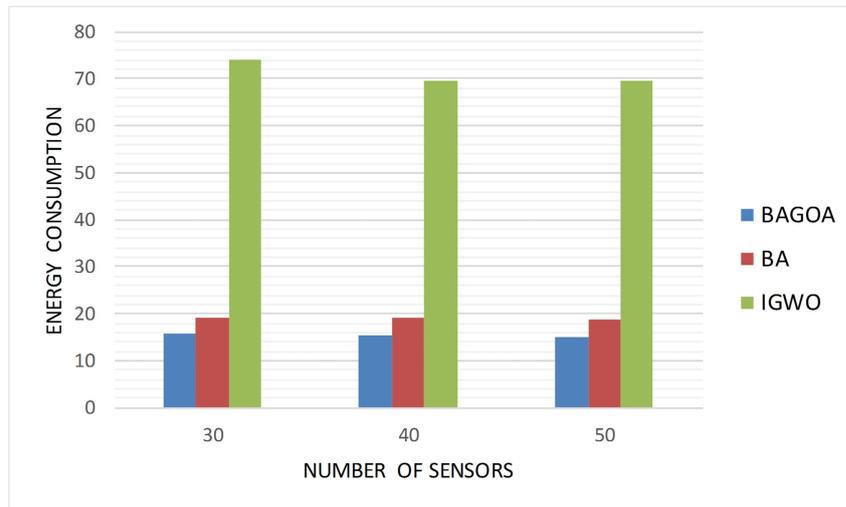


Figure 8: Energy consumption comparison between BA, IGWO, and BAGOA.

The second experiment was designed to observe the performance of the BAGOA in optimizing the WSN coverage when the deployment is scaled over a large geographical region. For this purpose, an

area of size $100 m \times 100 m$ is used with a perceptual radius equals to $7 m$, and a number of sensors varying between 20 and 100. In this experiment, the qABC is utilized as a comparison algorithm with the BAGOA. Each algorithm was executed 10 times with a maximum number of iterations equals to 200. The best and average results are reported in Table 4.

Table 4: Deployment results comparison between two algorithms in an area of size 100m by 100m.

Number of sensors	BAGOA				qABC			
	Best(%)	Mean(%)	Std	Avg Overlapp area	Best(%)	Mean(%)	Std	Avg Overlapp area
20	31.27	31.26	0.014	0	31.20	31.17	0.016	1.0000e-03
40	62	61.74	0.134	0,272	60.77	60.18	0.389	0.9780
60	84.19	83.68	0.340	6,171	82.26	80.48	1.072	8.2020
80	95.47	94.67	0.449	21,542	93.43	91.72	0.831	22.8320
100	99.08	98.75	0.206	41,407	97.77	97.12	0.461	40.3460

The results of Table 4 show that BAGOA again performs better in optimizing the coverage, mean (average coverage), and standard deviation values, also it minimizes the average overlapping area of the network. To be more specific, the BAGOA provides 31.27%, 62%, 84.19%, and 95.47% of coverage when deploying 20, 40, 60, and 80 sensors respectively. In addition, it achieves the best approximation of the full coverage with 100 sensors, where it reaches 99.08% with a difference of 1.31% from qABC. The results indicate that the proposed algorithm maintains the best deployment performance for a large area of size $100 m \times 100 m$. Besides, by observing Table 4, the BAGOA has a better mean and standard deviation values compared with the qABC in all the test cases, which confirms its stability and reliability. Moreover, it contributes to minimizing the overlapping area in the majority of the test cases except when the number of sensors is 100. It is well known that maximizing the coverage results in a decrease in the overlapping area. However, when the network is crowded and there are too many nodes deployed close to each other, the possibility for sensors to be overlapped will increase. For instance, when a coverage hole is between a set of sensors, the BAGOA requires them to overlap in order to eliminate that coverage hole. Therefore, maximizing coverage when a large number of sensors are deployed (100 sensors in our case) triggers a supplementary cost by increasing the overlapping area. Generally, as can be seen from Table 4, the BAGOA performs better in maximizing the coverage and reducing the overlapping area between sensors to a minimum value.

In the last experiment, we analyze the performance of the BAGOA in enhancing the coverage of a mixed wireless sensor network. For this purpose, a set of simulations are conducted with a WSN containing both mobile and stationary sensors. In the simulations, a total of 100 sensors with 80 stationary sensors and 20 mobile sensors are deployed in an area of size $100 m \times 100 m$, the perceptual radius is set to $7 m$, and the number of iterations is 100. The BBO algorithm used in [21] to solve this problem is chosen to make a comparison with BAGOA. At first, the stationary sensors are deployed in the sensing area, and then the algorithms will attempt to optimize the network coverage by changing the positions of mobile sensors. To obtain a fair comparison, it should be noted that the algorithms start with the same initial positions of stationary sensors. The algorithms were simulated for 10 runs, and the collected results are presented in Table 5.

Table 5: Deployment results comparison between two algorithms in the case of mobile and static sensors.

Algorithms	Initial coverage of stationary sensors	Best(%)	Mean(%)	Std
BAGOA	69.01	92.47	91.86	0.321
BBO	69.01	90.06	88.86	0.599

It can be seen from Table 5 that BAGOA offers excellent coverage results throughout the experiment.

More specifically, the BAGOA was successful in optimizing the coverage where it reaches 92.47%, with an increase of 23.46%. On the other hand, the BBO algorithm has an increase of 21.05%, which is 2.41% lower than BAGOA. Besides, the BAGOA outperforms the BBO in terms of average results with a difference of 3%. According to the collected results, the BAGOA is superior to the other competitor in optimizing the coverage in the case of static and mobile sensors. Moreover, the BAGOA provides a significant reduction in standard deviation value, which proves that it maintains a stable performance during the simulations.

Figure 9 shows the best deployment solutions of the two algorithms starting with the same initial deployment of stationary sensors. As can be seen from Figure 9c, the BAGOA was successful in maximizing the initial coverage of the network by properly choosing a set of optimal positions for the 20 mobile sensors.

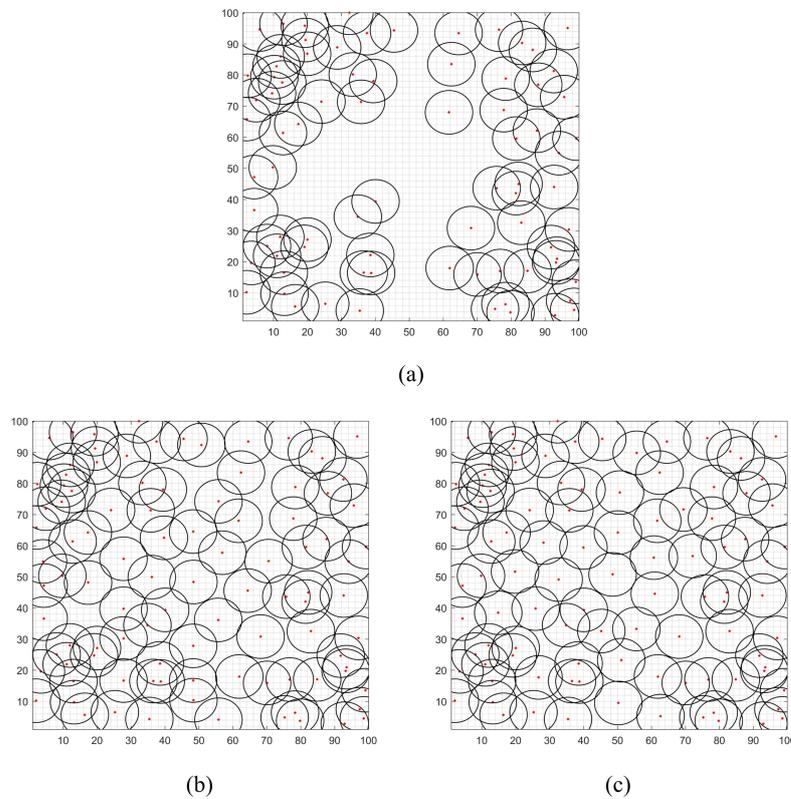


Figure 9: (a) Initial deployment of stationary sensors, (b) Final sensor distribution of BBO, and (c) Final sensor distribution of BAGOA.

Regarding energy, as can be observed from Figures 10 and 11, BAGOA once again proves its efficiency in minimizing the amount of energy consumed during the displacement of sensors. The energy consumed by the sensors using BAGOA is less than half the energy required to move the sensors to their final positions using BBO. BAGOA offers a significant contribution to saving energy by deploying the mobile sensors in a way that optimizes the overall coverage and minimizes energy consumption, thereby, prolonging the network lifetime. The superior results are due to the shrinking of movement boundaries performed by the BAGOA that contributes to decreasing the moving distance, where it does not allow movement steps greater than the neighborhood size. Moreover, the tendency toward the best agent in the neighborhood decreases the motion rate of sensors, which leads to avoiding the fast depletion of their energy and allows the sensors to remain functional for a longer time.

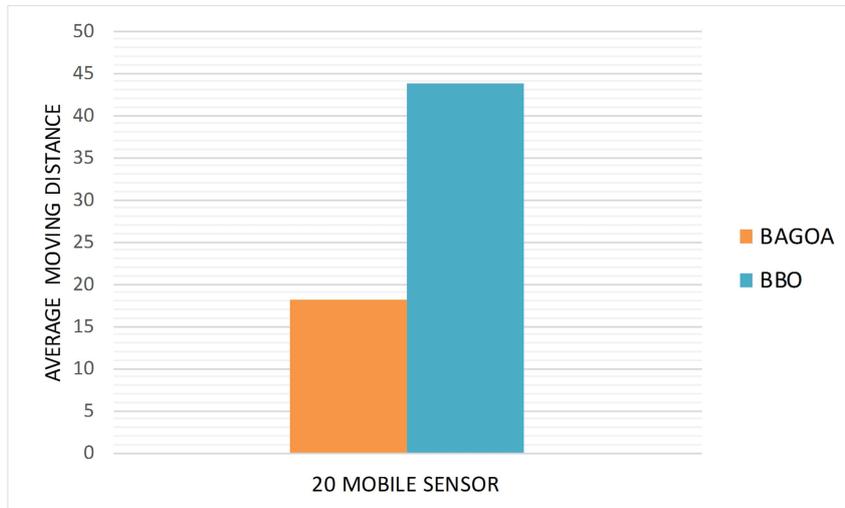


Figure 10: Average moving distance comparison between BBO and BAGO.

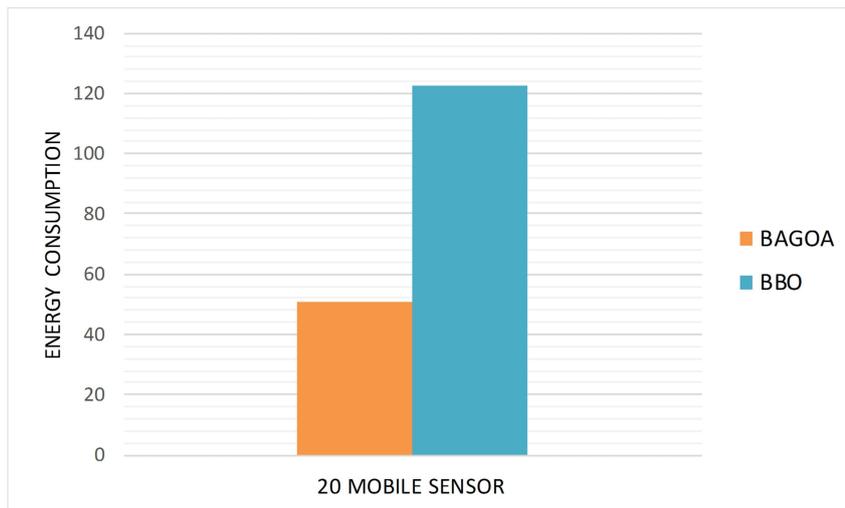


Figure 11: Energy consumption comparison between BBO and BAGO.

8 Conclusion

In this paper, a hybrid algorithm based on the Bees Algorithm and Grasshopper Optimization Algorithm named BAGO is proposed to solve the problem of deployment optimization in WSNs. The proposed BAGO algorithm utilizes the strength of the GOA to enhance the exploitative capability of the BA by searching the neighborhood more efficiently instead of the blind random search of the basic BA. By hybridizing the two algorithms, the BAGO achieves a significant acceleration in the convergence and an increase in the search accuracy. As an outcome, it led to superior results when solving the problem of deployment. The conducted comparative studies proved the efficiency of the BAGO in optimizing the deployment of WSNs compared with other state-of-the-art algorithms where it provides excellent coverage results throughout the experiments. Furthermore, BAGO is an energy-efficient algorithm, which contributes to maximizing the network lifetime by minimizing the amount of energy consumed by the sensors to reach their optimal positions. Besides, the superior results achieved by BAGO in the

three experiments show that it has excellent adaptability in solving different deployment problems under different experimental settings.

For future works, the BAGOA can be used to optimize several objectives such as cost and connectivity with multiple types of sensors.

References

- [1] Mohammed Farsi, Mostafa A Elhosseini, Mahmoud Badawy, Hesham Arafat Ali, and Hanaa Zain Eldin. Deployment techniques in wireless sensor networks, coverage and connectivity: A survey. *IEEE Access*, 7:28940–28954, 2019.
- [2] Vilabha S Patil, Yashwant B Mane, and Shraddha Deshpande. Fpga based power saving technique for sensor node in wireless sensor network (wsn). In *Computational Intelligence in Sensor Networks*, pages 385–404. Springer, 2019.
- [3] Paolo Bellavista, Giuseppe Cardone, Antonio Corradi, and Luca Foschini. Convergence of manet and wsn in iot urban scenarios. *IEEE Sensors Journal*, 13(10):3558–3567, 2013.
- [4] Ines Khoufi, Pascale Minet, Anis Laouiti, and Saoucene Mahfoudh. 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, 2017.
- [5] Shaimaa M Mohamed, Haitham S Hamza, and Iman Aly Saroit. Coverage in mobile wireless sensor networks (m-wsn): A survey. *Computer Communications*, 110:133–150, 2017.
- [6] Anju Sangwan and Rishi Pal Singh. Survey on coverage problems in wireless sensor networks. *Wireless Personal Communications*, 80(4):1475–1500, 2015.
- [7] Mustapha Reda Senouci, Abdelhamid Mellouk, and Amar Aissani. Random deployment of wireless sensor networks: a survey and approach. *International Journal of Ad Hoc and Ubiquitous Computing*, 15(1-3):133–146, 2014.
- [8] Huynh Thi Thanh Binh, Nguyen Thi Hanh, Nguyen Duc Nghia, Nilanjan Dey, et al. Metaheuristics for maximization of obstacles constrained area coverage in heterogeneous wireless sensor networks. *Applied Soft Computing*, 86:105939, 2020.
- [9] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Abdolreza Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2):495–513, 2016.
- [10] Alireza Askarzadeh. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*, 169:1–12, 2016.
- [11] Seyedeh Zahra Mirjalili, Seyedali Mirjalili, Shahrzad Saremi, Hossam Faris, and Ibrahim Aljarah. Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, 48(4):805–820, 2018.
- [12] Duc Truong Pham, Afshin Ghanbarzadeh, Ebubekir Koç, Sameh Otri, Shafqat Rahim, and Muhamad Zaidi. The bees algorithm a novel tool for complex optimisation problems. In *Intelligent production machines and systems*, pages 454–459. Elsevier, 2006.
- [13] Ercan Oztemel and Ayse Aycim Selam. Bees algorithm for multi-mode, resource-constrained project scheduling in molding industry. *Computers & Industrial Engineering*, 112:187–196, 2017.
- [14] Shafie Kamaruddin and Mohd Arif Hafizi Abd Latif. Application of the bees algorithm for constrained mechanical design optimisation problem. *International Journal of Engineering Materials and Manufacture*, 4(1):27–32, 2019.

- [15] Marco Castellani, Sameh Otri, and Duc Truong Pham. Printed circuit board assembly time minimisation using a novel bees algorithm. *Computers & Industrial Engineering*, 133:186–194, 2019.
- [16] Majdi Mafarja, Ibrahim Aljarah, Hossam Faris, Abdelaziz I Hammouri, Al-Zoubi Ala’M, and Seyedali Mirjalili. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, 117:267–286, 2019.
- [17] Zhendong Wang, Huamao Xie, Zhongdong Hu, Dahai Li, Junling Wang, and Wen Liang. Node coverage optimization algorithm for wireless sensor networks based on improved grey wolf optimizer. *Journal of Algorithms & Computational Technology*, 13:1748302619889498, 2019.
- [18] Recep ÖZDAĞ and Murat CANAYAZ. 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), 2017.
- [19] Beyza Gorkemli and Zahraa Al-Dulaimi. On the performance of quick artificial bee colony algorithm for dynamic deployment of wireless sensor networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 27(6):4038–4054, 2019.
- [20] Tripatjot Singh Panag and Jaspreet Singh Dhillon. Maximal coverage hybrid search algorithm for deployment in wireless sensor networks. *Wireless Networks*, 25(2):637–652, 2019.
- [21] Gaige Wang, Lihong Guo, Hong Duan, Luo Liu, and Heqi Wang. Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm. *Journal of Sensor and Actuator Networks*, 1(2):86–96, 2012.
- [22] Yongquan Zhou, Ruxin Zhao, Qifang Luo, and Chunming Wen. Sensor deployment scheme based on social spider optimization algorithm for wireless sensor networks. *Neural Processing Letters*, 48(1):71–94, 2018.
- [23] Mudassar Ali Syed and Raziuddin Syed. Weighted salp swarm algorithm and its applications towards optimal sensor deployment. *Journal of King Saud University-Computer and Information Sciences*, 2019.
- [24] Eva Tuba, Milan Tuba, and Marko Beko. Mobile wireless sensor networks coverage maximization by firefly algorithm. In *2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 1–5. IEEE, 2017.
- [25] M Rajeswari, J Amudhavel, and P Dhavachelvan. Moth-flame algorithm for solving coverage problem in wireless sensor network. *Advances and Applications in Mathematical Sciences*, 17(1):51–64, 2017.
- [26] Duc T Pham, Luca Baronti, Biao Zhang, and Marco Castellani. Optimisation of engineering systems with the bees algorithm. *International Journal of Artificial Life Research (IJALR)*, 8(1):1–15, 2018.
- [27] Duc Truong Pham and Marco Castellani. A comparative study of the bees algorithm as a tool for function optimisation. *Cogent Engineering*, 2(1):1091540, 2015.
- [28] Mohamed Amine Nemnich, Fatima Debbat, and Mohamed Slimane. Hybrid bees approach based on improved search sites selection by firefly algorithm for solving complex continuous functions. *Informatica*, 44(2), 2020.
- [29] Ahmed A Ewees, Mohamed Abd Elaziz, and Essam H Houssein. Improved grasshopper optimization algorithm using opposition-based learning. *Expert Systems with Applications*, 112:156–172, 2018.
- [30] Laith Abualigah and Ali Diabat. A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. *Neural Computing and Applications*, pages 1–24, 2020.
- [31] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software*, 105:30–47, 2017.

-
- [32] Laith Abualigah. Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Computing and Applications*, pages 1–24, 2020.
- [33] J Amutha, Sandeep Sharma, and Jaiprakash Nagar. Wsn strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. *Wireless Personal Communications*, 111(2):1089–1115, 2020.
- [34] Yijie Zhang and Mandan Liu. Regional optimization dynamic algorithm for node placement in wireless sensor networks. *Sensors*, 20(15):4216, 2020.
- [35] Osamah Ibrahim Khalaf, Ghaida Muttashar Abdulsahib, and Bayan Mahdi Sabbar. Optimization of wireless sensor network coverage using the bee algorithm. *J. Inf. Sci. Eng.*, 36(2):377–386, 2020.
- [36] Selina Sharmin, Fernaz Narin Nur, Md Abdur Razzaque, Md Mustafizur Rahman, Abdulhameed Alelaiwi, Mohammad Mehedi Hassan, and Sk Md Mizanur Rahman. α -overlapping area coverage for clustered directional sensor networks. *Computer Communications*, 109:89–103, 2017.
- [37] Mohamed Amin Benatia, M’hammed Sahnoun, David Baudry, Anne Louis, Abdelkhalak El-Hami, and Belahcene Mazari. Multi-objective wsn deployment using genetic algorithms under cost, coverage, and connectivity constraints. *Wireless Personal Communications*, 94(4):2739–2768, 2017.
- [38] Zhendong Wang, Huamao Xie, Daojing He, and Sammy Chan. Wireless sensor network deployment optimization based on two flower pollination algorithms. *IEEE Access*, 7:180590–180608, 2019.
- [39] Alia Ghaddar, Monah Bou Hattoum, Ghassan Fadlallah, and Hamid Mcheick. Mccm: An approach for connectivity and coverage maximization. *Future Internet*, 12(2):19, 2020.