

Article

Efficient WSN Node Placement by Coupling KNN Machine Learning for Signal Estimations and I-HBIA Metaheuristic Algorithm for Node Position Optimization

Bastien Poggi ^{1,*} , Chabi Babatounde ² , Evelynne Vittori ¹  and Thierry Antoine-Santoni ³ 

¹ UMR CNRS 6134 SPE, University of Corsica, 20250 Corte, France

² UAR CNRS 3514 Stella Mare, University of Corsica, 20250 Corte, France

³ UMR CNRS 6240 LISA, University of Corsica, 20250 Corte, France

* Correspondence: poggi_b@univ-corse.fr

Abstract: Wireless sensor network (WSN) deployment is an intensive field of research. In this paper, we propose a novel approach based on machine learning (ML) and metaheuristics (MH) for supporting decision-makers during the deployment process. We suggest optimizing node positions by introducing a new hybridized version of the “Hitchcock bird-inspired algorithm” (HBIA) metaheuristic algorithm that we named “Intensified-Hitchcock bird-inspired algorithm” (I-HBIA). During the optimization process, our fitness function focuses on received signal maximization between nodes and antennas. Signal estimations are provided by the machine learning “K Nearest Neighbors” (KNN) algorithm working with real measured data. To highlight our contribution, we compare the performances of the canonical HBIA algorithm and our I-HBIA algorithm on classical optimization benchmarks. We then evaluate the accuracy of signal predictions by the KNN algorithm on different maps. Finally, we couple KNN and I-HBIA to provide efficient deployment propositions according to actual measured signal on areas of interest.

Keywords: wireless sensor network; deployment; optimization; KNN; HBIA



Citation: Poggi, B.; Babatounde, C.; Vittori, E.; Antoine-Santoni, T. Efficient WSN Node Placement by Coupling KNN Machine Learning for Signal Estimations and I-HBIA Metaheuristic Algorithm for Node Position Optimization. *Sensors* **2022**, *22*, 9927. <https://doi.org/10.3390/s22249927>

Academic Editor: Tian Wang

Received: 12 September 2022

Accepted: 29 November 2022

Published: 16 December 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the past two decades, the number of deployed wireless sensors has increased dramatically. This spectacular growth is on the one hand correlated with the development of machine learning applications requiring large amounts of data and on the other hand made possible by the reduction in the production costs of the equipment.

Following this trend, many new technologies have emerged supporting different routing protocols, different operating systems, and different micro-controller boards. They have been successfully applied in several fields of research, and application such as environmental monitoring [1,2], smart agriculture [3], health [4], and security [5].

This current heterogeneity of technologies generates new challenges [6]. To deal with it, researchers have to provide new tools to help decision-makers during WSN deployment. Many issues need to be addressed: “Which technology is the most suitable for my problem?”; “How to configure WSN parameters?”; “Where to deploy sensors in order to observe a specific phenomenon?”; “How to ensure node connectivity within the studied area according to its specificities and constraints?”.

One of the most encountered problems with WSN is their efficient placement in an area of interest (AoI). On star topology, all sensors must receive signals from one or more fixed antennas. Many times these antennas are placed in specific places accepted by the population. Signal propagation is rarely the main criterion considered. Citizens usually focus first on other criteria, such as visual impact and health protection. Node localization must reconcile these constraints and provide a minimum level of connectivity.

2. Wireless Sensor Network Deployment

2.1. WSN Deployment Problematic

A WSN generally consists of two groups of entities: the gateways and the sensors. The gateways, or “sink nodes”, are usually antennas to which the collected data converge. They do not collect physical data but rather act as a relay between the WSN and other larger networks such as the Internet or an organization’s intranet. The sensors are composed of four parts: the acquisition unit, which allows physical collection of the observed phenomena; the calculation unit, which allows the sensor’s operating system and applications to function; the power supply unit, which provides energy to the various components; and the transmission unit, which allows the data to be transferred to the gateway or to other relay nodes. These devices will measure physical values and transmit them directly or by hops to the gateways.

WSN deployment quality is a highly subjective concept [7] depending on several factors: the application field, the integrated software technologies, the hardware technologies, the legislative constraints, as well as the access constraints to the deployment areas. The chosen deployment strategy can indirectly optimize different features of a WSN, including network coverage, network connectivity, network robustness, and network reactivity. Node localization is also an active field of research for WSN, which includes three groups of approaches: “range based algorithms” [8–10], “range free algorithms” [11–13], and “hybrid algorithms” [14]. The first group is based on signal physical information analysis such as time of arrival (TOA) or received signal strength indicator (RSSI). The second group is based on neighborhood analysis, with several methods such as DV-hop, APIT, or sequence-based algorithms. The third group is a combination of the first two. All these methods focus on network connectivity.

Several definitions of connectivity can be found in the literature [15–17], depending on the network topology used. Received and transmitted signal estimations are generally based on the calculation of Euclidean distances between nodes. For mesh topologies, the minimum number of hops between nodes to communicate with gateways is estimated and neighborhood analysis is performed. For star topologies (i.e., LoRa technology), the strongest received signal from different antennas describes connectivity quality.

Currently, two families of deployment strategies can be found in the literature: the “classical” approaches based on geometry and the “stigmergic” ones based on guided randomness.

2.1.1. Classical Approaches

The problem of deploying wireless sensor networks has emerged following the spread of this technology. The first deployment assistance tools were based on three kinds of methods. The first one was based on deployment grids [18,19]. The second relied on the concepts of attraction and repulsion forces between the nodes of the network [20,21], and the third one was based on “computational geometry” with Delaunay triangulation and Voronoi diagrams [22,23]. An example of such a deployment is shown in Figure 1.

Although these approaches have been frequently used successfully, they have some disadvantages. In return for their fast execution time, they are based on a theoretical approximation of the signal, making it challenging to consider the deployment zone’s specificities and dynamic aspects. Moreover, their adaptation to the particularities of the studied problem requires critical adaptation times and in-depth knowledge of the deployment zone. In other cases, these specificities are simply ignored, negatively impacting the proposed scenarios’ feasibility. Faced with these important limitations, their use is becoming less and less common in the literature. New methods based on guided random reasoning bypass these limitations.

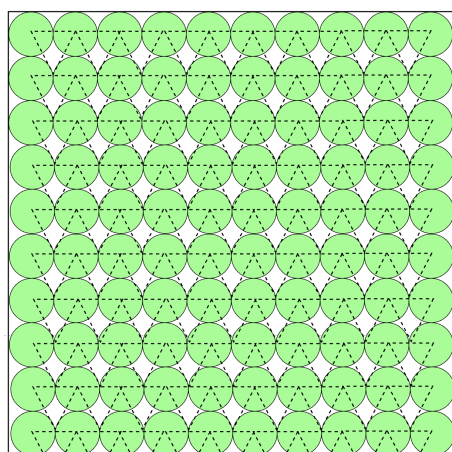


Figure 1. An example of a classical deployment strategy.

2.1.2. Modern Approaches

WSNs are a recent technology with many research directions [6]. The deployment of a WSN is an NP-complete problem [24]. Indeed, the geographical position of the sensors will influence many parameters of the WSN, such as its lifetime, fault tolerance, speed of detection of studied phenomena, as well as the accuracy of the measurements. There is no exact algorithm to solve all these objectives in a global and confident way.

Nevertheless, in order to meet this need for “intelligent deployments”, optimization methods known as “metaheuristics” are regularly used to respond to the problems encountered by the WSN [25].

These methods are a set of nature-inspired algorithms for minimizing or maximizing the parameters of an objective function (i.e., maximizing network coverage). They are regularly used with success in the field of WSN. Examples include localizing captured phenomena [26–29], creating and configuring routing protocols [30–35], or creating clusters to maximize network lifetime while minimizing energy consumption [36–45].

Due to their high genericity and adaptability, they are also regularly used for optimizing WSN deployments [46–60].

These modern, iterative optimization methods are based on two central concepts: “intensification” and “diversification”. They start with a random generation of solutions in the search space under study, followed by a guided search according to the results provided for an evaluation function representing the answer to the studied problem. Several hundred so-called “metaheuristic” methods are available in the literature today, as well as numerous variants. This phenomenon generates a certain complexity in the choice of the method for a given problem [61]. Moreover, their performance depends on many factors, such as the quality of the algorithm implementation, parameterization, the random generator used, and solution representation. There are many sources of inspiration for these methods, such as evolutionary theory [62,63], swarms [64–68], physics [69–71], or human behavior [72–74]. Examples of historically older metaheuristics are given in Table 1, and more recent methods are shown in Table 2.

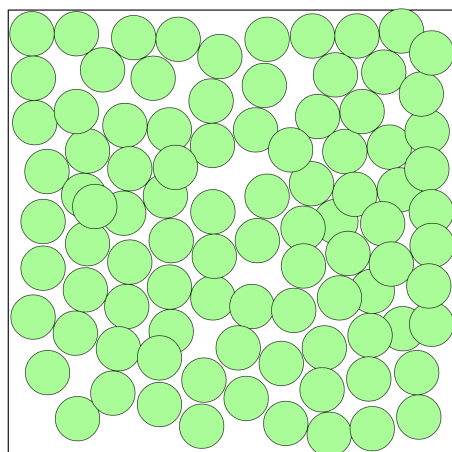
Table 1. Some most used metaheuristics.

| Name | Publication Date | Inspiration |
|----------------------------------|------------------|--------------|
| Simulated annealing [75] | 1983 | Physics |
| Differential evolution [62] | 1997 | Evolutionary |
| Particle swarm optimization [65] | 1995 | Swarm |
| Genetic algorithms [63] | 1962 | Evolutionary |
| Ant colony optimization [64] | 2006 | Swarm |

Table 2. Some promising recent metaheuristics.

| Name | Publication Date | Inspiration |
|-----------------------------------|------------------|-------------|
| Grey wolves optimizer [66] | 2014 | Swarm |
| Whale optimization algorithm [67] | 2016 | Swarm |
| Ant lion optimization [76] | 2015 | Swarm |
| Moth-flame optimization [68] | 2015 | Swarm |
| Sine cosine algorithm [77] | 2016 | Mathematics |

These optimization methods are not guaranteed to find the optimal deployment but an acceptable deployment in a reasonable time, in contrast to classical optimization methods whose computation time on such problems would exceed a thousand years. The main advantage of this method is the rapid integration of the different criteria for estimating the deployment quality into the evaluation function of the problem. An example of such a deployment is shown in Figure 2 with the evaluation function of maximizing area coverage. In this figure, the maximum coverage is not reached, but the proposed solution is very close to it.

**Figure 2.** Example of random deployment strategy.

Therefore, a fundamental question arises to guarantee the success of the proposed deployment: which evaluation function to choose? In the case of our work, this evaluation function will be strongly conditioned by the signal quality between sensors and gateways. However, this signal quality is difficult to estimate by simulations. Accurate measurements are generally necessary, but these are rarely exhaustive. It is, therefore, essential to propose an approach that combines the deployment's optimization with an accurate signal estimation based on measurements collected in situ.

3. Proposed Approach

3.1. Coupling a Metaheuristic with a KNN Evaluation Function

Our method is based on two basic concepts. The first one is the optimization of the sensor position by using a metaheuristic. The second one is an evaluation function based on signal quality estimations at the chosen location. This estimation is based on the KNN algorithm and real measures from the area of interest. This operation is explained in a simplified way in Figure 3.

In the example in Figure 3, we can see different signal qualities collected by field measurements. We can see that the antenna's signal is lower to the northeast of the antenna than to the southeast of the antenna, probably due to obstacles. If we want to place a node on the area of interest and ensure that it is connected to the antenna, its position will be the one represented by the yellow diamond. This process is repeated on different areas of

the map. We observe that it is possible to meet the dual objective of “collecting data in a specific area” (sensing) and “routing the data to the gateway” (transmitting).

Deployment Area

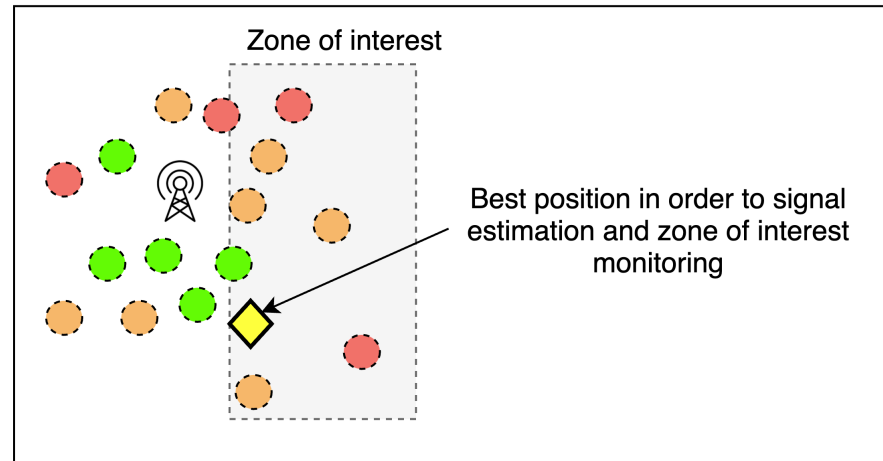


Figure 3. Example of optimal proposed position for a node.

3.2. Signal Estimation with KNN Algorithm

Many machine learning algorithms are available in the literature today. They can be split into two groups: “supervised algorithms” for classification and regression and “unsupervised algorithms” for clustering and dimensionality reduction. Optimization methods require a precise evaluation function to be minimized or maximized according to the problem to be solved. In our approach, we chose to use machine learning to “predict” by regression the quality of a signal on a studied area. Different supervised learning algorithms can be used for that purpose [78,79]. Most of these algorithms, such as neural networks (NN), require a training process. Moreover, they are challenging to explain [80], and many parameters need a tuning process [81].

However, some are simpler to implement and provide acceptable short calculation time. The KNN algorithm is a “lazy learning” machine learning method [82,83]. This algorithm estimates a point’s value according to its neighbors by computing average distances between instances, as shown in Figure 4 and Equation (1). We assume this concept is suitable for signal estimation on geolocated nodes inside a deployment area. For example, if neighbors B and C of node A have a good-quality signal, we can assume that node A will also have a good-quality signal without considering all nodes inside the deployment area.

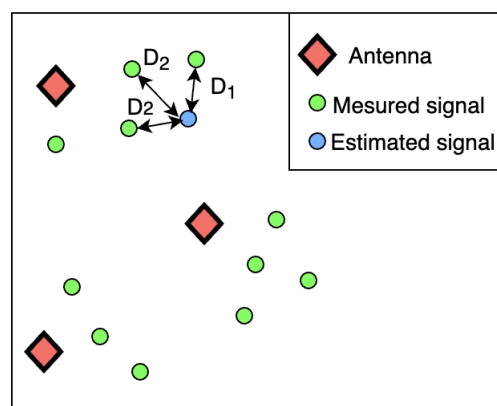


Figure 4. Signal estimation with KNN algorithm ($K = 3$).

$$estimated_signal_x = \frac{(\sum_{i=1}^K Neighbor_i, x)}{K} \quad (1)$$

3.3. Optimize Node Location with I-HBIA Metaheuristic

3.3.1. Canonical HBIA Algorithm

In our approach, we have chosen to use the HBIA algorithm [84]. The behavior of the birds in Alfred Hitchcock's famous film *The Birds* inspires this recent metaheuristic. As stated in the “no free lunch” theorem, predicting which metaheuristic will perform best on a given problem is impossible [61]. Nevertheless, there are three main reasons for the choice of this algorithm for our approach:

- This algorithm belongs to the family of P-Metaheuristics that base the optimization process on a set of solutions and are generally more efficient for optimization problems containing local minimums.
- This algorithm, in contrast to other metaheuristics, does not require many parameters. The only parameterization concerns the number of birds found in the colony. Moreover, the impact of the variations of this parameter on the results obtained remains marginal.
- This algorithm offers superior performance to many other metaheuristics on high dimensional problems. This aspect is particularly interesting for deploying sensor networks where the number of nodes and their positions can quickly become very large.

The intelligent behavior of birds during attacks inspired the authors of the HBIA algorithm. Each solution is represented as a bird in the colony. In the rest of this paper, we will use the following conventions: B for the whole bird colony, B_i for a specific bird, and B_* for the best bird.

As summarized in Figure 5, the algorithm starts with a random solution generation phase. A lurking phase is performed to avoid too-fast convergence of the algorithm towards a local minimum. It allows for the birds B_i to move away from the best bird B_* , allowing the search to diversify.

Once the lurking phase is over, the birds will be evaluated in order to designate them to one of three categories: the best, the elites, and the others. Only one best bird is defined in the algorithm. The number of birds belonging to the elite is estimated according to the colony's size. Once this distribution is complete, the birds will carry out successive waves of attacks. Two types of attack are possible depending on their relative distance from the best bird. The first will approach the best bird directly in the search area, while the second will take a less direct route. These attacks will allow birds to get progressively closer to the best bird without missing the diversification during the search. During optimization, a mechanism based on a dynamic parameterization of aggressiveness will ensure an efficient modulation between intensification and diversification. If a bird does not progress during optimization, its aggressiveness will increase. This aggressiveness will be correlated with the level of diversification of its search. On the other hand, if its evaluation improves, its aggressiveness will remain stable.

When a phase of “stagnation” is identified, a phase of reorganization of the colony starts. It will lead to the deletion of the solutions with the worst results. The best birds and the elite birds will survive. For the others, their probability of survival will be proportional to their evaluation, and the worse their rating, the greater their risk of dying. Dead birds will be removed from the colony and replaced by new birds. For this purpose, the algorithm will consider previously generated variables that led to better solutions than the previous ones.

This process is repeated until the maximum iteration number is reached or the evaluation function's result exceeds a specified threshold. When the optimization process ends, the proposed deployment coordinates are saved or displayed to the decision-makers.

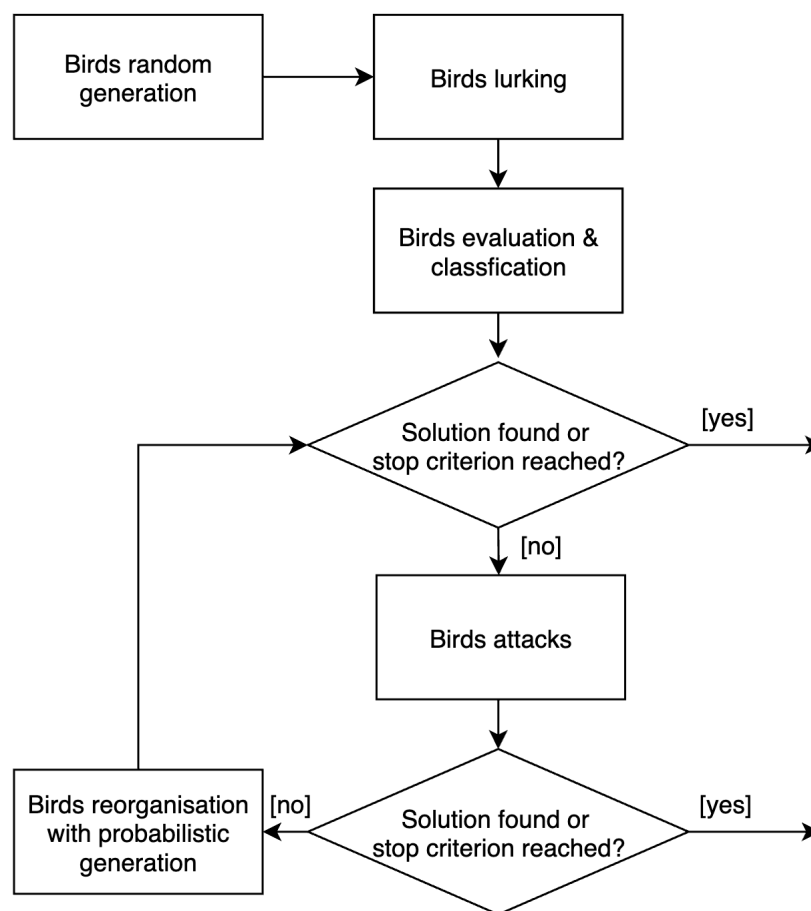


Figure 5. The different steps of the HBIA optimization algorithm.

To efficiently apply the HBIA algorithm to the specificity of the search space of a WSN deployment, a modified version of the algorithm has been designed: I-HBIA (“Intensified-Hichcok Bird Optimization Algorithm”). This hybrid algorithm includes a local search process to decrease the number of optimization iterations and increase solution quality.

3.3.2. The I-HBIA Algorithm

The optimization process by a metaheuristic presents some specificities compared to other optimization problems. The first step of a metaheuristic is a systematically random generation. Generally, this type of deployment offers acceptable solutions when the objective of the deployment is to maximize coverage. The sensors will be uniformly deployed over the area. The optimization process will therefore focus during the different iterations on refining a solution that is already correct from the first iteration.

In the classical version of the algorithm, the best bird does not evolve during iterations. In the I-HBIA algorithm, we have therefore chosen to add a step that runs in parallel with the different attacks: “cloning the best bird,” as described in Figure 6. This step consists of duplicating the best bird a number of times corresponding to the number of dimensions studied. For example, a problem of dimension 50 will generate the creation of 50 clones. These clones will be more or less faithful replicas of the best bird. Indeed, they will be subjected to random perturbations. Each clone will be assigned a mutation number corresponding to its replication number, as shown in Figure 7.

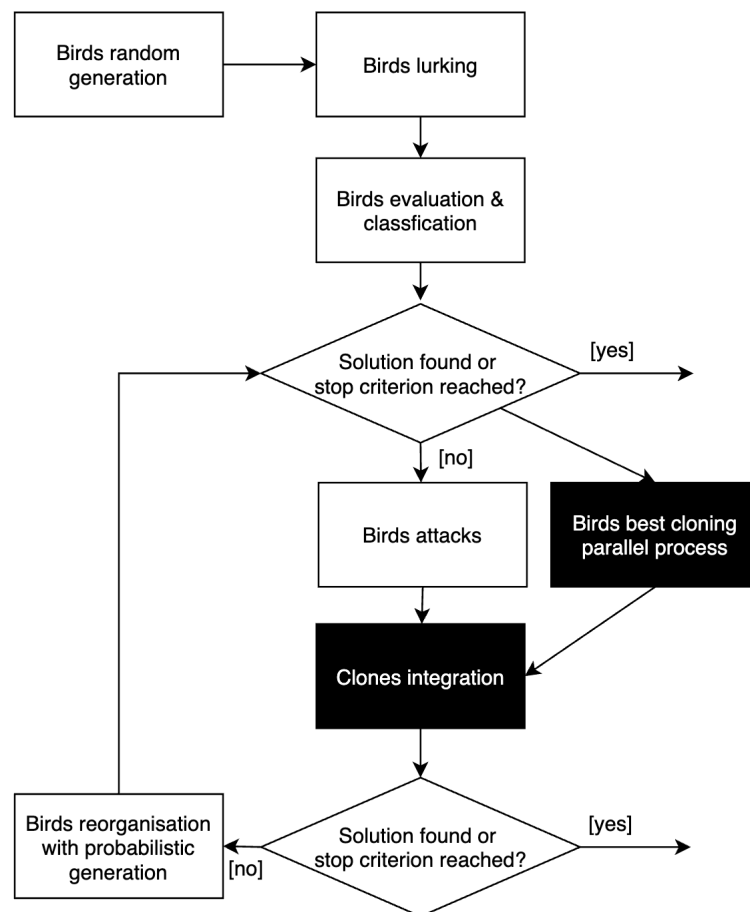


Figure 6. The different steps of I-HBIA hybrid optimization algorithm.

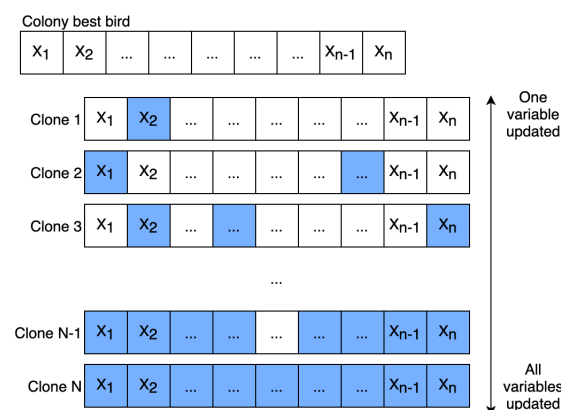


Figure 7. Intensified research step in I-HBIA.

This hybrid version of the algorithm allows two things. It will enable faster convergence of the algorithm. It also allows us to get closer to the optimum, especially when the number of dimensions is large (more than 30 dimensions). The following section presents the different results obtained.

4. Results

4.1. KNN Algorithm Parametrization

KNN algorithm parametrization is a complex task generally based on empirical observations. The parameter K is the number of considered neighbors to estimate the signal

quality. If the amount of measured data is limited, the algorithm's performance can be volatile and challenging to measure, as shown in Figure 8. We can see in this figure that the prediction precision is variable when the algorithm works with 200 or 250 measured values. The volatility of results is less intense in scenarios where we have a large volume of data, such as 900 or 1000 measured values.

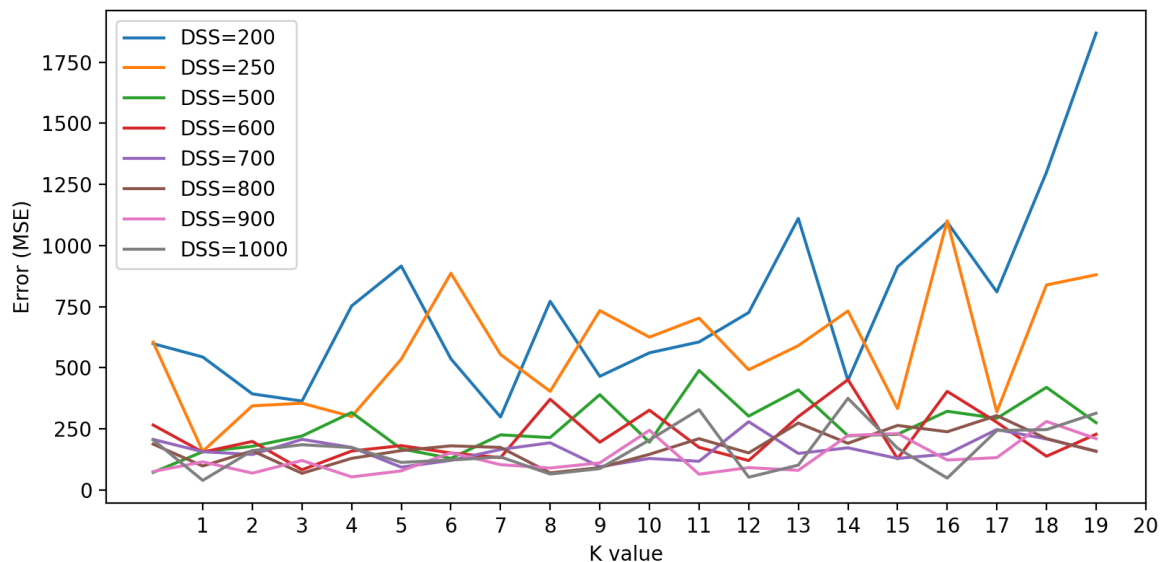


Figure 8. Errors for different dataset size (DSS) with different K values.

To find the best value of K, we test the algorithm performance on different maps with different measures. The various parameters used for this test are described in Table 3. As shown in Figure 9, the better performance is provided by $K = 2$. Other values of K provide acceptable results but need more computation time to compute the neighborhood.

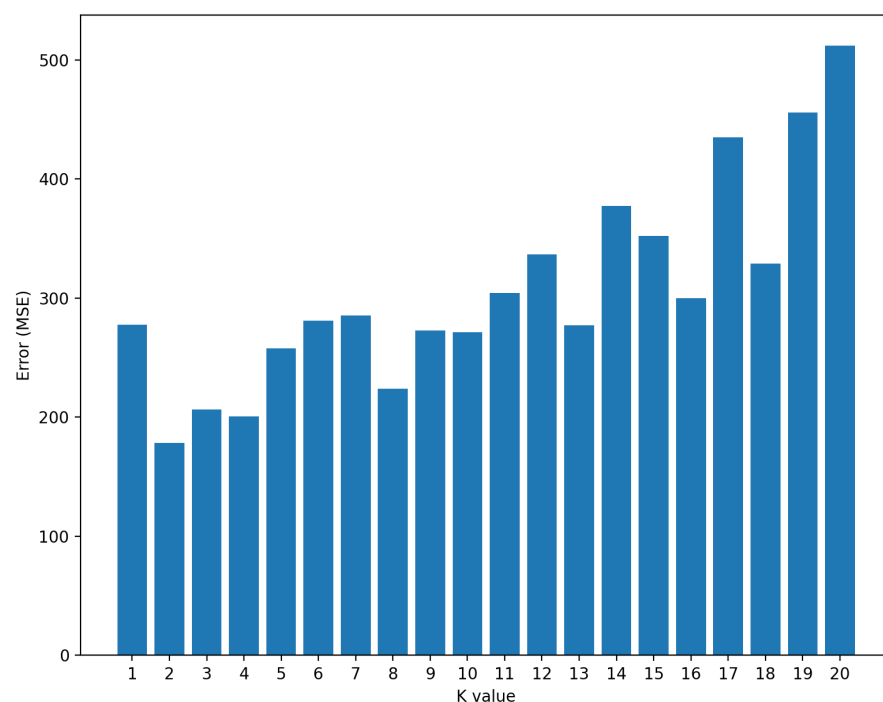


Figure 9. Estimated signal prediction error with different K values.

Table 3. Parameters used to find best K value.

| Parameters | Values |
|------------------------------|---|
| Number of measures | [200, 250, 500, 600, 700, 800, 900, 1000] |
| K values | [1–20] |
| Number of repetitions | 20 |
| Considering learning dataset | 80% |
| Considering testing dataset | 20% |

4.2. Signal Estimations on Maps

To evaluate the performance of the KNN algorithm for signal quality estimation, we have generated different signal maps. These signal maps are of two types: “without obstacles” and “with obstacles”. Maps estimate the circular signal around the antennas with a linear signal strength attenuation. These signal representations can easily be updated in our architecture to consider the specificities of the hardware used in a given deployment.

4.2.1. Signal Map without Obstacles

Figure 10 demonstrates the functioning of the KNN algorithm with a parameterization of $K = 2$. On this map, we can easily observe that the quality of the accuracy is strongly correlated to the number of available measurements. We can observe a high accuracy from 200 measures ($NM = 200$).

Once we validated signal estimation on signal measurements without obstacles, we choose to test our approach on maps containing antenna signal disturbances.

4.2.2. Signal Map with Obstacles

On the map shown in Figure 11, we also observe that the KNN algorithm performs well for estimating signals. We also notice that the amount of measurement data also conditions the accuracy of the predicted signals. This aspect is even more critical because some interfered areas might not be detected without a sufficiently large number of measurements. This aspect would lead to wrong signal quality predictions. For example, we can see in the figure that signal perturbations are not correctly estimated in the upper right corner of the map ($NM = 10$ to $NM = 200$). These perturbations are only estimated from a considerable number of measurements ($NM = 500$ and $NM = 1000$).

The different tests presented above confirm the high adaptability of the KNN algorithm to the dataset, and the obstacles are well taken into account by the algorithm. We now have a reliable estimation function to predict the signal received at a specific point in the deployment zone. This function will be used as an evaluation function during the execution of the optimization process by the I-HBIA algorithm.

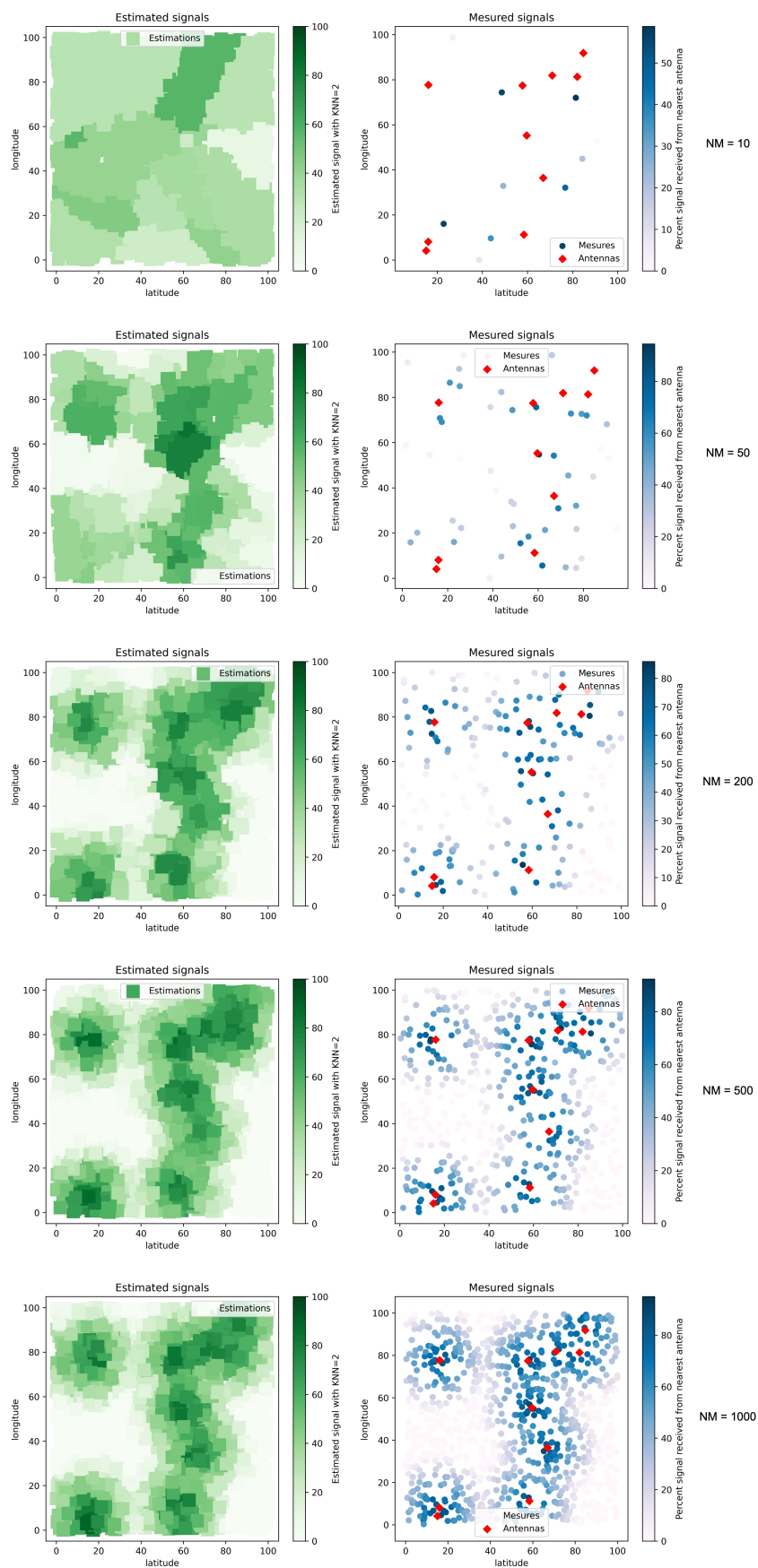


Figure 10. Signal estimations with different numbers of signal measures (NM).

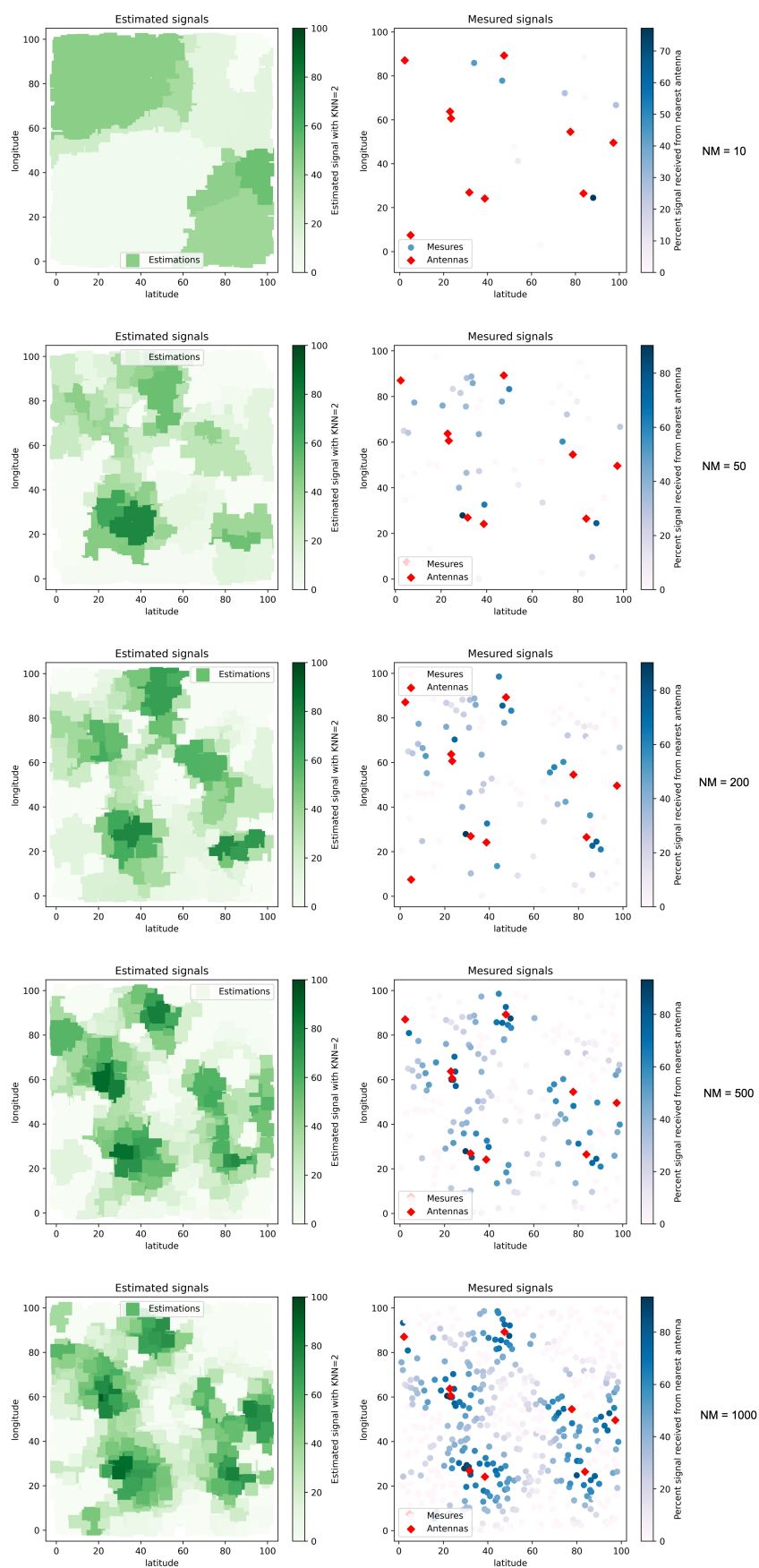


Figure 11. Signal estimations with different numbers of signal (NM) measures and obstacles.

4.2.3. Influence of the Number of Measurements on the Quality of Predictions

We run different estimations on a dataset with various maps to accurately estimate the number of “real signal quality measurements” needed to perform accurate “signal quality estimations”.

Each configuration (number of measurements) was tested on 100 different signal maps. The results shown on the curve in Figure 12 logically confirm the importance of a large number of real signal data for the accuracy of the estimated signals.

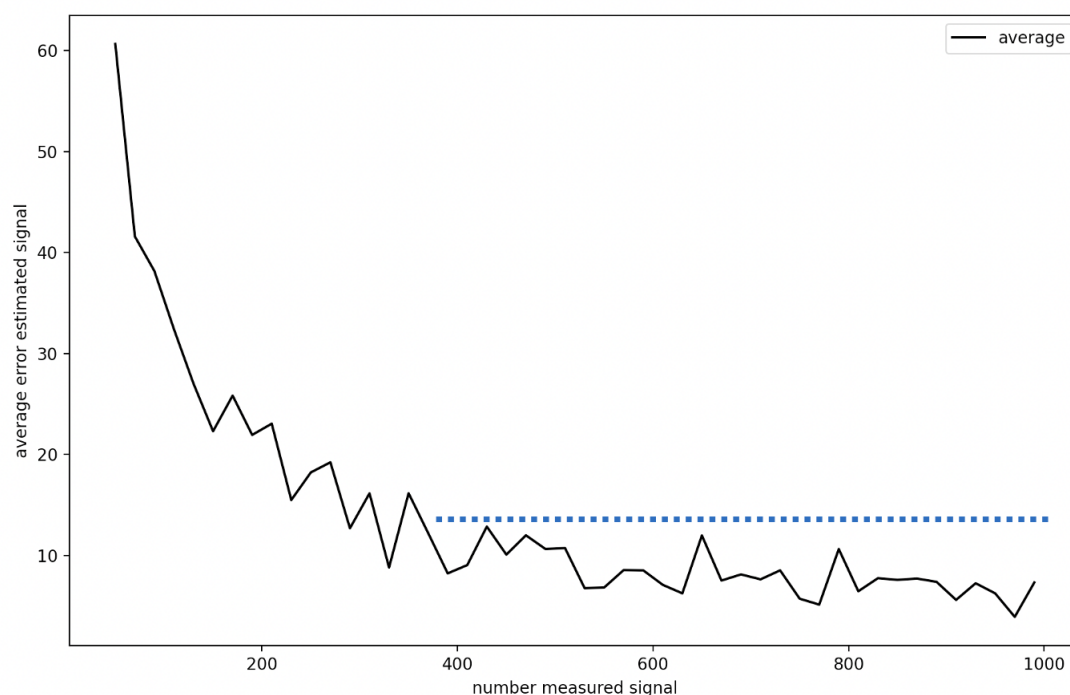


Figure 12. Influence of the number of measured signals on the estimated signal accuracy.

We can conclude that 400 measured signals provide an accurate signal estimation for the KNN algorithm. However, we must balance this aspect. Prediction precision can be influenced positively and negatively by the homogeneity of the measured signals. In our maps, signals are measured homogeneously, and there are examples of deployment where this is impossible.

Before optimizing the nodes’ positions, we chose to verify the performance of the I-HBIA algorithm on classical benchmark functions. We also compared the performance of our version of the HBIA algorithm (I-HBIA) to the classical version (HBIA).

4.3. I-HBIA Algorithm Performances

Each optimization process was repeated ten times, and four configurations were tested. These are described in the Table 4.

We evaluated the performances of these two algorithms on unimodal benchmark functions, presented in Table 5, and multimodal benchmark functions, shown in Table 6. The evaluation space of the unimodal and multimodal functions with two dimensions can be seen in the 3D graphs in Figure 13.

The evolution of the best solution on unimodal and multimodal functions is presented in Figures 14 and 15. These figures show that the four configurations converge towards the global minimum of the functions. We can observe from these average results that the I-HBIA algorithm converges systematically faster towards the global minimum. The hybrid aspect of this algorithm can explain this and the settlement principle presented previously. This improvement in performances is particularly significant for the following functions:

U_3 , U_4 , M_2 , and M_3 . The results between the different methods are close on other functions, but the I-HBIA algorithm remains superior (U_1 , U_2 , U_5 , U_6 , M_1 , M_5 , M_6). The function M_4 is the only function for which the results of the classical version of the algorithm are better.

However, these results must be nuanced because this algorithm version requires more calls to the evaluation function at each iteration due to cloning the best solution step. Nevertheless, the intensification of the search allows a more refined precision of the best solution obtained, which will not be negligible for a WSN deployment problem for which a slight variation in the quality of the connectivity can have consequences on the whole network.

Table 4. Used configurations.

| Configuration | Local Search by Cloning | Dimension Consideration | Number of Iterations |
|-----------------|-------------------------|-------------------------|----------------------|
| HBIA (all) | Disabled | All dimensions | 200 |
| HBIA (random) | Disabled | Random dimensions | 200 |
| I-HBIA (all) | Enabled | All dimensions | 200 |
| I-HBIA (random) | Enabled | Random dimensions | 200 |

Table 5. Unimodal benchmark functions used.

| Equation | Name | Dimension | Number of Iterations | Benchmark ID |
|---|---------------|-----------|----------------------|--------------|
| $\sum_{i=1}^n x_i^4 + \text{random}[0, 1]$ | Quartic | 50 | 200 | U_1 |
| $\sum_{i=1}^n x_i^2$ | Sphere | 50 | 200 | U_2 |
| $\sum_{i=1}^n (\sum_{j=1}^n x_j)^2$ | Schwefel 1.22 | 50 | 200 | U_3 |
| $\max_{i=1..n} x_i $ | Schwefel 2.21 | 50 | 200 | U_4 |
| $\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $ | Schwefel 2.22 | 50 | 200 | U_5 |
| $\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | Rosenbrock | 50 | 200 | U_6 |

Table 6. Multimodal benchmark functions used.

| Equation | Name | Dimension | Iterations | Benchmark ID |
|--|-----------------|-----------|------------|--------------|
| $20 + e - 20 \exp \left[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n-1} x_i^2} \right] - \exp \left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right]$ | Ackley | 50 | 200 | M_1 |
| $\sum_{i=1}^n x_i \sin x_i + 0.1 x_i $ | Alpine | 50 | 200 | M_2 |
| $\sum_{i=1}^n \sin(x_i) [\sin(\frac{jx_i^2}{\pi})]^{2m}, m = 10$ | Michalewicz | 50 | 200 | M_3 |
| $1 - \cos \left(2\pi \sqrt{\sum_{i=1}^n x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$ | Salomon | 50 | 200 | M_4 |
| $10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$ | Rastrigin | 50 | 200 | M_5 |
| $\frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$ | Styblinski Tank | 50 | 200 | M_6 |

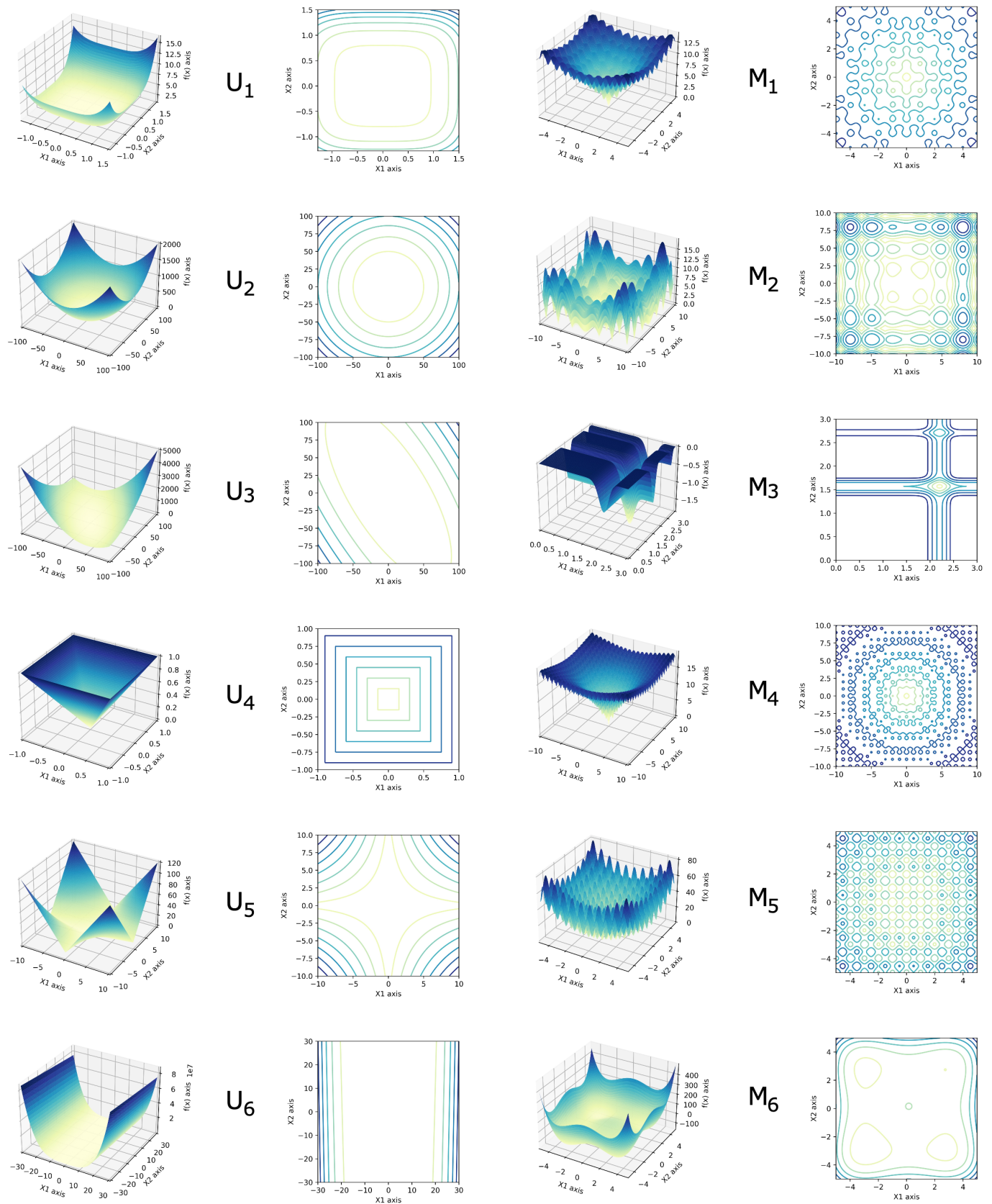


Figure 13. Three-dimensional plot of two dimension function representation.

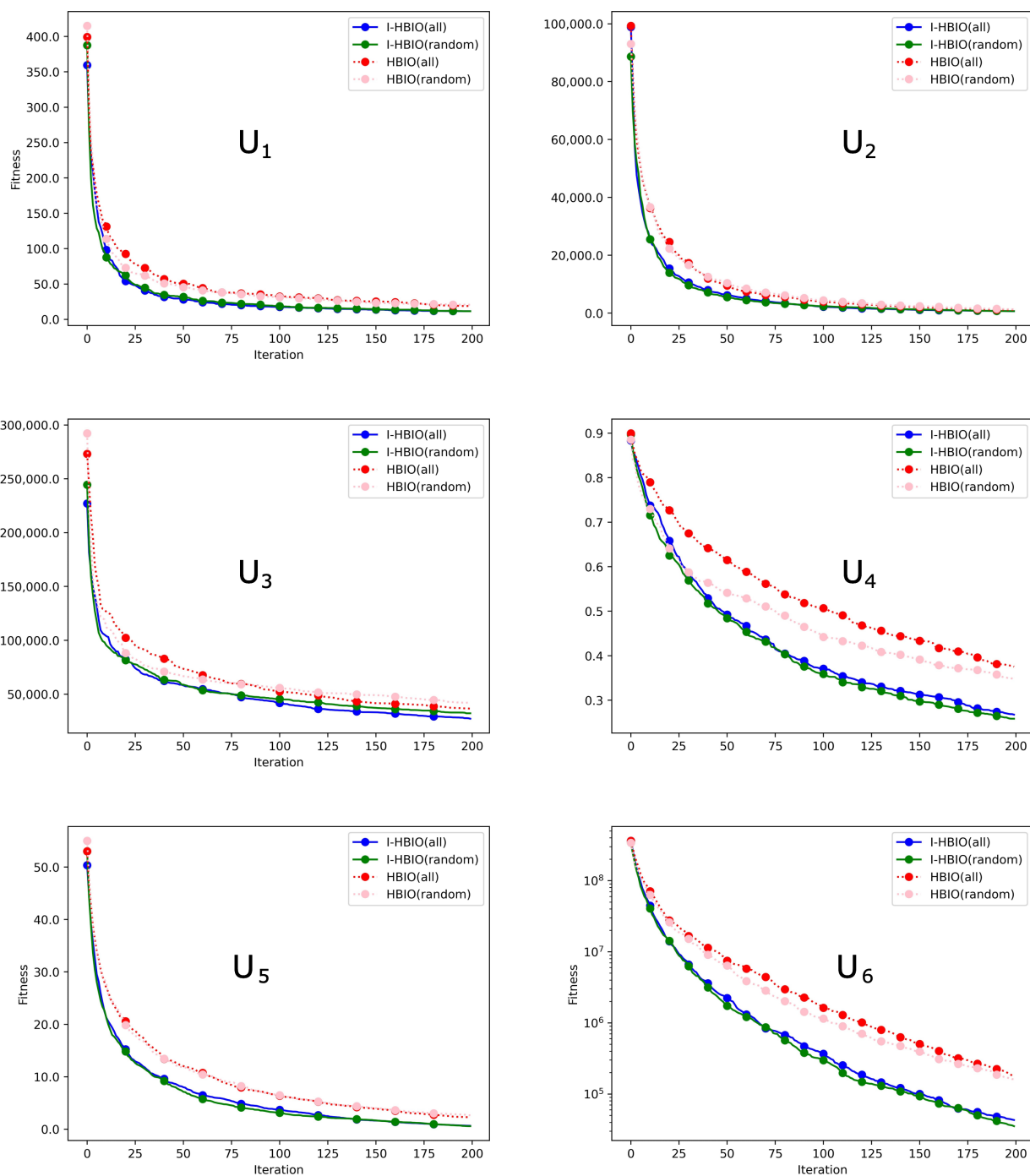


Figure 14. Optimization results on unimodal benchmark functions.

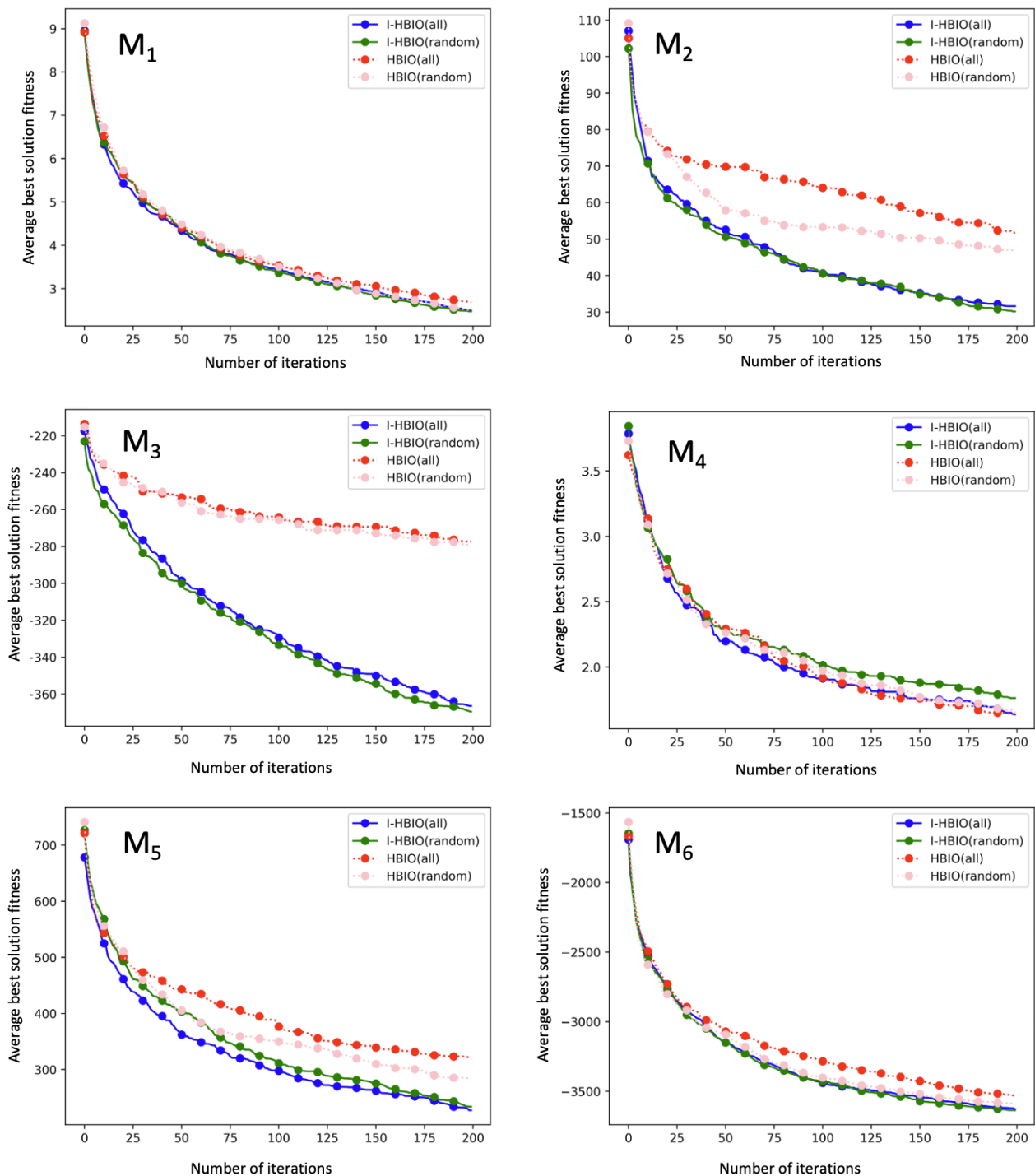


Figure 15. Optimization results on multimodal benchmark functions.

When the I-HBIA algorithm's effectiveness was validated on different benchmarks, we used the algorithm on deployment problems.

4.4. KNN and I-HBIA for WSN Deployment

Following the validation of the signal estimation by the KNN algorithm and the proof of the I-HBIA optimization method, we tested our approach on the different optimization scenarios described in the Table 7.

Table 7. Used configurations.

| Number of Measured Signals | Signal Perturbations | Number of Areas of Interest | Number of Iterations | Figure Example |
|----------------------------|----------------------|-----------------------------|----------------------|----------------|
| 1000 | no | 1 | 50 | Figure 16a |
| 1000 | no | 3 | 50 | Figure 16b |
| 1000 | no | 5 | 50 | Figure 16c |
| 1000 | no | 10 | 50 | Figure 16d |
| 1000 | yes | 1 | 50 | Figure 17a |
| 1000 | yes | 3 | 50 | Figure 17b |
| 1000 | yes | 5 | 50 | Figure 17c |
| 1000 | yes | 10 | 50 | Figure 17d |

We can observe three phenomena on the deployment examples proposed by the tool and presented in Figures 16 and 17. In the first case, the zone of interest does not contain any antenna. In this case, the sensor will try to get as close as possible to a boundary of its zone on which the estimated signal will be the highest. In the second case, the area of interest contains only one single antenna. In this case, the sensor will place itself as expected, close to the antenna, where the estimated signal will be the highest. Finally, in the third case, the area contains several antennas. In this case, the sensor will move closer to the antenna where the estimated received signal will be maximum, provided that no obstacle is present on any of the antennas in the area. If an obstacle is present and generates signal disturbances, the sensor will find the position where it receives the strongest signal from one of the antennas in the area.

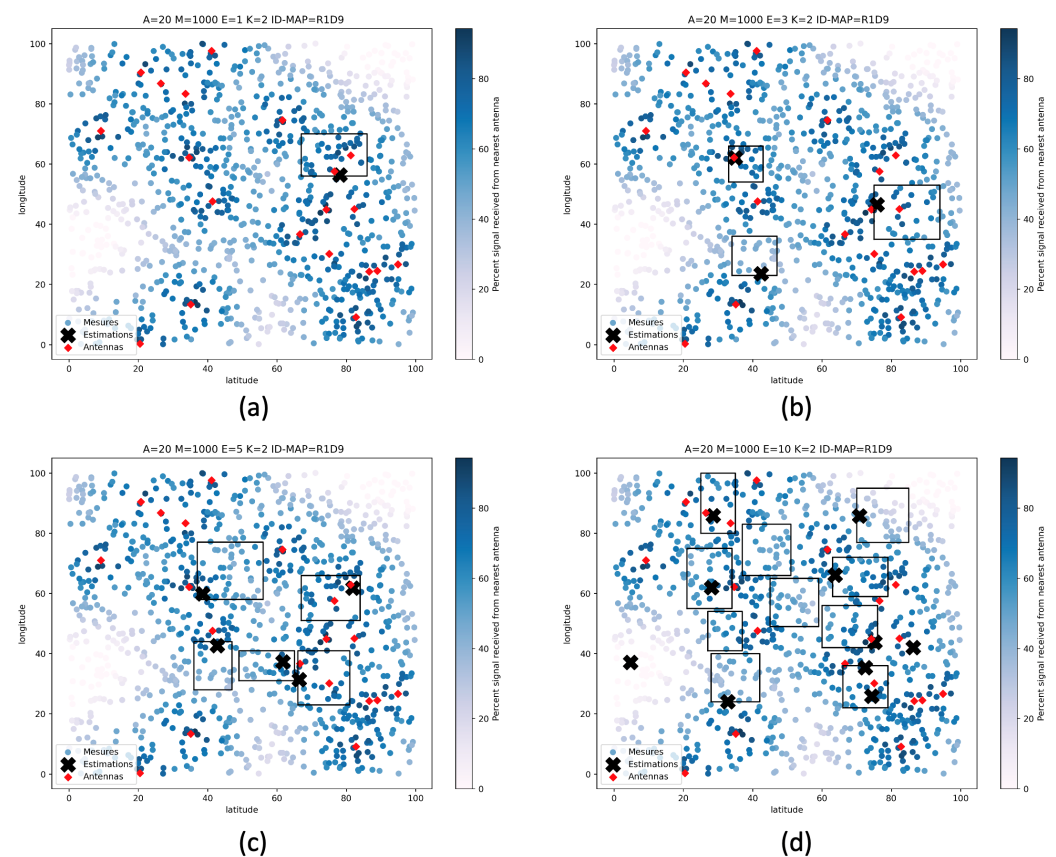


Figure 16. Proposed deployments after 50 iterations on 1, 3, 5, and 10 areas of interest without obstacles (a) 1 area of interest, (b) 3 areas of interest, (c) 5 areas of interest, (d) 10 areas of interest.

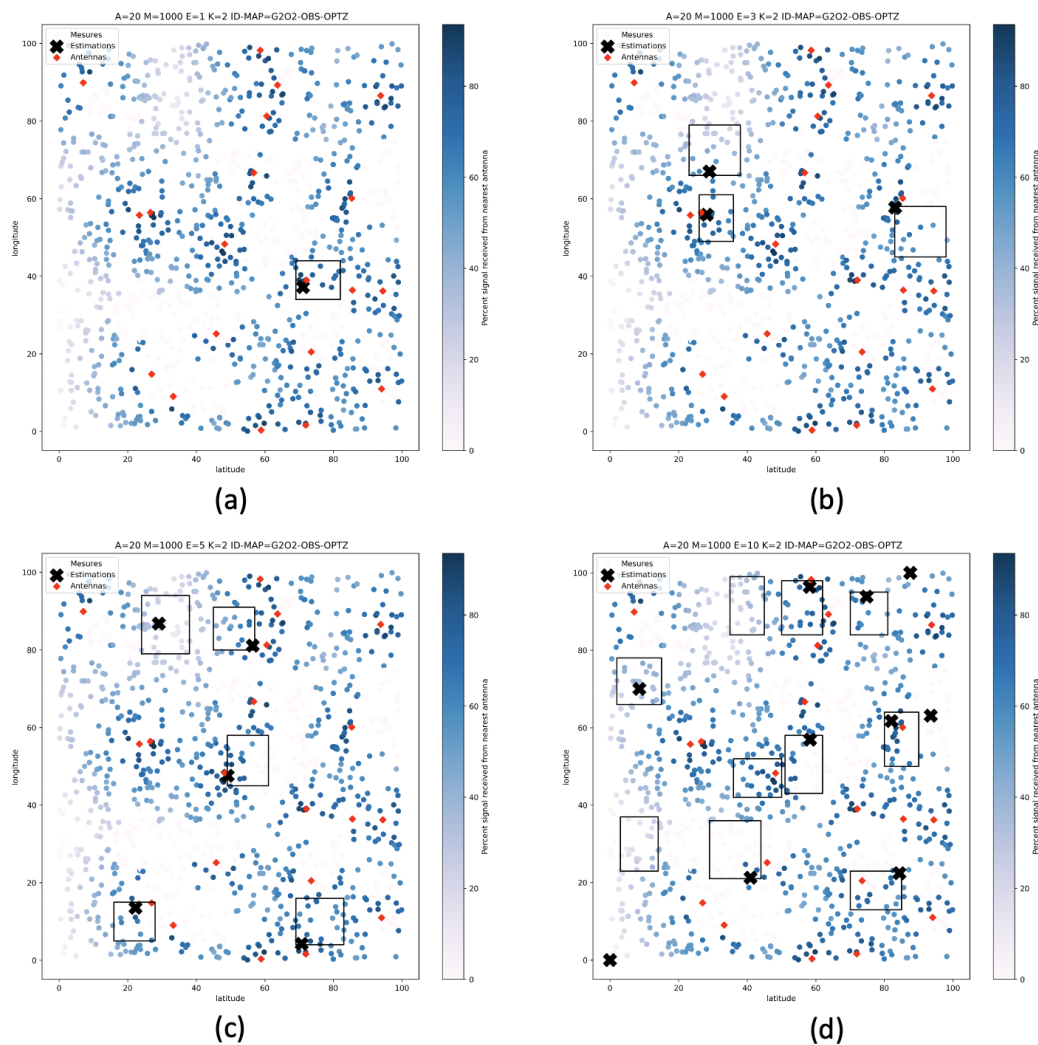


Figure 17. Proposed deployments after 200 iterations on 1, 3, 5, and 10 areas of interest with many obstacles (a) 1 area of interest, (b) 3 areas of interest, (c) 5 areas of interest, (d) 10 areas of interest.

However, as we can see in Figures 16d and 17d, the deployment optimization does not consistently achieve the optimum. Indeed, in these figures, we can see that a zone does not contain any sensor while another contains two. This phenomenon is characteristic of metaheuristics. In this case, the optimization has made good progress but has finally been blocked in a local minimum. Generally, the repetition of optimization processes brings a solution to this kind of problem, even with a smaller number of iterations. This problem is no longer visible when the optimization is performed on a more significant number of iterations on the same number of zones.

As shown in Figure 18, the method also works on many dimensions. On this map, we optimize a deployment on 25 zones. The problem to solve is dimension 50 (latitude and longitude of each node). We can observe from this figure that only one zone of the 25 defined does not contain any sensor. On the areas covered by a sensor, we can see that each sensor chooses the position at which its signal estimation is maximum.

Nevertheless, we can observe from this figure that a few sensors are outside areas of interest. Three factors can explain this. The first one concerns the coverage of the zone by signals. Indeed, if any signal does not cover this zone, the optimizer will not place any sensor in this zone. The second factor is the size of the area. A small area will be problematic for the optimizer, and the guidance towards this area by the evaluation

function will be weak. The third factor is the number of iterations used for this execution. Indeed, 500 iterations for a complex problem of dimension 50 have a high probability not of leading to an optimum but getting very close to it.

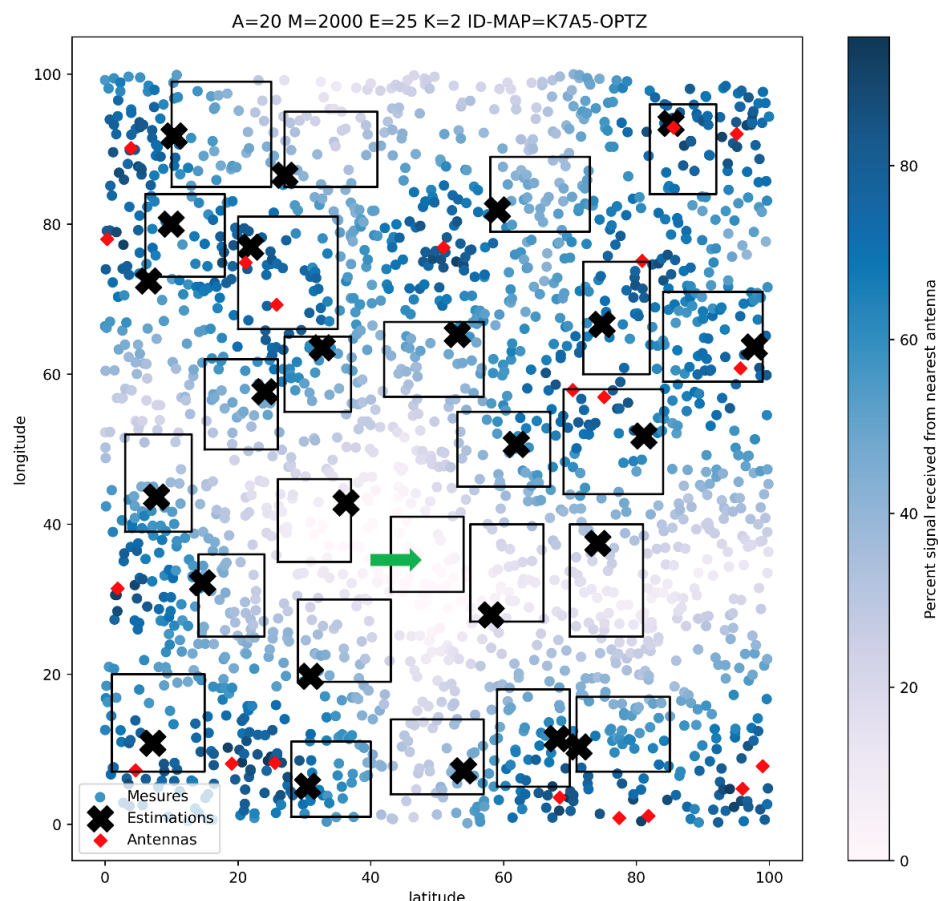


Figure 18. An example of proposed deployment for 25 areas of interest after 500 iterations.

5. Conclusions

Our approach provides a new method to optimize the deployment of WSN in signal estimation functions. The core of this approach is based on two algorithms. The KNN algorithm accurately estimates the signals on the deployment area from localized points. The I-HBIA algorithm optimizes the nodes' positioning according to the estimated signals and the defined area of interest.

This work opens the way to different research tracks. The first one concerns the accuracy of signal measurements. Currently, our method works on simulated signal collections on other test datasets. In the future, it would be interesting to integrate signal measurement data collected by drones or technicians traveling to the deployment areas.

During our simulations, we realized that the computation time remained important despite using metaheuristics. Indeed, the geometrical operations necessary to evaluate the solutions and search for the neighbors for the estimations require considerable computation time. Solutions might include discretizing the search space while maintaining an acceptable accuracy level and continue improving the I-HBIA method by creating a parallel version of I-HBIA.

During some optimization processes, I-HBIA does not find the best node positions. This aspect increases with the number of deployed sensors and can be explained by the curse of dimensionality. To address this limitation, we will integrate reduction methods into our approach.

Finally, we will focus on weather impact on the LoRa signal quality. Indeed, during our experimentations, we observed some variations in signal quality measures, and we assume this variation was generated by environmental factors (rain, wind, or snow). This aspect can be hazardous for critical applications such as animal tracking or water level thresholds monitoring.

Author Contributions: Conceptualization, B.P. and T.A.-S.; methodology, B.P.; software, B.P.; validation, B.P. and C.B.; writing review and editing, B.P. and E.V.; supervision, T.A.-S.; project administration, T.A.-S.; funding acquisition, T.A.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by SMART PAESI PROJECT and collectivity of Corsica.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Idrees, Z.; Zheng, L. Low cost air pollution monitoring systems: A review of protocols and enabling technologies. *J. Ind. Inf. Integr.* **2020**, *17*, 100123. [\[CrossRef\]](#)
2. Salman, N.; Kemp, A.H.; Khan, A.; Noakes, C. Real Time Wireless Sensor Network (WSN) based Indoor Air Quality Monitoring System. *IFAC-PapersOnLine* **2019**, *52*, 324–327. [\[CrossRef\]](#)
3. Rehman, A.U.; Abbasi, A.Z.; Islam, N.; Shaikh, Z.A. A review of wireless sensors and networks' applications in agriculture. *Comput. Stand. Interfaces* **2014**, *36*, 263–270. [\[CrossRef\]](#)
4. Zhang, Y.; Sun, L.; Song, H.; Cao, X. Ubiquitous WSN for Healthcare: Recent Advances and Future Prospects. *IEEE Internet Things J.* **2014**, *1*, 311–318. [\[CrossRef\]](#)
5. Moorthy, H.R.; Bangera, V.; Amrin, Z.; Avinash, N.; Rao, N.S.K. WSN in Defence Field: A Security Overview. In Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 7–9 October 2020; pp. 258–264. [\[CrossRef\]](#)
6. Asorey-Cacheda, R.; Garcia-Sanchez, A.J.; Garcia-Sanchez, F.; Garcia-Haro, J. A survey on non-linear optimization problems in wireless sensor networks. *J. Netw. Comput. Appl.* **2017**, *82*, 1–20. [\[CrossRef\]](#)
7. Farsi, M.; Elhosseini, M.A.; Badawy, M.; Arafat Ali, H.; Zain Eldin, H. Deployment Techniques in Wireless Sensor Networks, Coverage and Connectivity: A Survey. *IEEE Access* **2019**, *7*, 28940–28954. [\[CrossRef\]](#)
8. Adnan, T.; Datta, S.; MacLean, S. Efficient and Accurate Range-based Sensor Network Localization. *Procedia Comput. Sci.* **2012**, *10*, 405–413. [\[CrossRef\]](#)
9. Luomala, J.; Hakala, I. Adaptive range-based localization algorithm based on trilateration and reference node selection for outdoor wireless sensor networks. *Comput. Netw.* **2022**, *210*, 108865. [\[CrossRef\]](#)
10. Pandey, S.; Varma, S. A Range Based Localization System in Multihop Wireless Sensor Networks: A Distributed Cooperative Approach. *Wirel. Pers. Commun.* **2016**, *86*, 615–634. [\[CrossRef\]](#)
11. Khan, H.; Hayat, M.N.; Ur Rehman, Z. Wireless sensor networks free-range base localization schemes: A comprehensive survey. In Proceedings of the 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan, 8–9 March 2017; pp. 144–147. [\[CrossRef\]](#)
12. Singh, S.P.; Sharma, S.C. Range Free Localization Techniques in Wireless Sensor Networks: A Review. *Procedia Comput. Sci.* **2015**, *57*, 7–16. [\[CrossRef\]](#)
13. Jin, Y.; Zhou, L.; Zhang, L.; Hu, Z.; Han, J. A Novel Range-Free Node Localization Method for Wireless Sensor Networks. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 688–692. [\[CrossRef\]](#)
14. Cheikhrouhou, O.; Bhatti, G.M.; Alroobaea, R. A Hybrid DV-Hop Algorithm Using RSSI for Localization in Large-Scale Wireless Sensor Networks. *Sensors* **2018**, *18*, 1469. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Gui, L.; Xiao, F.; Zhou, Y.; Shu, F.; Val, T. Connectivity Based DV-Hop Localization for Internet of Things. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8949–8958. [\[CrossRef\]](#)
16. Dasgupta, S.; Mao, G.; Anderson, B. A New Measure of Wireless Network Connectivity. *IEEE Trans. Mob. Comput.* **2015**, *14*, 1765–1779. [\[CrossRef\]](#)
17. Zhu, C.; Zheng, C.; Shu, L.; Han, G. A survey on coverage and connectivity issues in wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 619–632. [\[CrossRef\]](#)
18. Liu, Y.; Suo, L.; Sun, D.; Wang, A. A Virtual Square Grid-Based Coverage Algorithm of Redundant Node for Wireless Sensor Network. *J. Netw. Comput. Appl.* **2013**, *36*, 811–817. [\[CrossRef\]](#)

19. Fadi, M.A.T.; Hossam, S.H.; Mohamed, A.I. Quantifying connectivity of grid-based Wireless Sensor Networks under practical errors. In Proceedings of the IEEE Local Computer Network Conference, Denver, CO, USA, 10–14 October 2010; pp. 220–223. [\[CrossRef\]](#)
20. Yu, X.; Liu, N.; Huang, W.; Qian, X.; Zhang, T. A Node Deployment Algorithm Based on Van Der Waals Force in Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 505710. [\[CrossRef\]](#)
21. Mahboubi, H.; Aghdam, A.G. Distributed Deployment Algorithms for Coverage Improvement in a Network of Wireless Mobile Sensors: Relocation by Virtual Force. *IEEE Trans. Control Netw. Syst.* **2017**, *4*, 736–748. [\[CrossRef\]](#)
22. Wang, S.Y.; Shih, K.P.; Chen, Y.D.; Ku, H.H. Preserving Target Area Coverage in Wireless Sensor Networks by Using Computational Geometry. In Proceedings of the 2010 IEEE Wireless Communication and Networking Conference, Sydney, Australia, 18–21 April 2010; pp. 1–6. [\[CrossRef\]](#)
23. Wu, C.H.; Lee, K.C.; Chung, Y.C. A Delaunay Triangulation based method for wireless sensor network deployment. *Comput. Commun.* **2007**, *30*, 2744–2752. [\[CrossRef\]](#)
24. Wang, B. Coverage Problems in Sensor Networks: A Survey. *ACM Comput. Surv.* **2011**, *43*, 1–53. [\[CrossRef\]](#)
25. Deif, D.S.; Gadallah, Y. Classification of Wireless Sensor Networks Deployment Techniques. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 834–855. [\[CrossRef\]](#)
26. Naik, C.; Shetty, D.P. A novel meta-heuristic differential evolution algorithm for optimal target coverage in wireless sensor networks. In *International Conference on Innovations in Bio-Inspired Computing and Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 83–92.
27. Chand, S.; Kumar, B. Genetic algorithm-based meta-heuristic for target coverage problem. *IET Wirel. Sens. Syst.* **2018**, *8*, 170–175.
28. Singh, S.; Kumar, S.; Nayyar, A.; Al-Turjman, F.; Mostarda, L. Proficient QoS-based target coverage problem in wireless sensor networks. *IEEE Access* **2020**, *8*, 74315–74325.
29. Karatas, M. Optimal deployment of heterogeneous sensor networks for a hybrid point and barrier coverage application. *Comput. Netw.* **2018**, *132*, 129–144. [\[CrossRef\]](#)
30. Mishra, P.; Kumar, N.; Godfrey, W.W. A Meta-heuristic-based Green-routing Algorithm in Software-Defined Wireless Sensor Network. In Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 36–41. [\[CrossRef\]](#)
31. El Ghazi, A.; Ahiod, B. Random waypoint impact on bio-inspired routing protocols in WSN. In Proceedings of the 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet, Tunisia, 18–20 December 2016; pp. 326–331. [\[CrossRef\]](#)
32. Gui, T.; Ma, C.; Wang, F.; Wilkins, D.E. Survey on swarm intelligence based routing protocols for wireless sensor networks: An extensive study. In Proceedings of the 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, 14–17 March 2016; pp. 1944–1949. [\[CrossRef\]](#)
33. Zhao, X.; Ren, S.; Quan, H.; Gao, Q. Routing Protocol for Heterogeneous Wireless Sensor Networks Based on a Modified Grey Wolf Optimizer. *Sensors* **2020**, *20*, 820. [\[CrossRef\]](#)
34. Ayedi, M.; Eldesouky, E.; Nazeer, J. Energy-Spectral Efficiency Optimization in Wireless Underground Sensor Networks Using Salp Swarm Algorithm. *J. Sens.* **2021**, *2021*, 6683988. [\[CrossRef\]](#)
35. Esmaeili, H.; Bidgoli, B.M.; Hakami, V. CMML: Combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks. *Appl. Soft Comput.* **2022**, *118*, 108477. [\[CrossRef\]](#)
36. Tsai, C.W.; Hong, T.P.; Shiu, G.N. Metaheuristics for the Lifetime of WSN: A Review. *IEEE Sens. J.* **2016**, *16*, 2812–2831. [\[CrossRef\]](#)
37. Gambhir, A.; Payal, A.; Arya, R. Chicken Swarm Optimization Algorithm Perspective on Energy Constraints in WSN. In Proceedings of the 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 27–29 November 2020; pp. 1–5. [\[CrossRef\]](#)
38. Wang, H.; Li, K.; Pedrycz, W. An Elite Hybrid Metaheuristic Optimization Algorithm for Maximizing Wireless Sensor Networks Lifetime with a Sink Node. *IEEE Sens. J.* **2020**, *20*, 5634–5649. [\[CrossRef\]](#)
39. Abba Ari, A.A.; Gueroui, A.; Yenke, B.O.; Labraoui, N. Energy efficient clustering algorithm for Wireless Sensor Networks using the ABC metaheuristic. In Proceedings of the 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 7–9 January 2016; pp. 1–6. [\[CrossRef\]](#)
40. Chung, V.; Tuah, N.; Lim, K.G.; Tan, M.K.; Saad, I.; Kin Teo, K.T. Metaheuristic Multi-Hop Clustering Optimization for Energy-Efficient Wireless Sensor Network. In Proceedings of the 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAET), Kota Kinabalu, Malaysia, 26–27 September 2020; pp. 1–6. [\[CrossRef\]](#)
41. Zivkovic, M.; Bacanin, N.; Zivkovic, T.; Strumberger, I.; Tuba, E.; Tuba, M. Enhanced Grey Wolf Algorithm for Energy Efficient Wireless Sensor Networks. In Proceedings of the 2020 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2020; pp. 87–92. [\[CrossRef\]](#)
42. Mohamed, A.; Saber, W.; Elnahry, I.; Hassanien, A.E. Coyote Optimization Based on a Fuzzy Logic Algorithm for Energy-Efficiency in Wireless Sensor Networks. *IEEE Access* **2020**, *8*, 185816–185829. [\[CrossRef\]](#)
43. Tuba, E.; Simian, D.; Dolicanin, E.; Jovanovic, R.; Tuba, M. Energy Efficient Sink Placement in Wireless Sensor Networks by Brain Storm Optimization Algorithm. In Proceedings of the 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 718–723. [\[CrossRef\]](#)

44. Shankar, T.; Eappen, G.; Sahani, S.; Rajesh, A.; Mageshvaran, R. Integrated Cuckoo and Monkey Search Algorithm for Energy Efficient Clustering in Wireless Sensor Networks. In Proceedings of the 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 22–23 March 2019; Volume 1, pp. 1–4. [\[CrossRef\]](#)
45. Zivkovic, M.; Bacanin, N.; Tuba, E.; Strumberger, I.; Bezdan, T.; Tuba, M. Wireless Sensor Networks Life Time Optimization Based on the Improved Firefly Algorithm. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1176–1181. [\[CrossRef\]](#)
46. Tsai, C.W.; Tsai, P.W.; Pan, J.S.; Chao, H.C. Metaheuristics for the deployment problem of WSN: A review. *Microprocess. Microsyst.* **2015**, *39*, 1305–1317. [10.1016/j.micpro.2015.07.003](#). [\[CrossRef\]](#)
47. Liao, W.H.; Kao, Y.; Li, Y.S. A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. *Expert Syst. Appl.* **2011**, *38*, 12180–12188. [\[CrossRef\]](#)
48. Metiaf, A.; Wu, Q. Particle Swarm Optimization Based Deployment for WSN with the Existence of Obstacles. In Proceedings of the 2019 5th International Conference on Control, Automation and Robotics (ICCAR), Beijing, China, 19–22 April 2019.
49. Kumar, G.; Ranga, V. Meta-heuristics for relay node placement problem in wireless sensor networks. In Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Wanknaghat, India, 22–24 December 2016; pp. 375–380. [\[CrossRef\]](#)
50. Ng, C.K.; Wu, C.H.; Ip, W.H.; Yung, K.L. A Smart Bat Algorithm for Wireless Sensor Network Deployment in 3-D Environment. *IEEE Commun. Lett.* **2018**, *22*, 2120–2123. [\[CrossRef\]](#)
51. Arsic, A.; Tuba, M.; Jordanski, M. Fireworks algorithm applied to wireless sensor networks localization problem. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 4038–4044. [\[CrossRef\]](#)
52. Das, P.P.; Chakraborty, N.; Allayear, S.M. Optimal coverage of Wireless Sensor Network using Termite Colony Optimization Algorithm. In Proceedings of the 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 21–23 May 2015; pp. 1–6. [\[CrossRef\]](#)
53. Ghosh, S.; Snigdh, I.; Singh, A. GA optimal sink placement for maximizing coverage in Wireless Sensor Networks. In Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 23–25 March 2016; pp. 737–741. [\[CrossRef\]](#)
54. Aziz, N.A.A.; Aziz, N.H.A.; Aziz, K.A.; Ibrahim, Z.; Aliman, M.N. Evaluation of Pure Gravitational Search Algorithm for Wireless Sensor Networks Coverage Maximization. In Proceedings of the 2018 International Electrical Engineering Congress (iEECON), Krabi, Thailand, 7–9 March 2018; pp. 1–4. [\[CrossRef\]](#)
55. Strumberger, I.; Tuba, E.; Bacanin, N.; Beko, M.; Tuba, M. Wireless Sensor Network Localization Problem by Hybridized Moth Search Algorithm. In Proceedings of the 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 316–321. [\[CrossRef\]](#)
56. Sabbella, V.R.; Edla, D.R.; Lipare, A.; Parne, S.R. An Efficient Localization Approach in Wireless Sensor Networks Using Krill Herd Optimization Algorithm. *IEEE Syst. J.* **2021**, *15*, 2432–2442. [\[CrossRef\]](#)
57. Strumberger, I.; Tuba, E.; Bacanin, N.; Beko, M.; Tuba, M. Monarch butterfly optimization algorithm for localization in wireless sensor networks. In Proceedings of the 2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA), Prague, Czech Republic, 19–20 April 2018; pp. 1–6. [\[CrossRef\]](#)
58. Shindarov, M.; Fidanova, S.; Marinov, P. Wireless sensor positioning algorithm. In Proceedings of the 2012 6th IEEE International Conference Intelligent Systems, Sofia, Bulgaria, 6–8 September 2012; pp. 419–424. [\[CrossRef\]](#)
59. Deif, D.; Gadallah, Y. Wireless Sensor Network deployment using stochastic optimization techniques—A comparative study. In Proceedings of the 2015 International Conference on Computing and Network Communications (CoCoNet), Trivandrum, India, 16–19 December 2015; pp. 131–138. [\[CrossRef\]](#)
60. Alia, O.M.; Al-Ajouri, A. Maximizing Wireless Sensor Network Coverage with Minimum Cost Using Harmony Search Algorithm. *IEEE Sens. J.* **2017**, *17*, 882–896. [\[CrossRef\]](#)
61. Idrees, Z.; Zheng, L. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *17*, 67–82.
62. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
63. Holland, J.H.; Reitman, J.S. Cognitive systems based on adaptive algorithms. *SIGART Newsl.* **1977**, *63*, 49. [\[CrossRef\]](#)
64. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [\[CrossRef\]](#)
65. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [\[CrossRef\]](#)
66. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
67. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
68. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [\[CrossRef\]](#)
69. Abedinpourshotorban, H.; Mariyam Shamsuddin, S.; Beheshti, Z.; Jawawi, D.N.A. Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm Evol. Comput.* **2016**, *26*, 8–22. [\[CrossRef\]](#)
70. Azizi, M. Atomic orbital search: A novel metaheuristic algorithm. *Appl. Math. Model.* **2021**, *93*, 657–683. [\[CrossRef\]](#)
71. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [\[CrossRef\]](#)

72. Ghasemian, H.; Ghasemian, F.; Vahdat-Nejad, H. Human urbanization algorithm: A novel metaheuristic approach. *Math. Comput. Simul.* **2020**, *178*, 1–15. [[CrossRef](#)]
73. Mousavirad, S.J.; Ebrahimpour-Komleh, H. Human mental search: A new population-based metaheuristic optimization algorithm. *Appl. Intell.* **2017**, *47*, 850–887. [[CrossRef](#)]
74. Salih, S.Q.; Alsewari, A.A. A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer. *Neural Comput. Appl.* **2020**, *32*, 10359–10386. [[CrossRef](#)]
75. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]
76. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [[CrossRef](#)]
77. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
78. Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 160. [[CrossRef](#)] [[PubMed](#)]
79. Ray, S. A quick review of machine learning algorithms. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 35–39.
80. Arrieta, A.B.; Díaz-Rodríguez, N.; Ser, J.D.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [[CrossRef](#)]
81. Yang, L.; Shami, A. On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *arXiv* **2020**, arXiv:2007.15745.
82. Fix, E.; Hodges, J.L. Discriminatory Analysis—Nonparametric Discrimination: Consistency Properties. *Int. Stat. Rev.* **1989**, *57*, 238. [[CrossRef](#)]
83. Zhang, S.; Cheng, D.; Deng, Z.; Zong, M.; Deng, X. A novel kNN algorithm with data-driven k parameter computation. *Pattern Recognit. Lett.* **2018**, *109*, 44–54. [[CrossRef](#)]
84. Morais, R.G.; Nedjah, N.; Mourelle, L.M. A novel metaheuristic inspired by Hitchcock birds' behavior for efficient optimization of large search spaces of high dimensionality. *Soft Comput.* **2020**, *24*, 5633–5655. [[CrossRef](#)]