applied sciences

MDPI

Article

A Novel Approach to Sensor Placement: Recursive Exhaustive Search and Stochastic Optimization Parameter Impact Analysis

Marina Banov [1,2], Domagoj Pinčić [1], Kristijan Lenac [1,2] and Diego Sušanj [1,*]

[1] Faculty of Engineering, University of Rijeka, Vukovarska 58, 51000 Rijeka, Croatia; marina.banov@uniri.hr (M.B.); pincic.domagoj@gmail.com (D.P.); kristijan.lenac@uniri.hr (K.L.)
[2] Center for Artificial Intelligence and Cybersecurity, University of Rijeka, R. Matejčić 2, 51000 Rijeka, Croatia
* Correspondence: diego.susanj@uniri.hr

Abstract: This study presents a comprehensive approach for single sensor placement optimization in two-dimensional and three-dimensional spaces. A traditional exhaustive search technique and a novel method called recursive exhaustive search are used to place a sensor in a way that maximizes the area coverage metric. Exhaustive search provides a baseline by methodically evaluating all potential placements, while recursive exhaustive search innovates by segmenting the search process into more manageable, recursive steps. Our findings highlight the significant impact of two key parameters, the number of evaluations and the rasterization value, on the achieved coverage and computation time. The results show that the right choice of parameters can significantly reduce the computational effort without compromising the quality of the solution. This underlines the critical need for a balanced approach that considers both computational complexity and placement efficacy. We show that exhaustive search is not feasible for three-dimensional environment models and propose to establish a modified exhaustive search method as a ground truth for the single sensor placement problem. We then explore nature-inspired genetic algorithms and the impact of the number of evaluations of the optimization function for these algorithms on both accuracy and computational cost.

Keywords: sensor placement; stochastic optimization; exhaustive search; sensor networks; Internet of Things

Citation: Banov, M.; Pinčić, D.; Lenac, K.; Sušanj, D. A Novel Approach to Sensor Placement: Recursive Exhaustive Search and Stochastic Optimization Parameter Impact Analysis. *Appl. Sci.* **2024**, *14*, 2786. https://doi.org/10.3390/app14072786

Academic Editor: Chilukuri K. Mohan

Received: 19 February 2024
Revised: 22 March 2024
Accepted: 23 March 2024
Published: 26 March 2024

1. Introduction

As the Internet of Things (IoT) continues to expand and sensor networks become more prevalent in various applications, the task of optimizing sensor utilization within specific environments with regard to certain key performance indicators is becoming increasingly important. One of the most important performance metrics is network coverage [1]. The overall coverage performance of the sensor network depends on the placement of each sensor. Depending on the sensor type, each sensor can have different individual coverage models.

The placement of isotropic beacons plays a pivotal role in the efficiency of indoor localization systems [2], which could lead to many applications, e.g., indoor navigation of drones [3]. Isotropic beacons emit signals uniformly in all directions, providing a consistent and reliable source of spatial information. This uniformity ensures that drones can determine their position within an indoor environment with high precision, by triangulating their location relative to multiple beacons. Optimal placement of these beacons is essential to minimize signal obstructions and interference, which can significantly degrade localization accuracy. Consequently, a well-considered configuration of isotropic beacons enables drones to navigate complex indoor spaces safely and efficiently, by ensuring continuous and accurate positional data. This is particularly important in environments where GPS signals are unavailable or unreliable, highlighting the critical role of isotropic beacon placement in the advancement of indoor drone technology.

An optimal placement of visual sensors can improve the performance of video surveillance systems while minimizing the number of sensors required to cover a given area, reducing the overall cost of installation and operation. It may be important to cover certain areas of the environment, such as pathways, areas around exhibited artwork and similar locations. When analyzing human behavior, extending the coverage of visual or isotropic sensors to specific areas—such as in front of product shelves, advertising displays or showcases—allows for a closer examination of people's interest in specific products or advertising types.

The challenge of determining the optimal positioning of sensors in an environment arises from the Art Gallery problem (AGP) [4]. This problem has its roots in the concept of visibility, a measure that quantifies the effectiveness of sensors in detecting and observing objects in an environment. Visibility appears as an important aspect in various fields such as robotics, computer vision, optimization and computer graphics. Originally, the AGP aimed to find optimal locations for guards that could monitor a full 360° area from fixed positions, typically within a polygonal art gallery. In its original form, the AGP assumed that visibility between two points is given if the line segment connecting them does not intersect any obstacles—a principle known as line-of-sight constraint.

Subsequent research extended the AGP by incorporating distance and incidence constraints [5], but retained the simplified two-dimensional polygonal approach. In these studies, polygonal representations of the floor plan of the environment were used, which often inadequately capture the complicated spatial arrangements of buildings. In many practical scenarios, understanding the three-dimensional spatial structure of a building is crucial for developing effective monitoring strategies [6]. In three-dimensional environments, research usually relies on elevation maps, which are limited to planar terrains [7] or orthogonal polyhedral environments [8].

Both the two-dimensional and the three-dimensional version of the Art Gallery problem have been shown to be NP-hard. However, while solutions can be found for certain instances of the two-dimensional problem and some three-dimensional scenarios with simple polygons [5,9,10], practical implementations are faced with increasing complexity due to the different shapes of real-world polygons. A universally applicable solution that covers all polygon types remains elusive [4].

The optimization of sensor placement involves several key aspects that shape the approach and solution strategies. These aspects encompass the type of coverage used, the dimensionality of the target environment, sensor detection capabilities and sensor models.

The choice of coverage metric has a direct influence on the optimization process. Commonly used metrics include area coverage, point coverage and barrier coverage, each tailored to specific application requirements [11,12]. In this work, we use area coverage, the most widely used metric, which quantifies the ratio of the area covered by the sensors to the total target area [3,13–16].

Based on the AGP mentioned above, it becomes clear that the second aspect of the problem relates to the dimension of the target environment. Environment models, such as voxel-based representations, help to capture the spatial subtleties required for accurate coverage computation. The dimensionality of the target environment model is crucial for accurate sensor placement. While two-dimensional environment models are computationally simple, they can lead to errors due to oversimplified representations. Three-dimensional environment models provide a more accurate representation, but are associated with higher computational costs [17]. In previous research, spaces are often simplified to the level of a floor plan or two-dimensional boundaries [18–26]. As an improvement of the two-dimensional environment models, three-dimensional models were mostly described with elevation maps [27–29]. In our previous research, we proposed three-dimensional environment models and a framework for optimizing sensor placement [15,16]. Throughout this manuscript, both two-dimensional and three-dimensional types of environments will be examined.

The detection capabilities of sensors play an important role in optimizing placement. Sensors can have binary or probabilistic sensing ability. The sensing ability of a sensor refers to whether it can perceive an object in the case of binary coverage [19–26,29–31], or how effectively it can perceive it in the case of probabilistic coverage, which provides a more refined model [15,16,28,32,33]. In addition to detection capability, the shape of the area covered by the sensor is also important. The coverage area of the sensor depends on the sensor type, mainly isotropic or directional [34].

In previous research on sensor placement optimization, various optimization algorithms have been used to improve the efficiency of sensor deployment tailored to specific needs. Early attempts that focused on the placement of binary directional [19] and isotropic sensors [21] in two-dimensional spaces employed techniques such as Integer Linear Programming, Binary Integer Programming, Greedy Search and random arrangement. Advances in optimal placement of binary sensors in two dimensions include methods based on genetic algorithms such as Individual Particle Optimization [13], Particle Swarm Optimization, Binary Genetic Algorithm, Simulated Annealing [27], Improved Cuckoo Search Algorithm and Chaotic Flower Pollination Algorithm [14].

Extending to three-dimensional spaces with height maps, studies compared deterministic methods with the Covariance Matrix Adaptation Evolution Strategy algorithm, the Limited-memory Broyden–Fletcher–Goldfarb–Shanno method and the Gradient Descent [17,28,32]. The Covariance Matrix Adaptation Evolution Strategy showed superior performance for smaller height maps, while Gradient Descent was effective for larger maps.

Some approaches utilized exhaustive search for optimal placement of sensors to monitor deformation and detect damage in structural systems [35–38]. In addition, exhaustive search has been used for the ordered placement of nodes in sensor networks. For example, the placement of sensors to monitor leaks in drinking water networks [39] or to achieve the target coverage of the wireless sensor network based on the selected coverage metric [40]. Most of the current research uses one-dimensional approaches [35,36,38] that usually simulate structural beams. Other researchers use two-dimensional simulated environments to determine the optimal placement of sensors [37,39,40], usually with additional constraints.

Arising from the aforementioned AGP, there is no general analytical solution to the sensor placement problem. The optimal solution for discrete search spaces can be determined using exhaustive search, but this involves a high computational cost. By using other, mostly meta-heuristic approaches, the problem with the high computational costs could be alleviated, but the selected sensor placement could not be guaranteed to be optimal.

The aim of the present study is two-fold. First, a variant of the exhaustive search approach was proposed to address the problem of two-dimensional and three-dimensional sensor placement. This modified exhaustive search method is used to establish a ground truth for the single sensor placement problem. As shown in Section 3, exhaustive search is not feasible for three-dimensional environment models and the proposed approach mitigates some computational costs.

Second, following the same idea of finding approaches that lead to (near) optimal solutions, we explore nature-inspired genetic algorithms and the impact of the number of evaluations of the optimization function for these algorithms on both accuracy and computational cost. In addition, the effects of the rasterization of the environment are analyzed. This enables users to select the right parameter values and balance coverage and computational costs.

To summarize, the following contributions are presented in this manuscript:

- An exhaustive search-based approach that serves as ground truth for sensor placement in two-dimensional and three-dimensional environments.
- Analysis of the influence of the number of evaluations of the optimization function and the rasterization of the environment on the performance of stochastic optimization sensor placement algorithms.

- Evaluation of three stochastic optimization algorithms for sensor placement in two-dimensional and three-dimensional environments with selected parameters and comparison with the ground truth values.

This manuscript is structured as follows. Section 2 describes the used environment and sensor models, optimization function and algorithms as well as the newly proposed recursive exhaustive Search. In Section 3, the results are then presented, analyzed and discussed. Finally, concluding remarks are given in Section 4.

2. Materials and Methods

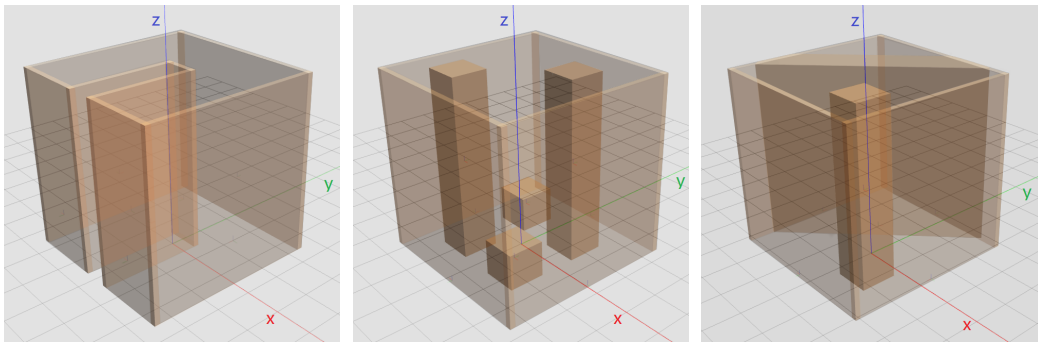
In our previous research [15], we introduced environment and sensor models as well as an optimization function that is crucial for evaluating sensor placement in a given environment. To ensure an accurate assessment of sensor positioning, we utilize environment models that provide a rough two-dimensional and three-dimensional representation of the real-world environment. Our proposed sensor models establish visibility calculations by employing probabilistic coverage models tailored for both isotropic and directional sensors. In addition, the optimization function utilizes a continuous, single-objective approach that aims to minimize the global loss value, which is the complement of the area coverage.

In this section, we provide a brief overview of these models and functions and explain the role they play in improving the efficiency of sensor placement in specific environments. Furthermore, a variant of the exhaustive search and its adaptation to the two-dimensional and three-dimensional sensor placement problem is presented to create a ground truth solution for single sensor placement.

2.1. Environment Model

To accurately assess sensor placement, we rely on environment models proposed in our previous research that closely represent the real environment based on descriptions of free and occupied spaces. These models, which typically describe the layout of the ground plane, are created through three-dimensional polygonal modeling. Although these models are primarily three-dimensional, they can also be simplified into two-dimensional representations if necessary by focusing on their ground planes.

Three distinct test environments were defined by altering their shapes and the arrangement of obstacles within them. The first environment, seen in Figure 1a, features a U-shape without obstacles, representing a scenario where multiple sensors are necessary due to the curved layout obstructing visibility. In the second environment, Figure 1b, a simple quadratic shape with multiple obstacles, limits the coverage of individual sensors. The third environment, Figure 1c, simulates scenarios where obstacles isolate parts of the environment, testing the optimization algorithm's response to discontinuities and the placement of sensors in both isolated and non-isolated areas.



(**a**) Environment 1 (**b**) Environment 2 (**c**) Environment 3

Figure 1. 3D isometric views of used environment models.

To effectively calculate the spatial coverage, the environment is divided into a final number of voxels based on a rasterization parameter r_v that determines the granularity of

coverage. The number of voxels depends on the layout, the dimensions of the free and occupied spaces, and the selected rasterization parameter, and is given in Table 1. In general, it is assumed that a higher number of voxels provides a more accurate description of the world.

Table 1. The number of voxels for two-dimensional and three-dimensional representations of used environment models.

r_v [m]	Environment 1		Environment 2		Environment 3	
	2D	3D	2D	3D	2D	3D
1	36	216	20	152	23	138
0.5	115	1265	85	1079	88	968
0.1	2331	116,550	2117	115,530	2099	104,950

By employing both two-dimensional and three-dimensional spatial representations and adjusting the rasterization parameter, we can evaluate the performance of optimization algorithms on test cases of varying complexity.

2.2. Sensor Models

Sensor models are defined by specifying rules for calculating the visibility for each voxel surrounding the sensor within the environment. Proposed sensor models employ a probabilistic coverage model. There are two main types of sensors that differ in their sensing capabilities: sensors with isotropic sensing and sensors with directional sensing.

The visibility v_{vs} between each voxel v_i and the sensor s_j is determined as the product of three fundamental functions:

$$v_{vs}(v_i, s_j) = v_d(v_i, s_j) \cdot v_\varphi(v_i, s_j) \cdot v_\theta(v_i, s_j), \tag{1}$$

where v_d represents the visibility based on the distance between the sensor and the observed voxel, while v_φ and v_θ correspond to the visibilities determined by azimuth and inclination angles between the sensor and the voxel. These calculations vary depending on the sensor type.

Visibility is only calculated if there is a line of sight between the voxel and the sensor. The distance d_{vs} between the voxel and the sensor, which is used in the distance visibility function v_d, is calculated using the following Euclidean distance formula:

$$d_{vs}(v_i, s_j) = \sqrt{(v_{i_x} - s_{j_x})^2 + (v_{i_y} - s_{j_y})^2 + (v_{i_z} - s_{j_z})^2}. \tag{2}$$

Both azimuth φ and inclination θ are calculated as angles within the spherical coordinate system:

$$\varphi_{vs}(v_i, s_j) = \arctan 2\Big(v_{i_y} - s_{j_y}, v_{i_x} - s_{j_x}\Big), \tag{3}$$

$$\theta_{vs}(v_i, s_j) = \arcsin\left(\frac{v_{i_z} - s_{j_z}}{d_{vs}(v_i, s_j)}\right), \tag{4}$$

and used in the calculation of the azimuth visibility v_φ and inclination visibility v_θ, respectively.

In this study, a model of an isotropic radio beacon is used, where the visibility of the distance v_d is based on the Received Signal Strength Indicator (RSSI). The SNR is normalized by $rssi_{max}$, resulting in v_d. The azimuth visibility v_φ and the inclination visibility v_θ are equal to 1, as it is assumed that the isotropic sensor detects (or emits) the signal equally in all directions.

2.3. Optimization Function

To determine optimal sensor positions, an optimization function is required, combining the environment and sensor models based on the selected metric. This function

represents a minimization problem suitable for various optimization algorithms that are non-derivative, nonlinear and constrained.

The metric chosen for this optimization is the area coverage ratio, which indicates the proportion of the target area that is covered by the sensors. The goal of the optimization is to minimize the global loss value, which is essentially the complement of the coverage.

The loss l_{vs} associated with each sensor–voxel pair is calculated as the complement of their visibility:

$$l_{vs}(v_i, s_j) = 1 - v_{vs}(v_i, s_j). \tag{5}$$

Since a voxel can be visible from multiple sensors, the voxel loss L_v is calculated as the product of the losses from each of the m sensors within the set S:

$$L_v(v_i, S) = \prod_{j=1}^{m} l_{vs}(v_i, s_j). \tag{6}$$

The global loss value L is then obtained by averaging the individual voxel losses L_v over all n voxels within the environment V:

$$L(V, S) = \frac{1}{n} \sum_{i=1}^{n} L_v(v_i, S). \tag{7}$$

Sensor positions, yaw and pitch angles are characterized by continuous values, while the voxel positions adopt discrete values derived from the environment's rasterization parameter. Although the proposed solution utilizes a loss minimization function that aims to reduce the loss value $L(V, S)$, a coverage value $C(V, S)$ is used in the results:

$$C(V, S) = 1 - L(V, S). \tag{8}$$

Moreover, this coverage value $C(V, S)$ can be used for optimization problems that utilize a maximization function.

2.4. Recursive Exhaustive Search for Sensor Placement

Analytically determining the optimal sensor placement in space is not feasible [5,6,9,10,41], necessitating methods such as exhaustive search (ES) to explore all possible configurations within a discrete search space defined by a rasterization value r_e. However, due to the continuous nature of the sensor positions, the coverage achieved with ES may not be optimal. In addition to the problems with the discrete search space that affect the accuracy of the optimization algorithms, the computational complexity is usually high and often results in the need to find a balance between accuracy and computational complexity.

To tackle both the accuracy and computational complexity of ES, a recursive exhaustive search (RES) approach was developed that combines the concepts of ES and greedy search (GS) to recursively traverse the environment in search of the optimal solution. In RES, the potential sensor positions are initially separated by r_e, same as in ES. As shown in Algorithm 1, in each iteration, the search space area is recursively refined around the positions with the highest coverage until the termination conditions are met.

The step size r_e for ES was chosen so that it is one order of magnitude lower than the rasterization value of the environment r_v. The coverage value C is calculated for each of the possible positions and the position or positions with the highest coverage value are selected. With RES, this process is repeated at each iteration, shrinking the search space around the selected position and halving its size for each of the axes of the global coordinate system. In each iteration, the value of r_e is also reduced by one order of magnitude. The execution is terminated if the size of the search space falls below the value of the floating point precision $r_e \leq 1 \times 10^{-8}$ m.

Algorithm 1 Recursive Exhaustive Search (RES)

$r_e = 1.0$
$lower\_bounds = [x_{min}, y_{min}, z_{min}]$-initially set to lower bounds of environment model
$upper\_bounds = [x_{max}, y_{max}, z_{max}]$-initially set to upper bounds of environment model
function RES($r_e, lower\_bounds, upper\_bounds$)
    if $r_e \leq 1 \times 10^{-8}$ **then**
        return $0, None$
    end if
    $potential\_solutions \leftarrow$ sample the area between min and max values for each axis with a step size of r_e
    $C_{max}, best\_solutions \leftarrow$ solution(s) with highest coverage value C from the $potential\_solutions$
    $half\_dim = (upper\_bounds - lower\_bounds)/4$
    for $solution$ in $best\_solutions$ **do**
        $C, current\_solutions \leftarrow$ RES($r_e/10, solution - half\_dim, solution + half\_dim$)
        if $C > C_{max}$ **then**
            $C_{max} = C$
            Replace $best\_solutions$ with the $current\_solutions$
        else if $C == C_{max}$ **then**
            Append $current\_solutions$ to the list of $best\_solutions$
        end if
    end for
    return $C_{max}, best\_solutions$
end function

While ES provides comprehensive coverage exploration, RES offers a refined approach with reduced computational demands. Further optimization of RES may enhance its practical applicability for sensor placement problems in real-world scenarios.

2.5. Optimization Algorithms

Three algorithms were selected for this study, all of which fall within the domain of population-based stochastic genetic optimization techniques. These methods, namely Particle Swarm Optimization (PSO) [42], Artificial Bee Colony (ABC) [43] and Fireworks Algorithm (FWA) [44,45], are tailored to the optimization of nonlinear functions within multidimensional spaces, as described earlier. The variables to be optimized are the positions of the sensors, which form a multidimensional search space whose dimensions vary depending on environmental factors and the properties of the sensors used.

The choice of the PSO algorithm stems from its status as one of the most widely used swarm optimization algorithms, which has proven itself in various optimization problems. Alongside PSO, the ABC algorithm is utilized, as a representative of more modern optimization techniques, which is characterized by its effectiveness in certain optimization scenarios [43]. Additionally, the inclusion of the FWA algorithm demonstrates a recent advancement in swarm optimization techniques, presenting notable improvements over the PSO algorithm, according to [44]. In particular, the combination of these algorithms, as used in [46] for the path planning of mobile robots, highlights their suitability for tackling optimization problems in continuous space, and a number of other applications [47–49].

Since there are no known solutions to the sensor placement problem and it cannot be solved analytically, the optimization cannot be directed toward a fixed target. Therefore, we use a "fixed-cost" approach, where the number of evaluations of the optimization functions for each algorithm within a given test case is limited [50]. Specifically, the number of evaluations is set to $E = E_v \cdot D$, where E_v is the number of evaluations of the optimization function per independent variable and D is the number of independent variables. In this study, the influence of the number of evaluations of the optimization function per independent variable E_v is analyzed.

For the placement of the single isotropic sensor in the two-dimensional environment, $D = 2$, which corresponds to the positions x and y of said sensor. In the case of a three-dimensional environment, $D = 3$ because an additional axis z is added.

2.6. Experimental Setup

The three environments described above were used for all experiments. Due to the computational requirements of these extensive experiments, the supercomputer "Bura" of the University of Rijeka was used [51]. The tests with ES and RES were parallelized on 20 nodes with two Xeon E5 processors each. The tests with stochastic optimization algorithms were performed on a varying number of nodes, with each test running exclusively on a single core.

3. Results and Discussion

In this section, we present the results of our study. We start with the results of a proposed solution approach for ground truth based on exhaustive search. We then compare the coverage results for the placement of a single isotropic sensor in three two-dimensional and three-dimensional environments based on the number of evaluations of the optimization function and the rasterization parameter. Following the comparison, the coverage values obtained with the three optimization algorithms are compared with the ground truth value calculated with modified RES.

3.1. Ground Truth Estimation

Since it is not possible to analytically determine the optimal arrangement of sensors and multiple sensor configurations could be optimal, we decided to use a single metric with which to compare the optimization algorithms. If there are multiple optimal sensor configurations, they result in the same coverage value C. Therefore, we chose the coverage value C as the criterion for determining the accuracy of sensor placement and used it as the ground truth representation instead of the sensor configuration.

The time per evaluation TPE mainly depends on the number of voxels in the environment and the number and type of sensors placed. On this basis, TPE and, extrapolated from it, the average computation time of an optimization function are constant. Both the ES and RES approaches are parallelized and are compared based on the number of evaluations of the optimization function and not on the computation time.

Both ES and RES were evaluated and compared on test cases involving the placement of a single omnidirectional sensor in two-dimensional and three-dimensional models of all three environments with a rasterization value of $r_v = 0.1\,\text{m}$. ES was conducted with a step size of $r_e = 0.01\,\text{m}$, while for RES, the initial value was $r_e = 1\,\text{m}$.

Table 2 shows the measured time per evaluation TPE in seconds, the achieved coverage values C and the number of evaluations E of the optimization function for both approaches. For two-dimensional environments, the computation time for ES was between 4 and 7.5 min, while for RES it was an order of magnitude lower, between 1 and 4 s. For three-dimensional environments, the difference is even more pronounced, ranging from 22 to 45 min for RES. Measuring the computation time for ES would be possible, but not feasible, as it is estimated to be between 80 and 204 days for the same test cases. The computation time between ES and RES ranged from a factor of 100 for two-dimensional environments to a factor of 5000 for three-dimensional environments.

From the data obtained, it is evident that there is no difference in the coverage achieved between the two approaches for two-dimensional environments. However, for three-dimensional environments, the achieved space coverage results for the ES approach are not shown since the computation time is extended beyond practical limits.

Table 2. Time per evaluation TPE, coverage values C and number of evaluations E of the optimization function for ES and RES for single sensor placement problem in all three environments.

		Environment 1		Environment 2		Environment 3	
		2D	3D	2D	3D	2D	3D
TPE [s]		0.53	30.72	0.99	74.49	0.85	56.25
ES	C	0.1446	-	0.1191	-	0.1359	-
	E	221,301	1.107×10^9	210,197	1.133×10^9	205,530	1.028×10^9
RES	C	0.1446	0.0795	0.1191	0.0814	0.1359	0.0742
	E	1567	21,230	1533	17,051	1534	16,874

3.2. Analysis of the Impact of the Number of Evaluations

The importance of choosing the right number of evaluations of the optimization function lies in the need for faster processing without significant loss of total coverage C. In this subsection, we study the impact of the evaluations for the problem of placing a single isotropic sensor in three selected environments. To account for the stochastic variability inherent in optimization algorithms, each test case was repeated 30 times. For the sake of brevity, the results are presented for one environment only. The analysis of the coverage in the other environments leads to the same results.

For all tests, a rasterization value is set to $r_v = 0.1$ m. Table 3 shows the descriptive statistics for using different numbers of evaluations E_v of the optimization for the sensor placement in the two-dimensional representation of the Environment 1. This table shows the average and standard deviation of the coverage C of all runs for each algorithm.

Table 3. Coverage averages and standard deviations of the different algorithms for each number of evaluations for two-dimensional Environment 1.

Evaluations E_v	ABC		FWA		PSO	
	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
25	0.1417	0.0026	0.1408	0.0034	0.1435	0.0012
50	0.1432	0.0017	0.1417	0.0025	0.1443	0.0002
75	0.1440	0.0007	0.1430	0.0011	0.1445	<0.0001
100	0.1442	0.0004	0.1430	0.0012	0.1445	<0.0001
150	0.1444	0.0002	0.1433	0.0011	0.1445	<0.0001
200	0.1444	0.0002	0.1437	0.0006	0.1445	<0.0001
300	0.1445	0.0001	0.1435	0.0006	0.1446	<0.0001
400	0.1445	0.0002	0.1436	0.0006	0.1446	<0.0001
500	0.1445	<0.0001	0.1439	0.0005	0.1446	<0.0001
600	0.1445	0.0001	0.1437	0.0004	0.1446	<0.0001
700	0.1445	0.0002	0.1439	0.0004	0.1446	<0.0001
800	0.1445	0.0001	0.1442	0.0002	0.1446	<0.0001
900	0.1445	0.0002	0.1440	0.0004	0.1446	<0.0001
1000	0.1445	0.0001	0.1441	0.0003	0.1446	<0.0001

In order to statistically analyze the effects of the optimization algorithm used and the number of evaluations of the optimization function on the average coverage, we performed a two-way repeated measures (RM) ANOVA. Namely, a 3×14 ANOVA was utilized, with *Algorithms* (three instances) and *Number of evaluations* (14 instances) being the within-subjects factors. The test yielded the following results:

- The mean coverage differed significantly between the optimization *Algorithms*:

$$F(2, 1218) = 173.998, p < 0.001.$$

A post hoc analysis with a Bonferroni adjustment showed that PSO achieved statistically higher coverage compared to ABC and FWA. ABC also outperformed FWA

in these tests. The full results of the pairwise post hoc comparisons can be found in Table 4.

- Mean coverage differed significantly between the *Number of evaluations*:

$$F(13, 1218) = 47.555, p < 0.001.$$

A post hoc analysis with a Bonferroni adjustment confirmed a statistically significant difference in area coverage obtained with $E_v = 25$ and $E_v = 50$ evaluations compared to more. In most other cases, there is no statistically significant difference.

- The interaction between the factors *Algorithm* $*$ *Number of evaluations* is statistically significant:

$$F(26, 1218) = 5.446, p < 0.001.$$

Table 4. Post hoc comparison for *Algorithms* for two-dimensional Environment 1 with factor *Number of evaluations*.

		Mean Difference	SE	t	p_{bonf}
ABC	FWA	8.408×10^{-4}	6.368×10^{-5}	13.203	<0.001
	PSO	-3.064×10^{-4}	6.368×10^{-5}	−4.811	<0.001
FWA	PSO	−0.001	6.368×10^{-5}	−18.014	<0.001

Table 5 shows the descriptive statistics for using different numbers of evaluations E_v of the optimization for the sensor placement in the three-dimensional representation of Environment 1. This table shows the average and standard deviation of the coverage C of all runs for each algorithm.

Table 5. Coverage averages and standard deviations of the different algorithms for each number of evaluations for three-dimensional Environment 1.

Evaluations E_v	ABC		FWA		PSO	
	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
25	0.0764	0.0045	0.0738	0.0039	0.0790	0.0004
50	0.0786	0.0012	0.0766	0.0020	0.0795	0.0001
75	0.0791	0.0005	0.0772	0.0017	0.0795	0.0001
100	0.0793	0.0004	0.0769	0.0024	0.0795	<0.0001
150	0.0794	0.0002	0.0773	0.0013	0.0795	<0.0001
200	0.0795	0.0001	0.0776	0.0011	0.0795	<0.0001
300	0.0795	0.0001	0.0779	0.0008	0.0795	<0.0001
400	0.0795	<0.0001	0.0781	0.0008	0.0795	<0.0001
500	0.0795	0.0001	0.0783	0.0007	0.0795	<0.0001
600	0.0795	0.0001	0.0784	0.0008	0.0795	<0.0001
700	0.0795	<0.0001	0.0783	0.0006	0.0795	<0.0001
800	0.0795	<0.0001	0.0786	0.0005	0.0795	<0.0001
900	0.0795	0.0001	0.0785	0.0005	0.0795	<0.0001
1000	0.0795	<0.0001	0.0786	0.0006	0.0795	<0.0001

We again performed a two-way 3×14 RM ANOVA, where *Algorithms* (three instances) and *Number of evaluations* (14 instances) were the within-subjects factors. The test yielded the following results:

- The mean coverage differed significantly between the optimization *Algorithms*:

$$F(2, 1218) = 335.709, p < 0.001.$$

- Mean coverage differed significantly between the *Number of evaluations*:

$$F(13, 1218) = 37.571, p < 0.001.$$

In both cases, the post hoc analysis with a Bonferroni adjustment led to the same conclusions as for the two-dimensional placement. The full results of the pairwise post hoc comparisons for the factor *Algorithm* can be found in Table 6.

- The interaction between the factors *Algorithm* $*$ *Number of evaluations* is statistically significant:

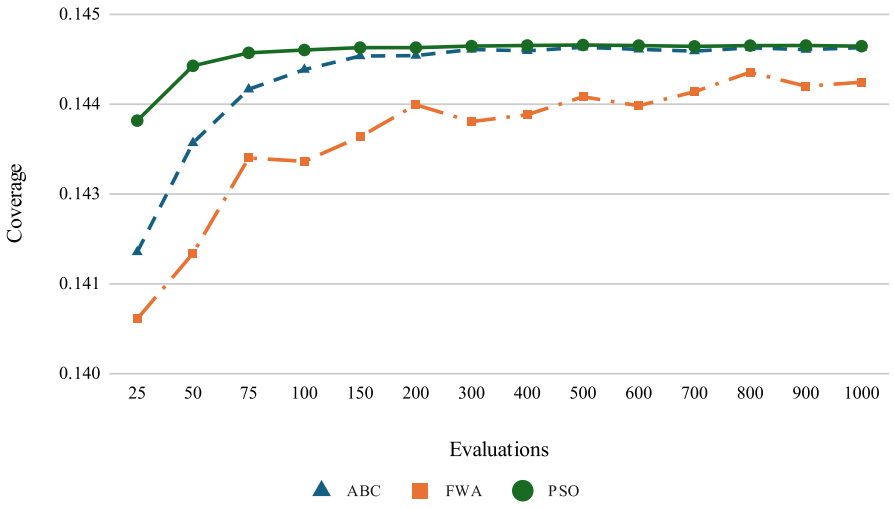
$$F(26, 1218) = 7.787, p < 0.001.$$

Table 6. Post hoc comparison for *Algorithm* for three-dimensional Environment 1 with factor *Number of evaluations*.

		Mean Difference	SE	t	p_{bonf}
ABC	FWA	0.002	7.942×10^{-5}	20.204	<0.001
	PSO	-3.135×10^{-4}	7.942×10^{-5}	−3.948	<0.001
FWA	PSO	−0.002	7.942×10^{-5}	−24.152	<0.001

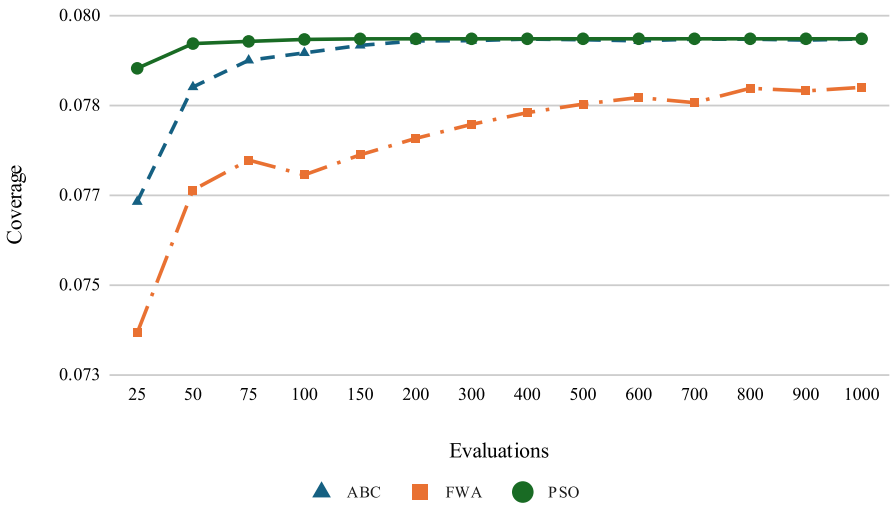
Although the statistical analysis showed that 50 evaluations per independent variable achieved a coverage value that was not significantly different from that achieved with more evaluations, the statistically significant interaction between the factors prompted us to perform a post hoc analysis of all factors. This post hoc analysis showed that PSO performs better than ABC only for $E_v = 50$ for two-dimensional environments and for $E_v = 25$ evaluations for three-dimensional environments, while there is no statistically significant difference for a higher number of evaluations. Both algorithms outperform FWA up to $E_v = 150$ for two-dimensional environments and $E_v = 400$ for three-dimensional environments, after which there is no statistically significant difference between the pairs of algorithms. This difference can be seen in Figure 2.

As can be seen in Figure 3, the computation time increases linearly with the number of evaluations E_v. The computation time for ABC and PSO overlaps, while it increases faster for FWA.



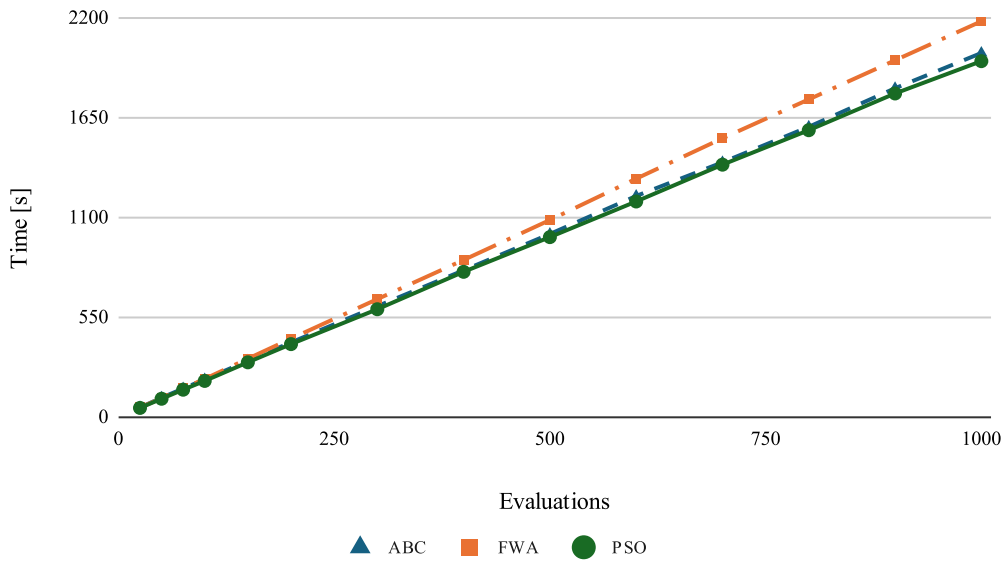
(a) 2D

Figure 2. *Cont.*



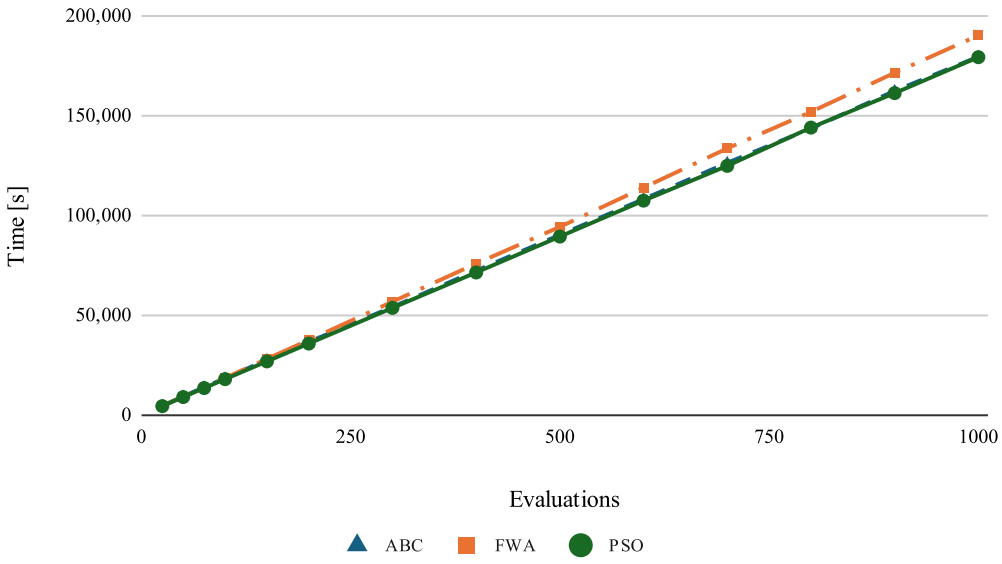
(**b**) 3D

Figure 2. Coverage value C for each of the algorithms based on the number of evaluations E_v for Environment 1.



(**a**) 2D

Figure 3. *Cont.*



(**b**) 3D

Figure 3. Computation time for each of the algorithms based on the number of evaluations E_v for Environment 1.

3.3. Analysis of the Impact of the Rasterization Value

While the number of evaluations has a linear influence on the computational time, it is expected that the rasterization value r_v has an exponential influence on the calculation. In this subsection, we examine the effects of the rasterization values on the computed coverage value C. For all tests, the number of evaluations per independent variable is set to $E_v = 200$. Table 7 shows the descriptive statistics for using different rasterization values r_v of the optimization for sensor placement in the two-dimensional Environment 1. This table shows the average and standard deviation of the coverage C of 30 runs.

Table 7. Coverage averages and standard deviations of the different algorithms for each rasterization for two-dimensional Environment 1.

Rasterization r_v [m]	ABC		FWA		PSO	
	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
0.1	0.1444	0.0002	0.1437	0.0006	0.1445	<0.0001
0.2	0.1405	0.0002	0.1398	0.0006	0.1407	<0.0001
0.3	0.1370	<0.0001	0.1359	0.0008	0.1370	<0.0001
0.4	0.1352	0.0004	0.1345	0.0006	0.1354	0.0001
0.5	0.1303	0.0003	0.1291	0.0007	0.1303	0.0003
0.6	0.1276	0.0006	0.1261	0.0009	0.1279	0.0007
0.7	0.1294	0.0001	0.1268	0.0017	0.1292	0.0005
0.8	0.1249	0.0002	0.1218	0.0021	0.1249	<0.0001
0.9	0.1200	0.0006	0.1167	0.0021	0.1198	0.0007
1.0	0.1201	0.0005	0.1161	0.0026	0.1198	0.0017

Again, to statistically analyze the effects of the optimization algorithm used and the rasterization value on the average coverage, we performed a two-way RM ANOVA. Namely, a 3×10 ANOVA was utilized, with *Algorithms* (three instances) and *Rasterization* (10 instances, 0.1 m to 1.0 m, step 0.1 m) being the within-subjects factors. The test yielded the following results:

- The mean coverage differed significantly between the optimization *Algorithms*:

$$F(2, 870) = 402.097, p < 0.001.$$

A post hoc analysis with a Bonferroni adjustment showed that there is no statistical difference between the coverage achieved by ABC and PSO, while both achieve statistically higher coverage compared to FWA in these tests. The full results of the pairwise post hoc comparisons can be found in Table 8.

- Mean coverage differed significantly between the *Rasterization*:

$$F(9, 870) = 7518.017, p < 0.001.$$

A post hoc analysis with a Bonferroni adjustment showed statistical significance between almost all *Rasterization* values, with the only pair that yielded a statistically non-significant difference being $r_v = 0.9\,\text{m}$ and $r_v = 1.0\,\text{m}$.

- The interaction between the factors *Algorithm* $*$ *Rasterization* is statistically significant:

$$F(18, 870) = 15.191, p < 0.001.$$

Table 8. Post hoc comparison for *Algorithm* for two-dimensional Environment 1 with factor *Rasterization*.

		Mean Difference	SE	t	p_{bonf}
ABC	FWA	0.002	7.760×10^{-5}	24.560	<0.001
	PSO	1.452×10^{-7}	7.760×10^{-5}	0.002	1.000
FWA	PSO	−0.002	7.760×10^{-5}	−24.558	<0.001

Table 9 shows the descriptive statistics for using different rasterization values r_v of the optimization for the sensor placement in the three-dimensional representation of Environment 1. This table shows the average and standard deviation of the coverage C of all runs for each algorithm.

Table 9. Coverage averages and standard deviations of the different algorithms for each rasterization for three-dimensional Environment 1.

Rasterization r_v [m]	ABC		FWA		PSO	
	Average	St. Dev.	Average	St. Dev.	Average	St. Dev.
0.1	0.0795	0.0001	0.0776	0.0011	0.0795	<0.0001
0.2	0.0760	0.0001	0.0743	0.0013	0.0761	0.0001
0.3	0.0728	0.0002	0.0709	0.0012	0.0729	<0.0001
0.4	0.0703	0.0002	0.0692	0.0005	0.0704	0.0001
0.5	0.0670	0.0001	0.0659	0.0007	0.0670	0.0001
0.6	0.0638	0.0002	0.0626	0.0007	0.0639	0.0002
0.7	0.0628	0.0001	0.0616	0.0007	0.0628	0.0001
0.8	0.0594	0.0001	0.0580	0.0005	0.0594	0.0001
0.9	0.0544	0.0001	0.0539	0.0003	0.0543	0.0001
1.0	0.0544	0.0001	0.0538	0.0005	0.0543	0.0001

Again, we performed a 3×10 two-way RM ANOVA with the same within-subjects factors. The test yielded the following results:

- The mean coverage differed significantly between the optimization *Algorithms*:

$$F(2, 870) = 683.308, p < 0.001.$$

- Mean coverage differed significantly between the *Rasterization*:

$$F(9, 870) = 28115.796, p < 0.001.$$

For both factors, a post hoc analysis with a Bonferroni adjustment yielded the same results as for the two-dimensional Environment. The full results of the pairwise post hoc comparisons for the factor *Algorithms* can be found in Table 10.

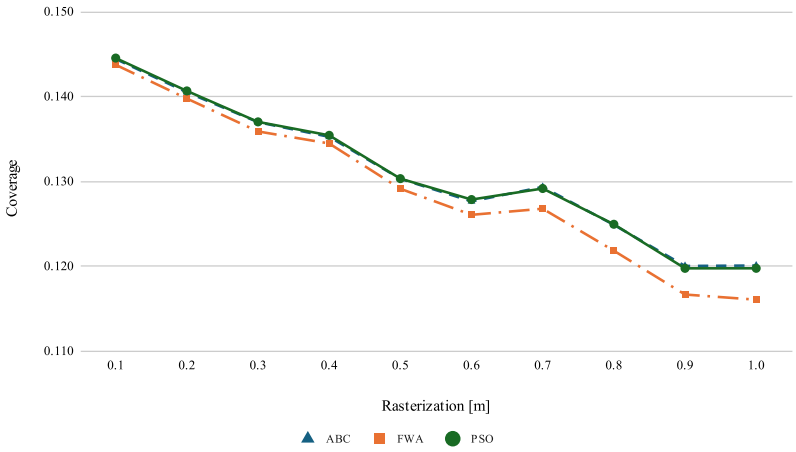
- The interaction between the factors *Algorithm* $*$ *Rasterization* is statistically significant:

$$F(18, 870) = 10.454, p < 0.001.$$

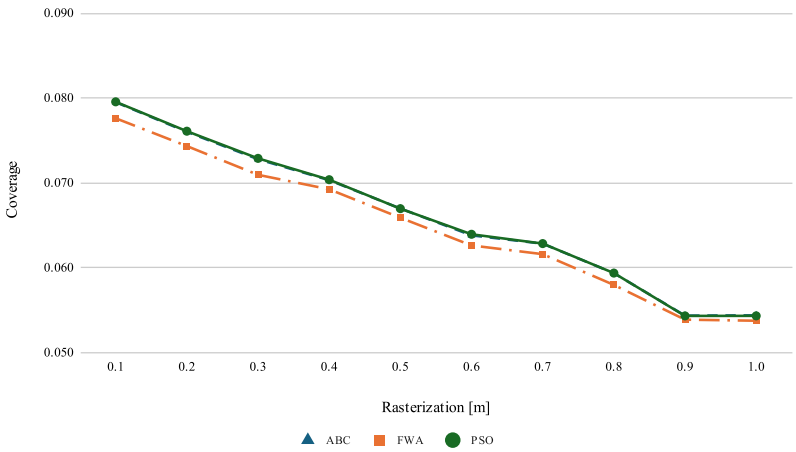
Table 10. Post hoc comparison for *Algorithm* for three-dimensional Environment 1 with factor *Rasterization*.

		Mean Difference	SE	t	p_{bonf}
ABC	FWA	0.001	3.952×10^{-5}	31.621	<0.001
	PSO	-3.056×10^{-5}	3.952×10^{-5}	−0.773	1.000
FWA	PSO	−0.001	3.952×10^{-5}	−32.395	<0.001

The statistically significant interaction between the factors prompted us to conduct a post hoc analysis of all factors. This post hoc analysis showed that for the two-dimensional environment there is no statistically significant difference between the factors *Algorithms* for $r_v = 0.1$ m and $r_v = 0.2$ m, while for values of $r_v \geq 0.3$ m both ABC and PSO perform better than FWA, with no significant difference between them. For the three-dimensional environment, a post hoc analysis showed that ABC and PSO outperform FWA for all rasterization values. This can be seen in Figure 4.



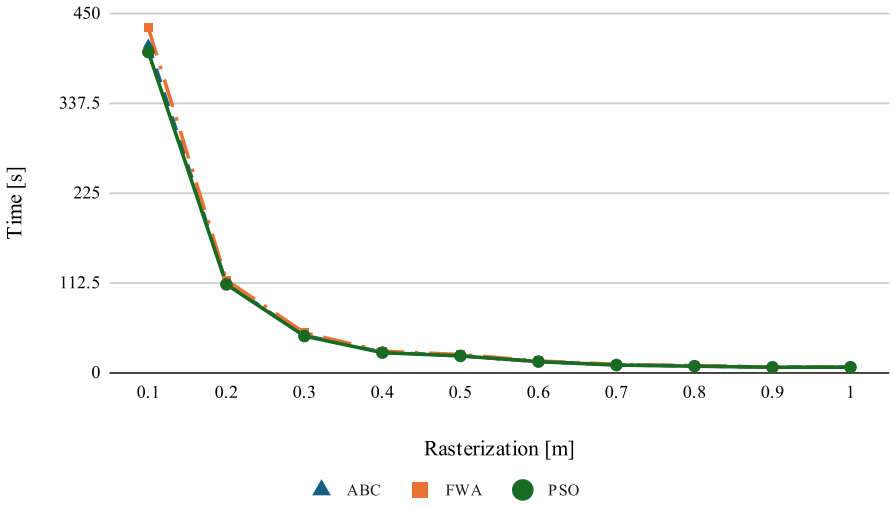
(**a**) 2D



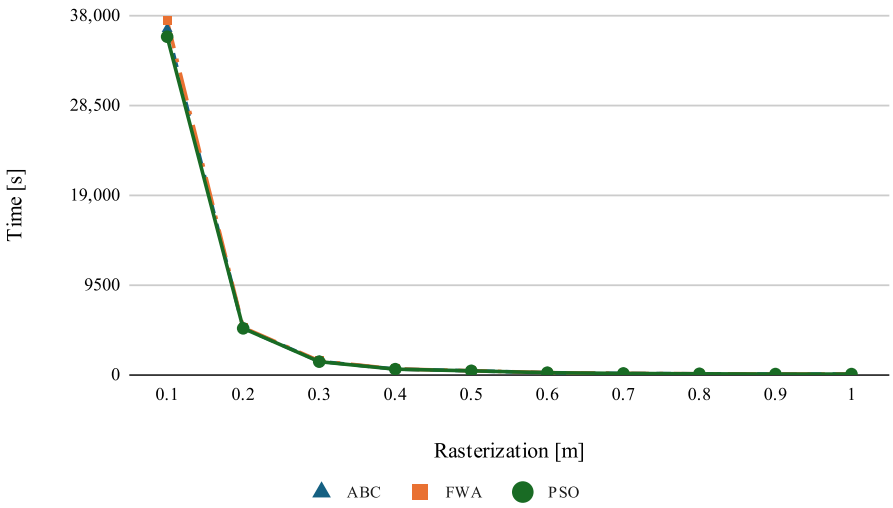
(**b**) 3D

Figure 4. Coverage value C for each of the algorithms based on the rasterization value r_v for Environment 1.

As can be seen in Figure 5, the computation time increases exponentially with the smaller rasterization value r_v and it overlaps again for ABC and PSO, while it increases somewhat faster for FWA.



(a) 2D



(b) 3D

Figure 5. Computation time for each of the algorithms based on the rasterization value r_v for Environment 1.

3.4. Comparison of RES and Stochastic Algorithms

Based on the conclusions from the analysis in the two previous subsections, we chose $r_v = 0.1$ m as this led to the best overall coverage C. For the stochastic optimization algorithms, $E_v = 150$ was chosen as this was the first value where there was a significant difference between the algorithms for two-dimensional environments. For three-dimensional environments, a value of $E_v = 400$ was chosen for the same reasons.

Table 11 shows the results of the coverage value C and the total number of evaluations E. The stochastic algorithms, in particular ABC and PSO, performed as well as RES and in most cases reached or came close to the ground truth coverage value. This coverage performance metric was tracked with a large impact on computation time, with the stochastic algorithms being more than 5 times faster for two-dimensional environments and between 14 and 18 times faster for three-dimensional environments. If we were to

compare the computation time with ES and not with the proposed RES, the impact would be about 500 times for the two-dimensional environments and almost 100,000 times for the three-dimensional environments.

Table 11. Coverage values C and the total number of evaluations E for RES and three selected optimization algorithms for single sensor placement problem in all three environments.

		Environment 1		**Environment 2**		**Environment 3**	
		2D	**3D**	**2D**	**3D**	**2D**	**3D**
RES	C	0.1446	0.0795	0.1191	0.0814	0.1359	0.0742
	E	1567	21,230	1533	17,051	1534	16,874
ABC	C	0.1444	**0.0795**	**0.1190**	**0.0814**	0.1352	0.0739
FWA	C	0.1433	0.0781	0.1170	0.0806	0.1346	0.0732
PSO	C	**0.1445**	**0.0795**	**0.1190**	**0.0814**	**0.1356**	**0.0741**
	E	300	1200	300	1200	300	1200

4. Conclusions

In this paper, we presented a novel methodology for addressing the complex challenge of optimal sensor placement in two-dimensional and three-dimensional spaces. We have proposed and thoroughly investigated a new variant of exhaustive search, the recursive exhaustive search (RES). It provides the same or better coverage than traditional exhaustive search while significantly reducing the computational overhead, especially in three-dimensional contexts. This efficiency makes RES a robust benchmark for evaluating the performance of various stochastic optimization algorithms in sensor placement tasks.

We then investigated the effects of two key factors—the number of evaluations of the optimization function and the rasterization of the environment—on the efficiency of the sensor placement strategies. Our comparative analysis of three prominent stochastic optimization algorithms—Artificial Bee Colony (ABC), Fireworks Algorithm and Particle Swarm Optimization (PSO)—with the benchmarks established by RES provided valuable insights into their effectiveness. In particular, the ABC and PSO algorithms performed commendably. They often achieved or came very close to the optimal coverage values determined by RES, while significantly reducing the computation time.

The main limitation of the proposed approach is the increase in computational complexity with the size and topological complexity of the environment models. This limitation can be mitigated by a parallelized implementation of the computation of the optimization functions, which is executed on multiple CPU or GPU cores. Another limitation related to the proposed RES is the potentially suboptimal solution, as mentioned in Section 2.4.

The insights gained emphasize the importance of fine-tuning parameters and balancing computational complexity and solution quality. These considerations can inform decision-making in real-world sensor networks and improve their efficiency and reliability. Future research directions could investigate advanced optimization techniques or hybrid approaches to further improve the efficiency of sensor placement. Furthermore, investigating the applicability of our findings to specific domains and considering dynamic environments could lead to more robust and adaptive sensor placement strategies. One of the possibilities is to train an agent (sensor) model using reinforcement learning, which could lead to the possibility of finding optimal results in an unknown environment.

Author Contributions: Conceptualization, D.S.; Methodology, M.B. and D.S.; Software, M.B. and D.S.; Validation, M.B., D.P. and K.L.; Formal analysis, M.B.; Data curation, D.P.; Writing—original draft, M.B., D.P., K.L. and D.S.; Writing—review & editing, M.B., D.P., K.L. and D.S.; Visualization, M.B. and D.S.; Supervision, D.S.; Project administration, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the University of Rijeka, Rijeka, Croatia under grant uniri-iskusni-tehnic-23-261.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

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

References

1. Zhang, G.; Dong, B.; Zheng, J. Visual Sensor Placement and Orientation Optimization for Surveillance Systems. In Proceedings of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Krakow, Poland, 4–6 November 2015; pp. 1–5. [CrossRef]
2. Batistić, L.; Tomic, M. Overview of indoor positioning system technologies. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 0473–0478.
3. Hržić, F.; Sušanj, D.; Lenac, K. Optimal beacon positioning for indoor drone navigation. In Proceedings of the 12th Annual Baška GNSS Conference (2018), Baška, Krk Island, Croatia, 6–9 May 2018; pp. 109–117.
4. Safak, G. *The Art-Gallery Problem: A Survey and an Extension*; Skolan för Datavetenskap och Kommunikation, Kungliga Tekniska högskolan: Stockholm, Sweden, 2009.
5. González-Banos, H. A Randomized Art-Gallery Algorithm for Sensor Placement. In Proceedings of the 17th Annual Symposium on Computational Geometry, Medford, MA, USA, 3–5 June 2001; pp. 232–240.
6. Marzal, J. The Three-Dimensional Art Gallery Problem and Its Solutions. Ph.D. Thesis, School of Information Technology, Murdoch University, Perth, Australia, 2012.
7. Marengoni, M.; Draper, B.A.; Hanson, A.; Sitaraman, R. A system to place observers on a polyhedral terrain in polynomial time. *Image Vis. Comput.* **2000**, *18*, 773–780. [CrossRef]
8. Viglietta, G. Guarding and searching polyhedra. *arXiv* **2012**, arXiv:1211.2483.
9. O'Rourke, J. *Art Gallery Theorems and Algorithms*; Oxford University Press: Oxford, UK, 1987.
10. Li, X.; Yu, W.; Lin, X.; Iyengar, S.S. On Optimizing Autonomous Pipeline Inspection. *IEEE Trans. Robot.* **2012**, *28*, 223–233.
11. Cardei, M.; Wu, J. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Comput. Commun.* **2006**, *29*, 413–420. [CrossRef]
12. Meguerdichian, S.; Koushanfar, F.; Potkonjak, M.; Srivastava, M.B. Coverage problems in wireless ad-hoc sensor networks. In Proceedings of the IEEE INFOCOM 2001, Conference on Computer Communications, Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), Anchorage, AK, USA, 22–26 April 2001; Volume 3, pp. 1380–1387.
13. Salehizadeh, S.M.A.; Dirafzoon, A.; Menhaj, M.B.; Afshar, A. Coverage in Wireless Sensor Networks Based on Individual Particle Optimization. In Proceedings of the International Conference on Networking, Sensing and Control (ICNSC), Chicago, IL, USA, 10–12 April 2010; pp. 501–506. [CrossRef]
14. Binh, H.T.T.; Hanh, N.T.; Van Quan, L.; Dey, N. Improved Cuckoo Search and Chaotic Flower Pollination optimization algorithm for maximizing area coverage in Wireless Sensor Networks. *Neural Comput. Appl.* **2018**, *30*, 2305–2317. [CrossRef]
15. Sušanj, D.; Pinčić, D.; Lenac, K. Effective Area Coverage of 2D and 3D Environments with Directional and Isotropic Sensors. *IEEE Access* **2020**, *8*, 185595–185608. [CrossRef]
16. Sušanj, D. Spatial Sensor Distribution Model for Indoor Environment Coverage. Ph.D. Thesis, Faculty of Engineering, University of Rijeka, Rijeka, Croatia, 2021.
17. Akbarzadeh, V.; Ko, A.H.R.; Gagné, C.; Parizeau, M. Topography-Aware Sensor Deployment Optimization with CMA-ES. In Proceedings of the Parallel Problem Solving from Nature, PPSN XI, Krakow, Poland, 11–15 September 2010; pp. 141–150. [CrossRef]
18. Dhillon, S.; Chakrabarty, K. Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks. In Proceedings of the IEEE Wireless Communications and Networking (WCNC), New Orleans, LA, USA, 16–20 March 2003; Volume 3, pp. 1609–1614. [CrossRef]
19. Hörster, E.; Lienhart, R. On the Optimal Placement of Multiple Visual Sensors. In Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, Santa Barbara, CA, USA, 23–27 October 2006; pp. 111–120. [CrossRef]
20. Bodor, R.; Drenner, A.; Schrater, P.; Papanikolopoulos, N. Optimal Camera Placement for Automated Surveillance Tasks. *J. Intell. Robot. Syst.* **2007**, *50*, 257–295. [CrossRef]
21. Gonzalez-Barbosa, J.J.; Garcia-Ramirez, T.; Salas, J.; Hurtado-Ramos, J.B.; Rico-Jimenez, J.d.J. Optimal Camera Placement for Total Coverage. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 844–848. [CrossRef]
22. Altahir, A.A.; Asirvadam, V.S.; Hamid, N.H.B.; Sebastian, P.; Saad, N.B.; Ibrahim, R.B.; Dass, S.C. Optimizing Visual Sensor Coverage Overlaps for Multiview Surveillance Systems. *IEEE Sens. J.* **2018**, *18*, 4544–4552. [CrossRef]

23. Altahir, A.A.; Asirvadam, V.S.; Sebastian, P.; Hamid, N.H.B.; Ahmed, E.F. Optimizing Visual Sensors Placement With Risk Maps Using Dynamic Programming. *IEEE Sens. J.* **2022**, *22*, 393–404. [CrossRef]
24. Suresh, M.S.S.; Menon, V. A Generic and Scalable Approach to Maximize Coverage in Diverse Indoor and Outdoor Multicamera Surveillance Scenarios. *IEEE Trans. Syst. Man. Cybern. Syst.* **2023**, *53*, 1172–1182. [CrossRef]
25. Yue, T.; Hu, Q. A Genetic Optimization Method for Spatial Layout of Cameras in Video Sensor Networks. In Proceedings of the 11th International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Wuhan, China, 25–28 July 2023; pp. 1–6. [CrossRef]
26. Bhat, P.P.; V, G. Performance analysis of PSO for solving coverage problem in WSN. In Proceedings of the 3rd International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 26–28 May 2023; pp. 49–54. [CrossRef]
27. Morsly, Y.; Aouf, N.; Djouadi, M.S.; Richardson, M. Particle Swarm Optimization Inspired Probability Algorithm for Optimal Camera Network Placement. *IEEE Sens. J.* **2012**, *12*, 1402–1412. [CrossRef]
28. Akbarzadeh, V.; Lévesque, J.C.; Gagné, C.; Parizeau, M. Efficient Sensor Placement Optimization Using Gradient Descent and Probabilistic Coverage. *Sensors* **2014**, *14*, 5525. [CrossRef] [PubMed]
29. Malhotra, A.; Singh, D.; Dadlani, T.; Morales, L.Y. Optimizing Camera Placements for Overlapped Coverage with 3D Camera Projections. In Proceedings of the International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5002–5009. [CrossRef]
30. Hefeeda, M.; Ahmadi, H. Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *21*, 579–593. [CrossRef]
31. Altahir, A.A.; Asirvadam, V.S.; Hamid, N.H.; Sebastian, P.; Saad, N.; Ibrahim, R.; Dass, S.C. Modeling Multicamera Coverage for Placement Optimization. *IEEE Sens. Lett.* **2017**, *1*, 1–4. [CrossRef]
32. Akbarzadeh, V.; Gagné, C.; Parizeau, M.; Mostafavi, M.A. Black-box Optimization of Sensor Placement with Elevation Maps and Probabilistic Sensing Models. In Proceedings of the IEEE International Symposium on Robotic and Sensors Environments (ROSE), Montreal, QC, Canada, 17–18 September 2011; pp. 89–94. [CrossRef]
33. Thiene, M.; Khodaei, Z.S.; Aliabadi, M.H. Optimal Sensor Placement for Maximum Area Coverage (MAC) for Damage Localization in Composite Structures. *Smart Mater. Struct.* **2016**, *25*, 095037. [CrossRef]
34. Huang, C.F.; Tseng, Y.C. The Coverage Problem in a Wireless Sensor Network. *Mob. Netw. Appl.* **2005**, *10*, 115–121. [CrossRef]
35. Papadimitriou, C. Optimal sensor placement methodology for parametric identification of structural systems. *J. Sound Vib.* **2004**, *278*, 923–947. [CrossRef]
36. Zhang, C.; Xu, Y.L. Optimal multi-type sensor placement for response and excitation reconstruction. *J. Sound Vib.* **2016**, *360*, 112–128. [CrossRef]
37. Zhou, J.; Cai, Z.; Zhao, P.; Tang, B. Efficient sensor placement optimization for shape deformation sensing of antenna structures with fiber Bragg grating strain sensors. *Sensors* **2018**, *18*, 2481. [CrossRef]
38. Mendler, A.; Döhler, M.; Ventura, C.E. Sensor placement with optimal damage detectability for statistical damage detection. *Mech. Syst. Signal Process.* **2022**, *170*, 108767. [CrossRef]
39. Sarrate, R.; Blesa, J.; Nejjari, F. Sensor placement for leak monitoring in drinking water networks combining clustering techniques and a semi-exhaustive search. In Proceedings of the 3rd Conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain, 7–9 September 2016; pp. 434–439.
40. Njoya, A.N.; Thron, C.; Barry, J.; Abdou, W.; Tonye, E.; Siri Lawrencia Konje, N.; Dipanda, A. Efficient scalable sensor node placement algorithm for fixed target coverage applications of wireless sensor networks. *IET Wirel. Sens. Syst.* **2017**, *7*, 44–54. [CrossRef]
41. Urrutia, J. Art Gallery and Illumination Problems. In *Handbook of Computational Geometry*; Elsevier: Amsterdam, The Netherlands, 2000; pp. 973–1027.
42. 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]
43. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
44. Tan, Y.; Zhu, Y. Fireworks Algorithm for Optimization. In Proceedings of the International Conference in Swarm Intelligence, Brussels, Belgium, 8–10 September 2010; pp. 355–364. [CrossRef]
45. Li, J.; Tan, Y. The bare bones fireworks algorithm: A minimalist global optimizer. *Appl. Soft Comput.* **2018**, *62*, 454–462. [CrossRef]
46. Tan, Y. *Handbook of Research on Design, Control, and Modeling of Swarm Robotics*; IGI Global: Hershey, PA, USA, 2015.
47. Aslan, S.; Karaboga, D. A genetic Artificial Bee Colony algorithm for signal reconstruction based big data optimization. *Appl. Soft Comput.* **2020**, *88*, 106053. [CrossRef]
48. Yue, Z.; Zhang, S.; Xiao, W. A Novel Hybrid Algorithm Based on Grey Wolf Optimizer and Fireworks Algorithm. *Sensors* **2020**, *20*, 2147. [CrossRef] [PubMed]
49. Fausto, F.; Reyna-Orta, A.; Cuevas, E.; Andrade, Á.G.; Perez-Cisneros, M. From ants to whales: Metaheuristics for all tastes. *Artif. Intell. Rev.* **2020**, *53*, 753–810. [CrossRef]

50. Beiranvand, V.; Hare, W.; Lucet, Y. Best practices for comparing optimization algorithms. *Optim. Eng.* **2017**, *18*, 815–848. [CrossRef]
51. Supercomputer "Bura", University of Rijeka, Center for Advanced Computing and Modelling. Available online: https://cnrm.uniri.hr/bura/ (accessed on 18 February 2024).