

Article

Hybrid Discrete Particle Swarm Optimization Algorithm with Genetic Operators for Target Coverage Problem in Directional Wireless Sensor Networks

Yu-An Fan and Chiu-Kuo Liang * 

Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu 30012, Taiwan

* Correspondence: ckliang@chu.edu.tw

Abstract: For a sensing network comprising multiple directional sensors, maximizing the number of covered targets but minimizing sensor energy use is a challenging problem. Directional sensors that can rotate to modify their sensing directions can be used to increase coverage and decrease the number of activated sensors. Solving this target coverage problem requires creating an optimized schedule where (1) the number of covered targets is maximized and (2) the number of activated directional sensors is minimized. Herein, we used a discrete particle swarm optimization algorithm (DPSO) combined with genetic operators of the genetic algorithm (GA) to compute feasible and quasioptimal schedules for directional sensors and to determine the sensing orientations among the directional sensors. We simulated the hybrid DPSO with GA operators and compared its performance to a conventional greedy algorithm and two evolutionary algorithms, GA and DPSO. Our findings show that the hybrid scheme outperforms the greedy, GA, and DPSO algorithms up to 45%, 5%, and 9%, respectively, in terms of maximization of covered targets and minimization of active sensors under different perspectives. Finally, the simulation results revealed that the hybrid DPSO with GA produced schedules and orientations consistently superior to those produced when only DPSO was used, those produced when only GA was used, and those produced when the conventional greedy algorithm was used.

Keywords: target coverage problem; directional sensor networks; genetic algorithm; discrete particle swarm optimization



Citation: Fan, Y.-A.; Liang, C.-K. Hybrid Discrete Particle Swarm Optimization Algorithm with Genetic Operators for Target Coverage Problem in Directional Wireless Sensor Networks. *Appl. Sci.* **2022**, *12*, 8503. <https://doi.org/10.3390/app12178503>

Academic Editor: Vincent A. Cicirello

Received: 14 July 2022

Accepted: 22 August 2022

Published: 25 August 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

A wireless sensor network (WSN) comprises many tiny sensors that can both acquire data and communicate. These sensors can monitor their environment and transmit the acquired data to other sensors or base stations to coordinate specific tasks [1]. WSNs are widely used in many fields, such as smart homes, military operations, environmental monitoring, animal conservation, medicine, and poaching prevention. Most studies have assumed that sensors are omnidirectional; however, this assumption may not hold for other types of sensors with directional sensing ranges, such as video, ultrasonic, or infrared [2]. Such sensors are called “directional sensors”. The sensing range of an omnidirectional sensor node is a circular disk, whereas that of a directional sensor is a smaller, sector-like sensing area determined by its sensing angle. Research results for omnidirectional sensor networks cannot be directly applied to directional sensor networks (DSNs), which comprise numerous directional sensors. Many challenges specific to DSNs still require investigation.

Several methods of amplifying the sensing capabilities of sensor nodes have been proposed. One involves placing several of the same type of directional sensors on one sensor node, with each sensor facing a different direction. In [3], four pairs of ultrasonic sensors on a single node were used for omnidirectional detection of ultrasonic signals. Another technique is placing the sensor node on a mobile device to enable it to move and change direction. Third, sensor nodes can be equipped with devices that enable them to

switch directions or to rotate. This third technique was adopted in the DSN investigated in this study.

A key challenge in DSN coverage is gathering data from a predetermined area. Coverage problems can be categorized as area coverage or target coverage problems. In area coverage, the purpose is to monitor a predetermined region; that of target coverage is to determine a set of sensors that can provide coverage for a set of targets [2]. Another challenge in DSNs is energy-consumption management; reducing energy usage can extend the network lifetime because sensor nodes are typically battery powered and, thus, can only continuously operate for a limited period of time [4].

In this study, we focused the target coverage problem for a DSN with a set of directional sensors that can rotate to any direction. Specifically, the full coverage of a set of targets in a predetermined region is required, but the limited sensing range of directional sensors prevents the targets from being entirely covered by the deployed sensors. Thus, rotating the directional sensors may be necessary to cover as many targets as possible. To increase the network lifetime, a scheduling technique in which some sensor nodes were active and others were inactivated was used to reduce energy use while maximizing the target coverage. This problem has increased complexity because of its multiple objectives. For such complex problems, the exact solution can be obtained by exhausting a lot of computational effort which increases exponentially with the problem size. The solutions obtained by traditional heuristic algorithms are only near to the optimal; thus, these algorithms should only be used to solve smaller or simple problems. As the problem size increases, the metaheuristic algorithms can be used to improve the solutions because of their high performance in a global search. Many metaheuristic algorithms, such as genetic algorithm (GA) [5–7], particle swarm optimization (PSO) [8], differential evolution (DE) [9], simulated annealing (SA) [10,11], and ant-colony optimization (ACO) [12,13], have been introduced for solving complex optimization problems that cannot be solved using traditional deterministic algorithms in the last two decades and some of them may lead to real improvement in optimization algorithms [14,15]. However, they will easily fall into the local optimal and the cost of computation iteration will increase explosively when the problem scale increases. Therefore, instead of proposing a new metaheuristic algorithm, this paper focuses on presenting a hybrid method for optimization problems. In our previous work [16], we focused on using a genetic algorithm to achieve optimal, or near-optimal, solutions to the target coverage problem. However, the accuracy of the solutions obtained by GA is affected by the selection and mutation operators and the convergence rate is slow. This paper first presents a discrete particle swarm optimization (DPSO) algorithm for the target coverage problem. Particle swarm optimization has gained increasing popularity in solving complex optimization problems due to its simplicity and high convergence speed. This paper also presents a hybrid metaheuristic algorithm based on the DPSO method combined with GA operators, namely DPSO_GA, to solve the target coverage problem. The use of GA operators aims to achieve the goal of intelligent exploration-exploitation. The hybridization of PSO and GA involves using two approaches sequentially or in parallel or using GA operators within the PSO framework [17–23]. In this study, the hybridization consists of a two-phase mechanism where the evolution process is accelerated by using DPSO and diversity is maintained by using GA. Our presented algorithm can achieve more precise solutions than conventional heuristic algorithms can. The main contributions of this paper are as follows:

1. We defined and proved the complexity of the target coverage problem in directional sensor networks with rotatable sensors, and we also mathematically formulated the problem;
2. We proposed a greedy algorithm according to the targets' maximally being covered by a number of directional sensors—the proposed greedy solutions are used as base for comparison;
3. We firstly proposed a metaheuristic algorithm, namely DPSO, for the target coverage problem to determine the schedule of the covered sector for each directional sensor

- with the goal of not only maximizing the total number of covered targets but also minimizing the number of active sensors to conserve energy;
4. We secondly proposed a hybrid metaheuristic algorithm based on the DPSO method and combined with GA operators, namely DPSO_GA, to solve the target coverage problem;
 5. We performed experiments to evaluate the performance of the proposed algorithms, described the results, and compared them with the results of a greedy algorithm;
 6. We discussed the quality of the solutions generated by proposed DPSO and DPSO_GA algorithms along with the solutions generated by previous proposed GA, highlighting the most suitable algorithm for the target coverage problem in various environments.

The remainder of this paper is organized as follows. In Section 2, we present a review of the related work of target coverage problem. In Section 3, we define the target coverage problem in the DSN. Section 4 presents the details of our proposed algorithms. Section 5 presents the experimental results. In Section 6, we present a discussion of the experimental results. Finally, we conclude the paper and propose some future research topics in Section 7.

2. Related Works

The coverage problem is regarded as a key task in both WSNs or DSNs that are intended to collect information from an area of interests or to monitor targets in a pre-determined region. Numerous studies have presented solutions for the coverage problem in omnidirectional WSNs [24–28]. However, the algorithms proposed in these WSNs cannot be directly applied to DSNs because of the limited angle of view of directional sensors. Therefore, new solutions specific to DSNs are necessary. The remainder of this section reviews studies conducted on the target coverage problem in DSNs.

Ma and Liu [29] presented a model of DSNs in which the orientations of the sensor nodes are static. They analyzed the probability of full coverage (i.e., each target is covered by at least one sensor). Ai and Abouzeid [30] presented a model of a sensor network in which the orientations of the sensors' nodes are adjustable. They defined the maximum-coverage-with-minimum-sensor (MCMS) problem, in which the coverage rate is maximized while minimizing the number of active sensors. They also demonstrated that the MCMS problem is NP-complete and presented two greedy algorithms, namely, the centralized greedy algorithm (CGA) and distributed greedy algorithm (DGA) for MCMS problems. Chen et al. [31] developed a weighted centralized greedy algorithm by modifying the CGA with adjustable weight functions to obtain a higher coverage rate. Cai et al. [4] addressed the multiple directional cover set (MDCS) problem of organizing sensor directions into a group of nondisjoint sets to prolong the network lifetime. Only one cover set is activated at a time, in which all targets are covered. They also demonstrated that the MDCS problem is NP-complete and proposed several algorithms for obtaining solutions to this problem. Han et al. [32] addressed the maximum set cover for DSN (MSCD) problems, which are NP-complete, and proposed an algorithm to provide energy-efficient cover sets that could cover all targets. The network lifetime was maximized by activating each cover set for various durations on a schedule. Gil and Han [33] presented a greedy target coverage scheduling algorithm for the MSCD problem and a GA that could identify optimal cover sets that could maximize the network lifetime while monitoring all targets by using the evolutionary global search technique. Li et al. [34] used a bounded service delay constraint to extend the network lifetimes on the target Q-coverage problem in DSNs. They proposed an algorithm to identify a collection of coverage sets that the bounded service delay constraint and the coverage quality requirement are satisfied; the target in each coverage set was not required to be served continuously but could be served with a tolerable service delay. Mohamadi et al. [35] reduced sensor energy consumption by partitioning the DSN into several cover sets, each of which could cover all targets, and activating these covers successively. They presented an irregular cellular learning automata-based distributed algorithm to identify a near-optimal solution for selecting an appropriate working direction for each sensor. Zannat et al. [36] addressed the target coverage problem in a visual sensor network

(VSN) that comprised a number of self-configurable visual sensors with adjustable spherical sectors with limited angles. Razali et al. [37] investigated priority-based target coverage with adjustable sensing ranges. They proposed two scheduling algorithms—greedy-based and learning-automata-based algorithms—to organize the sensors into a few cover sets that were successively activated to maximize network lifetime. The simulation revealed that the learning automata-based scheduling algorithm was superior to the greedy-based algorithm in terms of extending network lifetime. Zishan et al. [38] addressed heterogeneous coverage in VSNs, in which the coverage requirements of targets vary. Their main goal was to maximize the coverage of all targets (on the basis of their coverage requirements) by activating a minimal number of sensors. They solved this heterogeneous coverage problem by modifying the formulation of an existing integer linear programming method for the single and k -coverage MCMS problems. They also presented a sensor-oriented greedy algorithm to obtain an approximate solution of the formulated problem. Bakht et al. [39] successfully addressed the k -coverage problem by determining the orientation of a minimum number of directional sensors in which each target is monitored at least k times, with a learning automata-based algorithm.

The sensor nodes used in these studies comprise a fixed number of sensors with fixed orientations. However, many sensor nodes can rotate. Liang and Chen [40] addressed the maximum coverage with rotatable angles (MCRA) problem; they maximized the coverage rate of targets and minimized the total rotation (in degrees) of the sensors. They took advantage of the ability to adjust the working direction and presented two centralized greedy algorithms for the MCRA problem. In both greedy methods, the weights of sectors were used to select the appropriate working direction to which the sensors should rotate to cover more targets. Wu and Lu [41] redefined the MCMS problem for a DSN comprising rotatable sensors. They proposed a greedy algorithm that used a minimum number of sectors to cover targets within the sensing range of the sensors. Lo and Liang [16] also addressed the MCMS problem in a DSN with rotatable sensors. They proposed a GA for scheduling the orientation of active sensors to obtain a higher coverage rate of targets compared with that of other greedy algorithms.

We considered the target coverage problem with rotatable sensors. The goal aims to determine an appropriate orientation schedule in which each active sensor rotates to the scheduled orientation such that (1) the total number of covered targets is maximized and (2) the total number of active sensors is minimized. We focused on presenting different metaheuristic algorithms, including GA, DPSO, and the hybridization of DPSO and GA, for the problem due to its complexity. The hybridization of PSO and GA approaches has been applied to many applications in the last two decades [17–24]. Robinson et al. [17] proposed a hybrid PSO and GA algorithm by taking the population of one algorithm when the algorithm fails to improve, and using it as the new population of the other algorithm. Shi et al. [18] proposed a hybrid algorithm, namely PSO–GA-series-hybrid evolutionary algorithm (PGSHEA), which is to integrate PSO and GA methods in series. The PSO algorithm is terminated after a certain number of iterations and the best particles are selected and encoded into chromosomes to constitute the population for GA algorithm. The GA algorithm is terminated after a specific number of iterations and the best solutions of GA are transmitted back to the PSO populations. Yang et al. [19] proposed a PSO–GA-based hybrid evolutionary algorithm (HEA) for solving unconstrained and constrained optimization problems. Their proposed evolution strategy is divided into two stages in which the evolution process is accelerated by using PSO and diversity is maintained by using GA. Valdez et al. [20] proposed a fuzzy approach in the PSO–GA hybridization. They used several fuzzy rules to determine whether to consider GA or PSO particles and changes their parameters or to take a decision. Ghamisi and Benediktsson [21] proposed a feature selection method by hybridizing GA and PSO. The proposed method was confirmed to be able to automatically select the most informative features within an acceptable processing time without requiring the users to set the number of desired features beforehand. Moussa and Azar [22] introduced a combination of PSO and GA approach to classify software

modules as fault-prone or not using object-oriented metrics. Nik et al. [23] presented various combinations of the hybrid GA and PSO algorithms to find an optimal arrangement of surveyed pavement inspection units (SIUs) in massive networks.

In a preliminary work [16], we focused on using a GA to achieve optimal, or near-optimal, solutions to the target coverage problem with a small problem size. In this study, we extend the previous work by formulating the target coverage problem with rotatable sensors mathematically; proposing a greedy algorithm and two swarm algorithms (a DPSO and a hybrid DPSO with GA operators, namely DPSO_GA); and conducting experiments to verify the performance of our proposed algorithms compared with that of the greedy algorithm and previous GA.

3. Preliminaries and Problem Statement

3.1. Network Model and Problem Definition

We studied the target coverage problem with the following model. All of the networking nodes are located in an obstacle-free predetermined region, including sensors and targets. $T = \{t_1, t_2, \dots, t_M\}$ is a set of M targets distributed randomly; the position of each is known. $S = \{s_1, s_2, \dots, s_N\}$ is a set of N homogeneous directional sensors with the same field of view θ and the same sensing radius r ; these sensors should be scheduled to cover all targets. For a directional sensor s with field of view θ and sensing radius r located at location (x, y) in the region (Figure 1), the sensing area for sensing orientation α is a sector of a circle with center (x, y) and radius r ; the sector is bounded by two radii between $\alpha - \theta/2$ and $\alpha + \theta/2$. This sector is represented by D_α . Target t is covered by sensor s if its location is within D_α . The directional sensors can rotate to any direction to monitor a specific target.

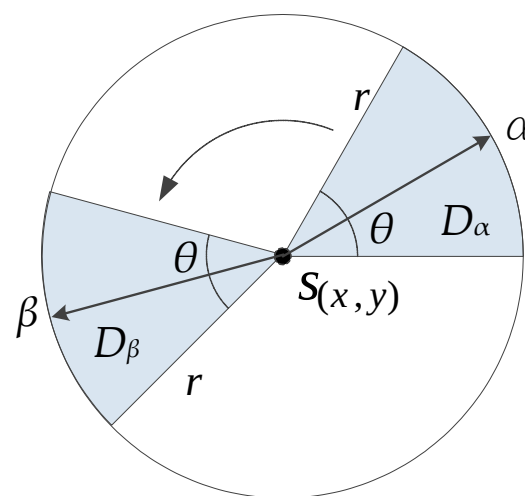


Figure 1. Directional sensing for a rotatable sensor.

Because of the limited field of view of each directional sensor, a target may not be covered by a sensor, even if it is located within the sensor's sensing radius. In this case, the directional sensor can rotate to cover the targets (Figure 2). In Figure 2a, three targets, namely, t_1 , t_2 , and t_3 , are located within the sensing range of deployed sensor s with original orientation α . Targets t_1 and t_2 are covered by the sensor; however, target t_3 can also be covered if sensor s rotates its sensing orientation clockwise to β .

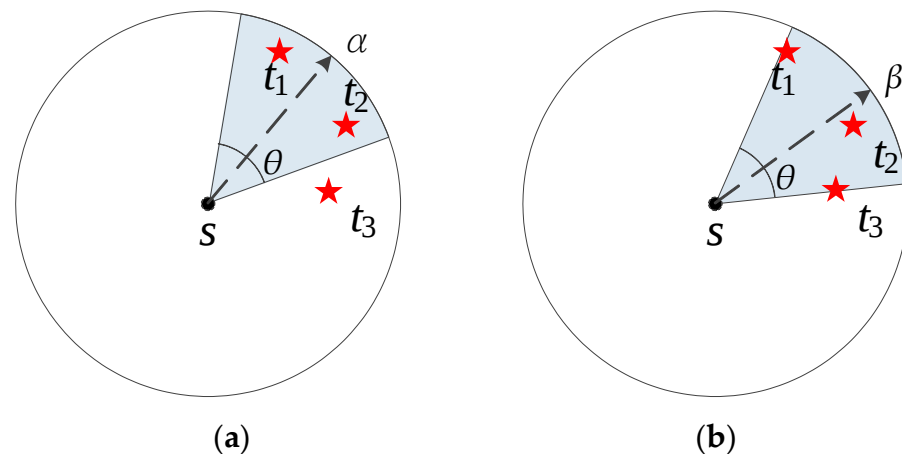


Figure 2. Rotation of a directional sensor s with three targets (indicated as red stars) located inside its sensing range. (a) Target t_3 is not covered by sensor s with sensing orientation α . (b) Target t_3 is covered by sensor s with sensing orientation β .

Definition 1. For sector D_α , $|T(D_\alpha)|$ denotes the number of targets in this sector, where $T(D_\alpha)$ is the set of targets in T covered by sector D_α . Then, D_α is called a cover sector of T if $|T(D_\alpha)| > 0$.

Definition 2. Let D_α and D_β be two different cover sectors of T for directional sensor s . Then, D_α is equivalent to D_β and vice versa if $T(D_\alpha) = T(D_\beta)$. Furthermore, D_α is dominated by D_β or D_β dominates D_α if $T(D_\alpha) \subset T(D_\beta)$.

Definition 3. Let D_α be a cover sector of T for a directional sensor s . Then, D_α is called a maximal cover sector if D_α is not dominated by any other cover sector in sensor s .

Proposition 1. Let k be the number of targets within the sensing radius r of sensor s . Then, at most k different (or nonequivalent) maximal cover sectors exist for sensor s .

For the target coverage problem, after the initial deployment of sensors, not all targets are covered under the random deployment fashion. Our goal is to rotate the initial orientations of the sensor to maximize the number of covered targets while minimizing the number of activated sensors. Therefore, the target coverage problem with rotatable sensors is defined as follows:

Definition 4 . Maximum Coverage with Minimum Rotatable Sensors (MCMRS) Problem: Assume a set of targets $T = \{t_1, t_2, \dots, t_M\}$ to be covered, a set of homogeneous rotatable directional sensors $S = \{s_1, s_2, \dots, s_N\}$ with p_i possible orientations each, and a set of cover sectors $D = \{D_{ij}, 1 \leq i \leq n, 1 \leq j \leq p_i\}$, where D_{ij} is the j th maximal cover sector for sensor s_i , p_i denotes the maximum number of cover sectors for sensor s_i , and $T(D_{ij})$ is a subset of T . Find a subset Z of D with the constraint that at most one D_{ij} can be selected for each i to maximize the cardinality of the union of chosen $\cup_{(i,j)} T(D_{ij})$ while minimizing the cardinality of $Z = \{D_{ij}, (i, j) \text{ chosen}\}$.

Let x_{ij}^k and y_{ij} be binary variables defined as follows:

$$x_{ij}^k = \begin{cases} 1 & \text{if target } t_k \text{ is covered by } D_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if } D_{ij} \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\varphi_k = \begin{cases} 1 & \text{if target } t_k \text{ is covered by any directional sensor,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\delta_k^i = \begin{cases} 1 & \text{if target } t_k \text{ is located within the sensing radius of sensor } i, \\ 0 & \text{otherwise.} \end{cases}$$

The MCMRS problem can be formulated mathematically as follows:

$$\max \sum_{k=1}^m \varphi_k, \quad (1)$$

$$\min \sum_{i=1}^n \sum_{j=1}^{p_i} y_{ij} \quad (2)$$

Subject to:

$$\frac{\sum_{i=1}^n \sum_{j=1}^{p_i} x_{ij}^k \cdot y_{ij}}{n} \leq \varphi_k \leq \sum_{i=1}^n \sum_{j=1}^{p_i} x_{ij}^k \cdot y_{ij} \text{ for every target } t_k \in T, \quad (3)$$

$$\sum_{j=1}^{p_i} y_{ij} \leq 1 \text{ for every sensor } s_i \in S, \quad (4)$$

$$\frac{\sum_{k=1}^m \delta_k^i}{m} \leq p_i \leq \sum_{k=1}^m \delta_k^i \text{ for every sensor } s_i \in S, \quad (5)$$

The objective in (1) represents the number of covered targets to be maximized while the objective in (2) represents the number of activated sensors to be minimized. Equation (3) reveals that the value of φ_k can only be 0 or 1, indicating that each target must be covered or not covered. If target t_k is covered by any directional sensor, then $\varphi_k = 1$ to conform with the left inequality; otherwise, $\varphi_k = 0$ to conform with the right inequality. For each sensor, at most one cover sector that can be activated exists, as indicated by (4), and the number of maximal cover sectors is bounded by the number of targets located within that sensor, as indicated in (5).

3.2. Hardness of the Problem

We demonstrate that the MCMRS problem is NP-complete.

Theorem 1. *The MCMRS problem is NP-complete.*

Proof. Consider the special case of the MCMRS problem where sensors are omnidirectional (i.e., $\theta = 360^\circ$). In this special case, maximizing the number of covered targets for a given k which represents the upper bound of the number of activated sensors can be viewed as a known NP-complete problem, the classic Maximum Coverage Problem [42]. Thus, the result is as follows. $\square$

3.3. Greedy Algorithm

In this subsection, we propose a heuristic algorithm in which, at each step, the available maximal cover sector of an unselected sensor that covers the uncovered targets with greatest importance is identified. In our algorithm, we assign each target a weight indicating its importance. We first consider the maximally covered number (MCN) [31] for each target t_k , where $\text{MCN}(t_k) = |\{D_{ij} | t_k \in T(D_{ij}), 1 \leq i \leq n, 1 \leq j \leq p_i\}|$ indicates the number of maximal sectors that cover t_k . The weight of target t_k is $1/\text{MCN}(t_k)$. Therefore, targets with a smaller MCN value have greater weight (i.e., coverage of targets covered by fewer sectors should be considered more critical). We determine the weight of each maximal cover sector by summing the weights of all targets covered by the maximal cover sector. We then begin scheduling by selecting the sector with the highest weight among all unselected sectors. When a sector is selected, the corresponding sensor is scheduled as active with the orientation of the corresponding selected sector, and all remaining cover sets are eliminated from the unselected sectors for that sensor. The weights of all targets covered by the selected

sector are set to 0, and the weights of the remaining unselected sectors that cover those targets are updated. This scheduling process continues until the set of unselected sectors is empty. The pseudocode of our proposed greedy algorithm is shown in Algorithm 1.

Algorithm 1: Greedy Algorithm (GREEDY)

```

1: Find the set of cover sectors  $D = \{D_{ij}, 1 \leq i \leq N, 1 \leq j \leq p_i\}$ , where  $D_{ij}$  is the  $j$ th maximal
   cover sector of sensor  $i$ . Find the weight of each target  $t_k, 1 \leq k \leq M$ , and the weight of each
   sector in  $D$ .
2:    $Z = \{\}$ 
3:   while ( $D$  is not empty) do
4:     Choose the cover sector in  $D$  with maximum weight, denoted  $D_{nq}$ 
5:      $Z = Z \cup D_{nq}$ 
6:     Delete  $D_{nj}$  from  $D$  where  $1 \leq j \leq p_n$ 
7:     Set the weight all targets covered by  $D_{nq}$  to 0
8:   Update the weight of all sectors in  $D$  that contain the targets covered by  $D_{nq}$ 
9:   end while
10: return  $Z$ 

```

The time complexity of the greedy target coverage algorithm, denoted as GREEDY, is as follows. Determining the MCN values of all targets requires mn steps. The running time for identifying all cover sectors is thus bounded by $O(mn)$. In the worst case, D has $O(mn)$ cover sectors. Therefore, we can identify the cover sector in D with maximum weight in $O(mn \log(mn))$. Because at most n loops are required to select the cover sectors and each update takes at most $O(m \log(mn))$, the time complexity of GREEDY is $O(mn \log(mn))$ in the worst case.

4. Proposed Metaheuristic Schemes

In this section, we propose two metaheuristic algorithms for the MCMRS problem: DPSO and DPSO_GA, to obtain better solutions than that obtained by the greedy algorithm.

4.1. Particle Swarm Optimization (PSO)

Swarm intelligence is an innovative distributed intelligence paradigm for solving optimization problems. The paradigm was inspired by biological phenomena, such as flocking, swarming, and herding in vertebrates. Particle swarm optimization (PSO) incorporates swarming behaviors observed in schools of fish, flocks of birds, swarms of bees, or even human social behavior [43,44]. PSO is a population-based optimization technique that can be easily applied to solve various optimization problems. The major strength of a PSO algorithm is its fast convergence, which has advantages over other global optimization algorithms, such as GAs and simulated annealing.

In PSO, a swarm of particles explores the solution space of an optimization problem to identify an optimal or quasioptimal solution. Each particle represents a candidate solution and is identified with specific position in the D -dimensional search space. The position and velocity of i th particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, respectively. The fitness function is evaluated for each particle in the swarm, and the result is compared with the best previous fitness obtained by that particle and to the best fitness ever achieved by any particles of the swarm. Then, each particle updates its velocity and position in accordance with the following equations:

$$V_i^{t+1} = \omega * V_i^t + c_1 * r_1 * (P_i^t - X_i^t) + c_2 * r_2 * (G^t - X_i^t) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

where $1 \leq i \leq P$; P is the size of the swarm; V_i^t and X_i^t are the velocity and position of the i th particle at iteration t , respectively; P_i^t is the current best solution for the i th particle at iteration t ; G^t is the global best solution at iteration t ; ω is the inertia weight; and c_1

and c_2 are acceleration coefficients. Here, r_1 and r_2 denote uniform random numbers between (0, 1).

4.1.1. Particle Encoding

In target coverage problem, the coverage of targets depends on determining the appropriate sector of each directional sensor. Therefore, each particle in a population can be encoded as the selection of a cover sector to represent a candidate solution to the problem. The length of each particle is the same as the number of directional sensors. Thus, a possible schedule of directional sensors or particle can be expressed as follows:

$$(d_1, d_2, \dots, d_n),$$

where $0 \leq d_i \leq p_i$ represents the selected cover sector of the i th directional sensor. In this representation, $d_i = 0$ indicates that sensor s_i is inactive.

4.1.2. Fitness

The proposed particle encoding representation was evaluated to determine an optimal schedule of directional sensors with a fitness function. To determine the optimal solution for the problem, the evolutionary process of our proposed PSO should achieve two objectives: (1) maximizing the target coverage rate and (2) minimizing the active sensor rate. The second condition is equivalent to maximizing the number of inactive sensors. These two objectives suggest the following. If two schedules of directional sensors or particles have the same total number of covered targets, a chromosome with fewer active sensors is preferable because of its reduced energy consumption. To achieve these two objectives, an appropriate fitness function is proposed which is used in the metasearch algorithm for searching for a schedule that maximizes the fitness function. In general, a schedule of directional sensors with a high fitness value would satisfy all of the objectives well. Therefore, the fitness function for the target coverage problem is defined as follows:

$$F = w \cdot R_T + (1 - w) \cdot (1 - R_S),$$

where R_T is the target coverage rate and R_S is the active sensor rate. The target coverage rate is the ratio of the total number of covered targets to M , and the active sensor rate is the ratio of the total number of active sensors to N . In addition, w is a predefined weight where $0 \leq w \leq 1$.

4.2. DPSO Algorithm

Because the MCMRS problem defined in Equations (1)–(5) is a discrete optimization problem, traditional PSO algorithm cannot be used and, thus, discrete particle swarm optimization algorithm will be developed for the MCMRS problem. Therefore, the evolutionary process of particle velocity and position represented by (6) and (7) in the original PSO algorithm, respectively, must be modified such that they can span the discrete search domain. For our proposed MCMRS problem, we developed a DPSO algorithm which is an adaptation of the method established in [45] for updating the particle positions. According to [45], the position of the i th particle at iteration t can be updated as follows:

$$X_i^{t+1} = c_2 \oplus F_3 \left(c_1 \oplus F_2 \left(\omega \oplus F_1(X_i^t), P_i^t \right), G^t \right) \quad (8)$$

where F_1 , F_2 , and F_3 are operations with the probabilities ω , c_1 , and c_2 , respectively.

In the DPSO algorithm, the particle velocities and positions are updated by three operations. The first operation is $\tau_i^{t+1} = \omega \oplus F_1(X_i^t)$, which represents the velocity operation for a particle and generates a temporary particle τ_i^{t+1} . F_1 denotes the mutation operator, which is applied with a probability of ω . Generate a random number $r \in (0, 1)$. A new particle $\tau_i^{t+1} = F_1(X_i^t)$ is generated by applying the mutation operator to the current particle if $r < \omega$. Otherwise, set the new particle to be the current particle, i.e., $\tau_i^{t+1} = X_i^t$. The second operation is $\delta_i^{t+1} = c_1 \oplus F_2(\tau_i^{t+1}, P_i^t)$, which is the cognitive part of the particle. F_2 represents the crossover operator, which is applied with a probability of c_1 . The outcome is, thus, either $\delta_i^{t+1} = F_2(\tau_i^{t+1}, P_i^t)$ or $\delta_i^{t+1} = \tau_i^{t+1}$, where δ_i^{t+1} is a temporary particle. The third operation is $X_i^{t+1} = c_2 \oplus F_3(\delta_i^{t+1}, G^t)$, which is the social part of the particle. F_3 represents the crossover operator, which is applied with a probability of c_2 . The outcome is either $X_i^{t+1} = F_3(\delta_i^{t+1}, G^t)$ or $X_i^{t+1} = \delta_i^{t+1}$.

Mutation operator F_1 for position X_i^t represents the replacement of the original cover sector for a sensor by a randomly selected cover sector. For example, with $X_i^t = (1, 3, 0, 2, 3, 1, 4, 0, 2)$ and mutations occurring in the second, third, and seventh positions, we might obtain $\tau_i^{t+1} = (1, 2, 1, 2, 3, 1, 3, 0, 2)$. Operator F_2 represents selecting τ_i^{t+1} and P_i^t to be the first and second parents for a crossover operation, respectively. Each of the cover sectors in τ_i^{t+1} and P_i^t are exchanged, from left to right, if a random number $r \in (0, 1) < c_1$. For example, with $\tau_i^{t+1} = (1, 3, 0, 2, 3, 1, 4, 0, 2)$ and $P_i^t = (1, 3, 1, 2, 4, 1, 3, 0, 1)$, we obtain $\delta_i^{t+1} = (1, 3, 1, 2, 4, 1, 3, 0, 1)$ if all generated uniform random numbers are less than c_1 . The crossover operator F_3 represents selecting δ_i^{t+1} and G^t as the first and second parents, respectively, for a crossover operation in a similar manner for that of F_2 ; each of the different cover sectors in δ_i^{t+1} and G^t , from left to right, is exchanged if a random number $r \in (0, 1) < c_2$.

The pseudocode of the procedure of our proposed DPSO algorithm for the MCMRS problem is shown in Algorithm 2.

Algorithm 2: Discrete Particle Swarm Optimization (DPSO) Algorithm

```

1:  Initialize the parameters:  $P$ ,  $\omega$ ,  $c_1$ , and  $c_2$ . Initialize the position and velocity for each particle
    randomly. Set  $t_{\max}$  to be the maximum generations, and set  $t$  to 1
2:  Evaluate the fitness of each particle, and set particle_best to the particle itself
3:  Set global_best to the particle with the highest fitness
4:  while  $t < t_{\max}$  do
5:      for each particle do
6:          Update particle.position and particle.velocity by applying the operations  $F_1$ ,  $F_2$ , and  $F_3$  in (8)
            with their respective probabilities
7:              Compute new fitness
8:              if new fitness > particle_best.fitness
9:                  particle_best.fitness  $\leftarrow$  new fitness
10:                 particle_best  $\leftarrow$  particle
11:             if new fitness > global_best.fitness
12:                 global_best.fitness  $\leftarrow$  new fitness
13:                 global_best  $\leftarrow$  particle
14:             end
15:          $t \leftarrow t + 1$ 
16:     end
17:     return global_best

```

4.3. Hybrid DPSO Algorithm with GA Operators

The proposed hybrid DPSO algorithm with GA operators comprises two stages in each iteration of the evolutionary process. The first stage is the same process as in the DPSO algorithm; the new position and velocity of each particle are generated in accordance with the particle's best solution and the global swarm's best solution. In the second stage, the GA operators, namely, selection, crossover, and mutation operations, are applied to create new particles for the next iteration. We present the GA operators used in the hybrid algorithm in the following subsections.

4.3.1. Selection

The selection operation selects candidate particles, based on their fitness, from the population of the current generation. Roulette-wheel selection was employed in our proposed DPSO_GA scheme. Specifically, a slot on a biased roulette wheel was assigned to each particle with the slot size proportional to its fitness value. As a result, particles with higher fitness values were more likely to be chosen into the crossover operation queue.

4.3.2. Crossover

In the crossover operation, two individuals were selected and used to produce two new individuals by exchanging genes or chromosomes in accordance with a probability denoted the crossover rate. A uniform crossover mask was used to produce two children by exchanging two parent genes from the corresponding mask positions. An example of a crossover operation is presented in Figure 3a:

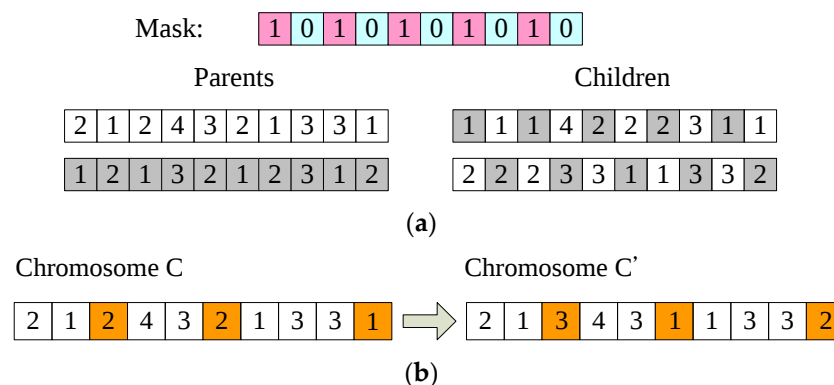


Figure 3. Genetic operators. (a) Crossover operation with mask. (b) Mutation operation (colored box numbers show the mutation of chromosome C to obtain chromosome C').

4.3.3. Mutation

For the MCMRS problem, the mutation operator is applied to each particle or chromosome to prevent premature convergence. It is used on each gene of a particle with a probability. A cover sector of the sensor is randomly selected to be the new value each chosen gene. An example of a mutation operation is presented in Figure 3b.

4.3.4. Pseudocode of the Hybrid DPSO_GA Algorithm

Algorithm 3 shows the pseudocode of the procedure of our hybrid DPSO_GA algorithm for the MCMRS problem.

Algorithm 3: Discrete PSO with GA operators (DPSO_GA) Algorithm

```

1:  Initialize the parameters:  $P$ ,  $\omega$ ,  $c_1$ , and  $c_2$ . Initilize the position and velocity for each particle
    randomly. Set  $t_{\max}$  to be the maximum generations, and set  $t$  to 1
2:  Evaluate the fitness of each particle, and set particle_best to the particle itself
3:  Set global_best to the particle with the highest fitness
4:  while  $t < t_{\max}$  do
5:      for each particle do
6:          Update particle.position and particle.velocity by applying the operations  $F_1$ ,  $F_2$ , and  $F_3$  in (8)
          with their respective probabilities
7:          Compute new fitness
8:          if new fitness > particle_best.fitness
9:              particle_best.fitness  $\leftarrow$  new fitness
10:             particle_best  $\leftarrow$  particle
11:          if new fitness > global_best.fitness
12:              global_best.fitness  $\leftarrow$  new fitness
13:              global_best  $\leftarrow$  particle
14:          end
15:      while crossover_queue_length <  $P$  do
16:          Randomly select two particles and compare their fitness
17:          Add particle with higher fitness into crossover_queue
18:      end
19:      while crossover_queue  $\neq \emptyset$  do
20:          Randomly select and remove two particles from crossover_queue
21:          Crossover the selected particles with a certain probability
22:          Insert these two particles into new population
23:      end
24:      for each particle in new population do
25:          Mutate particle with a certain probability
26:          Compute fitness
27:          particle_best.fitness  $\leftarrow$  fitness
28:          particle_best  $\leftarrow$  particle
29:      end
30:      Descending sort of particles in new population according to their fitness
31:      Select the particles with higher fitness as the swarm
32:      Set global_best to the particle with the best fitness
33:       $t \leftarrow t + 1$ 
34:  end
35:  return global_best

```

5. Performance Evaluation

We evaluated the results obtained with the DPSO and hybrid DPSO_GA algorithms in a set of experiments and compared the results with the results obtained with the greedy algorithm and GA. We performed all experiments with a program in C# on a Windows 10 computer. We investigated the effects of four network parameters on the number of covered targets and the number of activated sensors: the number of sensors N , the number of targets M , the field of view θ , and the sensing radius R . In the experiments, all targets and sensors were randomly scattered in a region of size 800 m $\times$ 800 m. Specifically, for each combination of network parameters, we randomly generated 40 instances of the network and reported the mean results. The simulation parameters are shown in Table 1.

Our simulations focused on determining the algorithm's performance for various network parameters in terms of the total number of covered targets, the total number of activated sensors, and the quality of the solutions. The quality of a solution was represented by its fitness value.

Table 1. Simulation parameters.

Parameter	Value	Meaning
N	50, 75, 100, 125, 150, 175, 200	Number of directional sensors
M	50, 100, 150, 200	Number of targets
R	60, 80, 100, 120	Sensing radius (m)
θ	30°, 60°, 90°, 120°	Field of view
w	0.5	Weight of fitness function
P	100	Population size
P_c	0.8	Probability of GA crossover operator
P_m	0.1	Probability of GA mutation operator
ω	0.1	Probability of DPSO mutation (F_1)
c_1	0.5	Probability of DPSO crossover (F_2)
c_2	0.5	Probability of DPSO crossover (F_3)
I_c	1000	Number of generations or iterations

Figure 4 shows how the number of sensors has an impact on the target coverage rate, active sensor rate, and fitness values under the GA, DPSO, and DPSO_GA algorithms, compared with the solutions obtained by the greedy algorithm. This experiment had 200 targets, and number of sensors N was 50, 75, 100, 125, 150, 175, or 200; the sensor and targets were randomly generated in the specified region. The sensors' coverage radius $R = 80$ m, the field of view $\theta = 60^\circ$, and the weight of the fitness function $w = 0.5$. Figure 4a reveals that the greedy algorithm has at least a 17% higher target coverage rate than do the other three evolutionary algorithms. However, the active sensor rate of the greedy algorithm is also higher than others by at least 15% (Figure 4b). Therefore, in terms of the solution quality, the evolutionary algorithms achieve better performance than does the greedy algorithm by 0.01% to 40.08% (Figure 4c). Furthermore, among the proposed metaheuristic algorithms, the DPSO_GA algorithm outperforms both the GA and DPSO algorithms in terms of target coverage rate and fitness values; however, the DPSO_GA algorithm requires the greatest number of active sensors. The DPSO_GA algorithm outperforms the GA and DPSO algorithms by 0.4% to 2.9% in terms of the fitness value. Figure 4d plots the fitness values versus the number of generations for the GA, DPSO, and DPSO_GA schemes with $N = 100$, $M = 200$, $R = 80$, $\theta = 60^\circ$, $w = 0.5$, and $I_c = 1000$. The DPSO_GA and DPSO approaches had higher fitness than GA did after 10 generations, and the DPSO_GA approach achieved solutions that were 0.42% and 0.82% better than those of DPSO and GA, respectively. Therefore, the proposed hybrid DPSO_GA algorithm can better identify high-fitness particles and transmit them to subsequent generations than the other algorithms. As a result, the hybrid scheme was more likely to achieve the global optimum solutions for the MCMRS problem in terms of the number of sensors.

Figure 5 shows how the number of targets has an impact on the target coverage rate, active sensor rate, and fitness values for the GA, DPSO, and DPSO_GA algorithms compared with the solutions obtained with the greedy algorithm. In the experiment, the number of targets varied from 50 to 200, and the number of sensors $N = 100$. All targets and sensors are randomly generated in the specified region. The sensor coverage radius $R = 80$ m, the field of view $\theta = 60^\circ$, and the weight of fitness function $w = 0.5$. Figure 5a reveals that as the number of targets increases, the number of targets covered increases, but the target coverage rate decreases because increasing the number of targets increases both the number of uncovered and covered targets. The number of uncovered targets increases more quickly; thus, the total target coverage rate decreases. The greedy algorithm achieves 6% to 35% higher target coverage rates than do the other three evolutionary algorithms as the number of targets increases. However, Figure 5b reveals that the active sensor rate of the greedy algorithm is 4–41% higher than the other three algorithms. Therefore, the quality of the solutions of the evolutionary algorithms is higher than that of greedy algorithm for larger numbers of targets (Figure 5c). Moreover, among the proposed metaheuristic algorithms, the DPSO_GA algorithm outperforms the GA and DPSO algorithms in terms of the target coverage rate and the fitness values despite requiring the most active sensors.

The fitness of the DPSO_GA algorithm is 0.17–1.23% higher than that of the GA and DPSO algorithm. Figure 5d plots the fitness values versus the number of generations for the GA, DPSO, and DPSO_GA schemes with $N = 100$, $M = 150$, $R = 80$, $\theta = 60^\circ$, $w = 0.5$, and $I_c = 1000$. The fitness values of the DPSO_GA and DPSO approaches were greater than that of the GA approach after the first three generations, that of the DPSO_GA approach was greater than the DPSO approach after 110 generations, and DPSO_GA obtained solutions that were 0.58% and 1.23% better than those of DPSO and GA, respectively, after 1000 generations. As a result, the hybrid scheme was more likely to achieve the global optimum solutions for the MCMRS problem in terms of the number of targets.

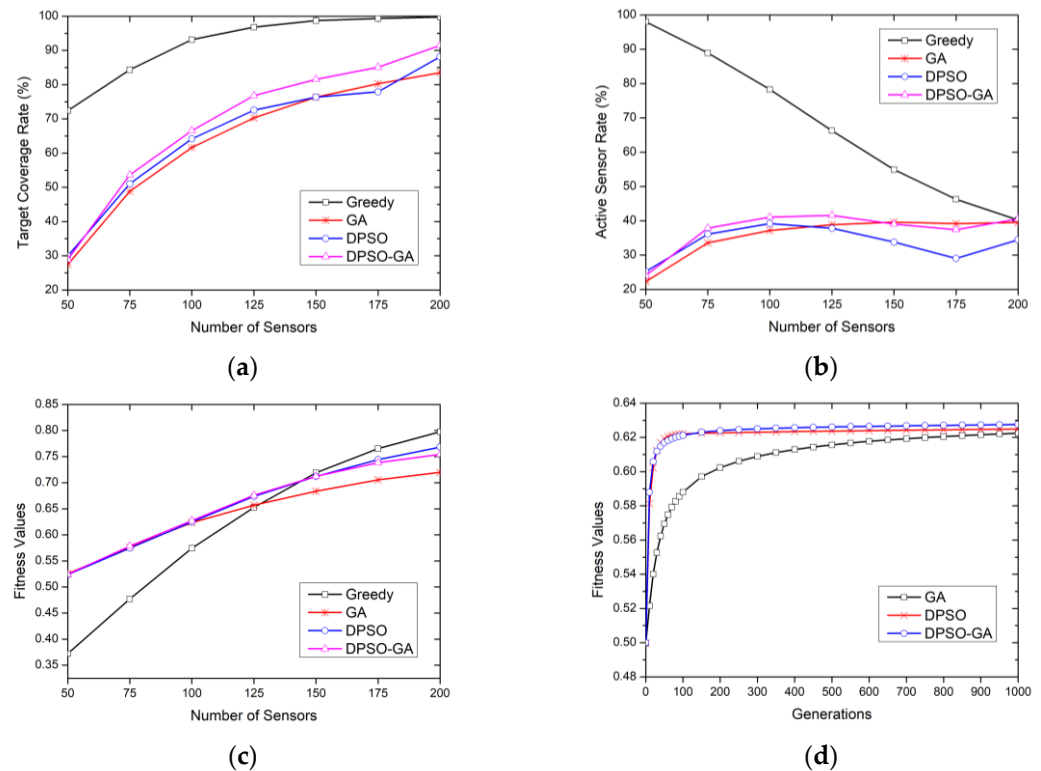


Figure 4. Effect of the number of sensors on the target coverage rate, active sensor rate, and fitness values for the greedy, GA, DPSO, and DPSO_GA algorithms. (a) Target coverage rate, (b) active sensor rate, (c) fitness values, and (d) fitness values versus the number of generations for GA, DPSO, and DPSO_GA.

Figure 6 shows how the field of view has an impact on the target coverage rate, active sensor rate, and fitness values in the GA, DPSO, and DPSO_GA algorithms compared with the solutions obtained by the greedy algorithm. In this experiment, $N = 200$ targets and $M = 100$ sensors were randomly generated in the specified region. The sensors' coverage radius $R = 80$ m and the weight of fitness function $w = 0.5$. The field of view varied from 30° to 120° in increments of 30° . Figure 6a,b reveals that, as in the other experiments, the greedy algorithm achieves a $> 20\%$ higher target coverage rate but requires a $> 25\%$ higher active sensor rate and thus is outperformed by the evolutionary algorithms by 2.53–45.61% (Figure 6c). The DPSO_GA algorithm again outperforms the GA and DPSO algorithms by 4.73% to 8.89% but requires more active sensors. Figure 6d plots the fitness values versus the number of generations for GA, DPSO, and DPSO_GA with $N = 100$, $M = 200$, $R = 80$, $\theta = 90^\circ$, $w = 0.5$, and $I_c = 1000$. DPSO_GA and DPSO outperformed the GA approach after 12 generations, and DPSO_GA ultimately obtained solutions that were 3% and 5% better than those of DPSO and GA, respectively. As a result, the hybrid scheme was more likely to achieve the global optimum solutions for the MCMRS problem in terms of the field of view.

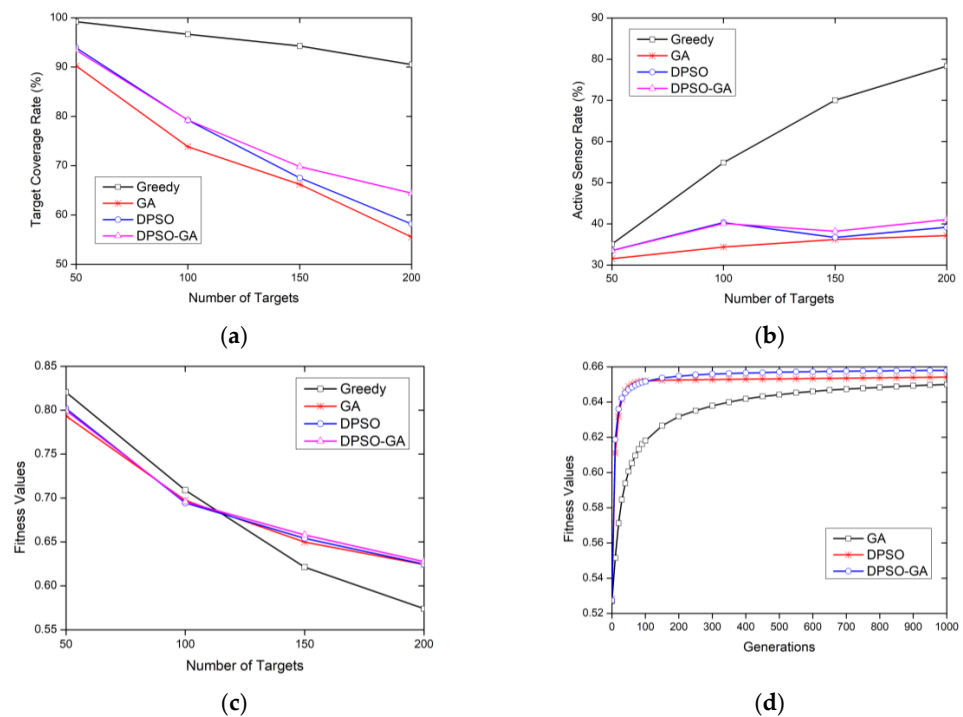


Figure 5. Effect of the number of targets on the target coverage rate, active sensor rate, and fitness values under the greedy, GA, DPSO, and DPSO_GA algorithms. (a) Target coverage rate, (b) active sensor rate, (c) fitness values, and (d) fitness values versus number of generations for the GA, DPSO, and DPSO_GA algorithms.

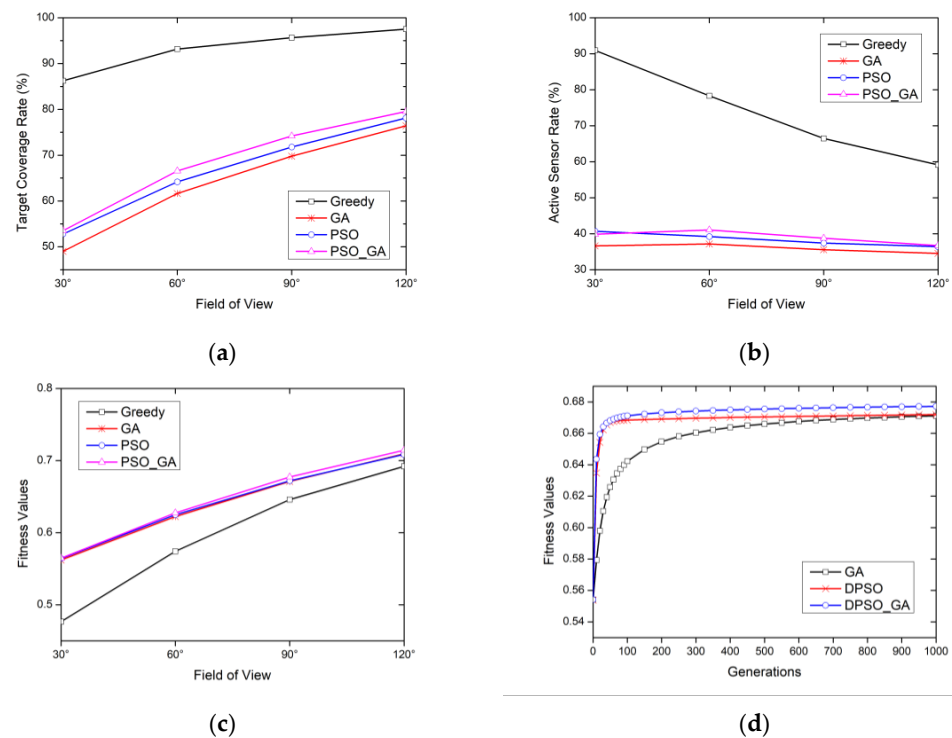


Figure 6. Effect of the field of view on the target coverage rate, active sensor rate, and fitness values in greedy, GA, DPSO, and DPSO_GA algorithms. (a) Target coverage rate, (b) active sensor rate, (c) fitness values, and (d) fitness values versus number of generations for the GA, DPSO, and DPSO_GA algorithms.

Figure 7 presents the effect of the sensing radius on the target coverage rate, active sensor rate, and fitness values for the GA, DPSO, and DPSO_GA algorithms compared with the solutions of the greedy algorithm. In this experiment, $N = 200$ targets and $M = 100$ sensors were randomly generated in the specified region. The field of view $\theta = 60^\circ$ and the weight of fitness function $w = 0.5$. The sensing radius varied from 60 to 120 m in increments of 20 m. Figure 7a reveals that the number of covered targets increases as the sensing radius increases because the cover sector area increases. Again, Figure 7a,b reveals that the greedy algorithm achieved a 23–35% higher target coverage rate but required 19–51% more activated sensors than did the evolutionary algorithms and thus was outperformed by them by 1.2–15% (Figure 7c). Notably, the greedy algorithm outperforms the GA and DPSO algorithms if the sensing radius is 120 m because both the GA and DPSO algorithms become trapped in local optima. However, the proposed DPSO_GA operators' algorithm still outperforms the greedy algorithm. Figure 7d plots the fitness values versus the number of generations for GA, DPSO, and DPSO_GA with $N = 100$, $M = 200$, $R = 100$, $\theta = 60^\circ$, $w = 0.5$, and $I_c = 1000$. DPSO_GA and DPSO had higher fitness than did the GA approach after five generations, and DPSO_GA, ultimately, achieved solutions 1.1% and 3.3% better than those of DPSO and GA, respectively. As a result, the hybrid scheme was more likely to achieve the global optimum solutions for the MCMRS problem in terms of the sensing radius.

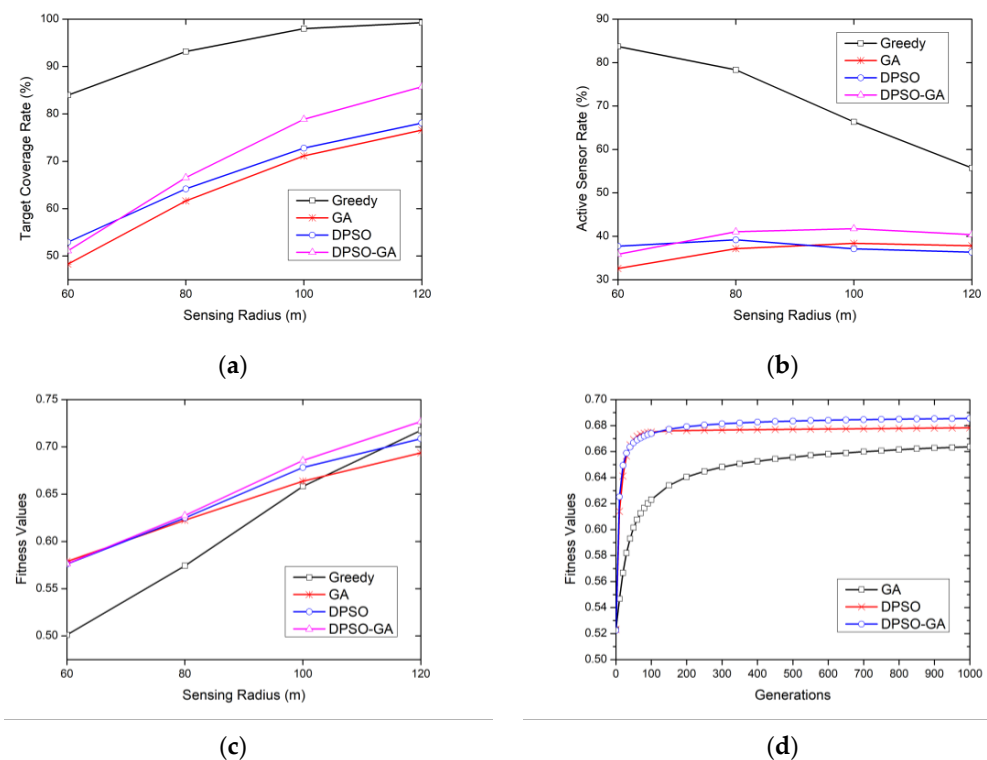


Figure 7. Effect of the sensing radius on the target coverage rate, active sensor rate, and fitness values under greedy, GA, DPSO, and DPSO_GA algorithms. (a) Target coverage rate, (b) active sensor rate, (c) fitness values, and (d) fitness values versus the number of generations for the GA, DPSO, and DPSO_GA algorithms.

6. Discussion

In this study, we presented a DPSO algorithm and a hybrid DPSO with GA operators' algorithm for solving the MCMRS problem in a DSN and compared their performance in terms of solution optimality and system efficiency. Directional sensors are increasingly used in traditional WSNs for target monitoring and tracking. Therefore, determining an effective schedule of active sensors to maximize target coverage rates is a crucial research topic. This

paper provides effective algorithms to obtain high coverage and quality of solutions and contributes to the work of research topics related to the target coverage problem.

Although the proposed metaheuristic algorithms can effectively resolve the target coverage problem, this work has some limitations. (1) All targets were assumed to be static to reduce the complexity of the problem. (2) The coverage requirements of all targets were assumed to be identical; priority of target coverage was not considered. (3) All directional sensors were assumed to have identical parameters. (4) The proposed metaheuristic schemes have disadvantages, such as the premature convergence of the DPSO algorithm, and no guarantee of achieving the optimal solution.

Despite these limitations, the simulation results reveal that these algorithms outperform a conventional heuristic algorithm for solving complex optimization problems, such as the MCMRS problem in a DSN.

Our simulation results also revealed that the hybrid DPSO_GA algorithm outperformed both GA and DPSO algorithms in all of the evaluation metrics, especially for solving the largest and most complex problems. Thus, the hybrid algorithm effectively achieved the advantages of both the GA and DPSO algorithms; combining these algorithms resulted in a new metaheuristic algorithm with practical utility and high performance. Inspired by the satisfactory achievements obtained by the hybrid algorithm, it is worthy of discussing the applicability of the proposed mechanism to related nature-inspired and evolutionary algorithms to solve MCMRS problem. Finally, additional factors affecting the energy consumption of active sensors must be considered in practical applications, such as the energy required to rotate a sensor.

7. Concluding Remarks and Future Work

We investigated the MCMRS problem for a DSN. We required full coverage of a set of targets in a predetermined region. To conserve energy, we determined an arrangement of cover sectors such that the number of covered targets was maximized but the number of active sensors was minimized. Because the MCMRS problem is NP-complete, it will take a long computational time to determine the optimal schedule of active sensors. We employed two nondeterministic algorithms, the DPSO algorithm and the DPSO_GA algorithm, which is a hybrid DPSO scheme with GA operators, to manage this complexity and produce solutions in a relatively short computation time. Our proposed algorithms quickly determined high-quality sensor schedules and could be applied in situations with more numerous targets and sensors. The hybrid DPSO_GA algorithm produced solutions that were consistently superior to those produced by both the GA and the DPSO algorithms.

In the future, our proposed algorithms could be applied for targets with different coverage requirements (i.e., different coverage weights or priorities). It is also worth focusing on combining other metaheuristic methods to solve MCMRS problem. Moreover, energy is required to rotate the sensors in practical applications; this cost of reorientation could be investigated in future research.

Author Contributions: Conceptualization, C.-K.L.; software, Y.-A.F.; investigation, Y.-A.F.; methodology, C.-K.L.; validation, C.-K.L.; writing—original draft preparation, Y.-A.F. and C.-K.L.; and writing—review and editing, C.-K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330. [\[CrossRef\]](#)
2. Amac Guvensan, M.; Yavuz, G. On coverage issues in directional sensor networks: A survey. *Ad Hoc Netw.* **2011**, *9*, 1238–1255. [\[CrossRef\]](#)
3. Djughash, J.; Singh, S.; Kantor, G.; Zhang, W. Range-only slam for robots operating cooperatively with sensor networks. In Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006. [\[CrossRef\]](#)
4. Cai, Y.; Lou, W.; Li, M.; Li, X.Y. Energy efficient target-oriented scheduling in directional sensor networks. *IEEE Trans. Comput.* **2009**, *58*, 1259–1274. [\[CrossRef\]](#)
5. Goldberg, D.E. *Genetic Algorithms*; Pearson Education: New Delhi, India, 2006.
6. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
7. Gen, M.; Cheng, R. *Genetic Algorithms and Engineering Optimization*; Wiley: Toronto, ON, Canada, 2000.
8. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948. [\[CrossRef\]](#)
9. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
10. Chiang, W.C.; Russell, R.A. Simulated annealing meta-heuristics for the vehicle routing problem with time windows. *Ann. Oper. Res.* **1996**, *63*, 3–27. [\[CrossRef\]](#)
11. Tavares, R.S.; Martins, T.C.; Tsuzuki, M.S.G. Simulated annealing with adaptive neighborhood: A case study in off-line robot path planning. *Expert Syst. Appl.* **2011**, *38*, 2951–2965. [\[CrossRef\]](#)
12. Dorigo, M.; Maniezzzi, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [\[CrossRef\]](#)
13. Blum, C. Ant colony optimization: Introduction and recent trends. *Phys. Life Rev.* **2005**, *2*, 353–373. [\[CrossRef\]](#)
14. Jha, S.K.; Eyang, E.M. An energy optimization in wireless sensor networks by using genetic algorithm. *Telecommun. Syst.* **2018**, *67*, 113–121. [\[CrossRef\]](#)
15. Parvin, J.R.; Vasanthanayaki, C. Particle swarm optimization-based energy efficient target tracking in wireless sensor network. *Measurement* **2019**, *147*, 106882. [\[CrossRef\]](#)
16. Lo, Y.S.; Liang, C.K. A genetic algorithm for target coverage problem in directional sensor networks. In Proceedings of the 2nd Eurasian Conference on Educational Innovation (ECEI), Singapore, 27–29 January 2019; pp. 333–336. [\[CrossRef\]](#)
17. Robinson, J.; Sinton, S.; Rahmat-Samli, Y. Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna. In Proceedings of the IEEE Antennas and Propagation Society International Symposium, San Antonio, TX, USA, 16–21 June 2002; pp. 16–21. [\[CrossRef\]](#)
18. Shi, X.H.; Lu, Y.H.; Lee, H.P.; Lin, W.Z.; Liang, Y.C. Hybrid evolutionary algorithms based on PSO and GA. In Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, Australia, 8–12 December 2003; pp. 2393–2399. [\[CrossRef\]](#)
19. Yang, B.; Chen, Y.; Zhao, Z. A hybrid evolutionary algorithm by combination of PSO and GA for unconstrained and constrained optimization problems. In Proceedings of the 2007 IEEE International Conference on Control and Automation, Guangzhou, China, 30 May–1 June 2007; pp. 166–170. [\[CrossRef\]](#)
20. Valdez, F.; Melin, P.; Castillo, O. Evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic for decision making. In Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, Jeju, Korea, 20–24 August 2009; pp. 2114–2119. [\[CrossRef\]](#)
21. Ghamisi, P.; Benediktsson, J.A. Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 309–313. [\[CrossRef\]](#)
22. Moussa, R.; Azar, D. A PSO-GA approach targeting fault-prone software modules. *J. Syst. Softw.* **2017**, *132*, 41–49. [\[CrossRef\]](#)
23. Nik, A.A.; Nejad, F.M.; Zakeri, H. Hybrid PSO and GA approach for optimizing surveyed asphalt pavement inspection units in massive network. *Autom. Constr.* **2016**, *71*, 325–345. [\[CrossRef\]](#)
24. Cardei, M.; Du, D.Z. Improving wireless sensor network lifetime through power aware organization. *Wirel. Netw.* **2005**, *11*, 333–340. [\[CrossRef\]](#)
25. Cardei, M.; Thai, M.T.; Li, Y.; Wu, W. Energy-efficient target coverage in wireless sensor networks. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005. [\[CrossRef\]](#)
26. Zorbas, D.; Glynos, D.; Kotzanikolaou, P.; Douligeris, C. Solving coverage problems in wireless sensor networks using cover sets. *Ad Hoc Netw.* **2010**, *8*, 400–415. [\[CrossRef\]](#)
27. Ting, C.K.; Liao, C.C. A memetic algorithm for extending wireless sensor network lifetime. *Inf. Sci.* **2010**, *180*, 4818–4833. [\[CrossRef\]](#)
28. Mostafaei, H.; Shojafar, M. A new meta-heuristic algorithm for maximizing lifetime of wireless sensor networks. *Wirel. Pers. Commun.* **2015**, *82*, 723–742. [\[CrossRef\]](#)
29. Ma, H.; Liu, Y. On coverage problems of directional sensor networks. In Proceedings of the International Conference on Mobile Ad-Hoc and Sensor Networks (MSN 2005), Wuhan, China, 13–15 December 2005; pp. 721–731. [\[CrossRef\]](#)
30. Ai, J.; Abouzeid, A. Coverage by directional sensors in randomly deployed wireless sensor networks. *J. Comb. Optim.* **2006**, *11*, 21–41. [\[CrossRef\]](#)

31. Chen, U.R.; Chiou, B.S.; Chen, J.M.; Lin, W. An adjustable target coverage method in directional sensor networks. In Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC 08), Yilan, Taiwan, 9–12 December 2008; pp. 174–180. [\[CrossRef\]](#)
32. Han, Y.H.; Kim, C.M.; Gil, J.M. A greedy algorithm for target coverage scheduling in directional sensor networks. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2010**, *9*, 96–106. [\[CrossRef\]](#)
33. Gil, J.M.; Han, Y.H. A target coverage scheduling scheme based on genetic algorithms in directional sensor networks. *Sensors* **2011**, *11*, 1888–1906. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Li, D.; Liu, H.; Lu, X. Target Q-coverage problem with bounded service delay in directional sensor networks. *Int. J. Distrib. Sens. Netw.* **2012**, *2012*, 1–10. [\[CrossRef\]](#)
35. Mohamadi, H.; Ismail, A.S.B.H.; Salleh, S. A learning automata-based algorithm for solving coverage problem in directional sensor networks. *Computing* **2013**, *95*, 1–24. [\[CrossRef\]](#)
36. Zannat, H.; Akter, T.; Tasim, M.; Rahman, A. The coverage problem in visual sensor networks: A target oriented approach. *J. Netw. Comput. Appl.* **2016**, *75*, 1–15. [\[CrossRef\]](#)
37. Razali, M.N.; Salleh, S.; Mohamadi, H. Solving priority-based target coverage problem in directional sensor networks with adjustable sensing ranges. *Wirel. Pers. Commun.* **2017**, *95*, 847–872. [\[CrossRef\]](#)
38. Zishan, A.A.; Karim, I.; Shubha, S.S.; Rahman, A. Maximizing heterogeneous coverage in over and under provisioned visual sensor networks. *J. Netw. Comput. Appl.* **2018**, *124*, 44–62. [\[CrossRef\]](#)
39. Bakht, A.J.; Motameni, H.; Mohamadi, H. A learning automata-based algorithm to solve imbalanced k-coverage in visual sensor networks. *J. Intell. Fuzzy Syst.* **2020**, *39*, 2817–2829. [\[CrossRef\]](#)
40. Liang, C.K.; Chen, Y.T. The target coverage problem in directional sensor networks with rotatable angles. In Proceedings of the International Conference on Advances in Grid and Pervasive Computing (GPC 2011), Oulu, Finland, 11–13 May 2011; pp. 264–273. [\[CrossRef\]](#)
41. Wu, M.C.; Lu, W.F. On target coverage problem of angle rotatable directional sensor networks. In Proceedings of the 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Taiwan, 3–5 July 2013; pp. 605–610. [\[CrossRef\]](#)
42. Hochbaum, D.S. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In *Approximation Algorithms for NP-hard Problems*; Hochbaum, D.S., Ed.; PWS Publishing Co.: Boston, MA, USA, 1997; pp. 94–143.
43. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [\[CrossRef\]](#)
44. Van den Bergh, F.; Engelbrecht, A.P. A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 225–239. [\[CrossRef\]](#)
45. Pan, Q.; Tasgetiren, M.F.; Liang, Y.C. A discrete particle swarm optimization algorithm for the permutation flowshop sequencing problem with makespan criterion. In *Research and Development in Intelligent Systems XXIII, SGAI 2006*; Bramer, M., Coenen, F., Tuson, A., Eds.; Springer: London, UK, 2006; pp. 19–31. [\[CrossRef\]](#)