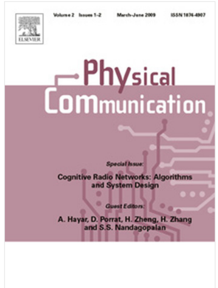
Journal Pre-proof



A genetic algorithm-based approach for solving the target Q-coverage problem in over and under provisioned directional sensor networks

Nemat allah Mottaki, Homayun Motameni, Hosein Mohamadi

PII: S1874-4907(22)00066-0
DOI: https://doi.org/10.1016/j.phycom.2022.101719
Reference: PHYCOM 101719

To appear in: *Physical Communication*

Received date : 20 July 2021
Revised date : 18 February 2022
Accepted date : 6 April 2022

Please cite this article as: N.a. Mottaki, H. Motameni and H. Mohamadi, A genetic algorithm-based approach for solving the target Q-coverage problem in over and under provisioned directional sensor networks, *Physical Communication* (2022), doi: https://doi.org/10.1016/j.phycom.2022.101719.

A genetic algorithm-based approach for solving the target Q-coverage problem in over and under provisioned directional sensor networks

Nemat allah Mottaki[a], Homayun Motameni[b,*], Hosein Mohamadi[c]

[a]*Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran*
[b]*Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran*
[c]*Department of Computer Engineering, Azadshahr Branch, Islamic Azad University, Azadshahr, Iran*

Abstract

Target coverage and network lifetime extension have been addressed as two major research topics over the last two decades. This paper focuses on "target Q-Coverage" in Directional Sensor Networks (DSNs) where coverage requirement of each target in the environment differs from that of the others. In such network, how to achieve the coverage requirement and simultaneously prolong the network lifetime is a major problem. In this study, two target-oriented genetic-based algorithms were developed to solve the problem. The first algorithm was developed to cover the targets in an over-provisioned environment, and the second algorithm was developed in an under-provisioned environment. The main objective of the first algorithm is satisfying the coverage requirement of targets by activating minimal sensors, while the second algorithm was developed to achieve a maximum balanced coverage for all the targets in the network. To evaluate the performance of the developed algorithms, they were compared with some state-of-the-art algorithms presented in recent studies. In this regard, several parameters, including Distance Index, Q-Balancing Index, Coverage Quality, Power Consumption, and Activate Sensors were taken into account. The comparative results indicated that the developed algorithms performed efficiently in solving the Q-coverage problem in both environments.

Keywords: Directional sensors, Q-coverage, Genetic Algorithm (GA)

*Corresponding author. E-mail address: motameni@iausari.ac.ir.

1. Introduction

Wireless sensor networks (WSNs) are composed of low-energy, small-sized sensors powered with batteries; they are deployed over a geographical area to monitor physical phenomena [1]. The sensor nodes are mainly capable of sensing the environment, collecting and processing data, and transferring them to a defined sink [2]. Over the past few years, sensor networks have been applied to various applications such as cloud computing, Internet of Things, and home automation [3]. Regarding the application and type of such networks, it is often impossible to recharge or replace the batteries of the sensors. Therefore, the management of energy consumption in sensors is considered a major challenge in this field [5]. In terms of application, sensors are categorized into omnidirectional and directional sensors. In the former, sensors have a 360-degree sensing angle, while in the latter, the sensors have a restricted sensing angle and the area is observed by a sector of a sensor. The directional sensors are able to sense in several directions; however, they can be activated in just one direction and at a given direction within a unit of time. In fact, they are regarded as a kind of ultrasonic, infrared, and video sensors [5].

1.1. Coverage and network lifetime in DSN

Coverage is considered as a major problem in (DSNs) [6], indicates how each target in an area is observed by sensors [7]. The quality of service is a parameter that is measured by the concept of coverage, and coverage is regarded as a criterion for measuring the service quality [8]. The situation in which the sensors monitor the given target(s) in the environment is called target coverage that is divided into three groups, including simple coverage, K-coverage, and Q-coverage. If each target in the environment is covered by just one sensor, the network is referred to as simple coverage. The K-coverage refers to a situation in which all targets in a network are of equal importance, and each target needs to be covered by at least k sensors. In contrast, the Q-coverage refers to the environments where the targets are not of equal importance, and each target requires a different number of sensors to be fully covered.
Moreover, the network lifetime has considerable importance in DSNs [9, 10]. The network lifetime is defined as the amount of time that the network may operate properly, i.e., the time period from the network start-up until it can no longer accomplish the coverage objectives [11]. To manage energy consumption in sensor networks and to extend the network lifetime, a coverage optimization protocol might be utilized to remove the sensors with common coverage. In this vein,

different researchers have proposed different methods, one of which is sensor scheduling [12]. According to this method, the sensors are divided into a number of cover sets each of which must be able to satisfy the whole network coverage requirement by itself. To conserve energy, the other sensors of the remained cover sets switch to inactive mode when a cover set is active. The method in which each sensor can be the member of only one cover set is called disjoint and, on the other hand, the non-disjoint method is referred to the situations in which the sensors are shared among more than one cover set.

1.2. Categories of environments

DSNs are typically used in perilous and remote environments, where sensors may fail to operate properly and be removed from the network. In this paper, with the aim of evaluating the network performance under various conditions, the performance of DSNs in both over-provisioned and under-provisioned environments is investigated. In over-provisioned networks, sufficient sensors are provided for satisfying the coverage requirements of all targets. On the other hand, under-provisioned networks lack sufficient sensors for providing the coverage requirements of all targets. Some reasons cause an over-provisioned network to become under-provisioned, including the battery discharge of some sensors, increasing the number of targets without any increase in the number of sensors, and loss of sensors due to natural factors. In such networks, the targets are prioritized and higher priority targets will be covered by more sensors, or it is intended to provide a balanced coverage for targets with the same coverage requirements [13, 14, 15]. Accordingly, this research requires to investigate target coverage in both types of environments.

1.3. Necessity of Q-coverage in DSN

For most applications in real world, the sensors are deployed in an environment at random, and unpleasant conditions may occur for them, such as placing an obstacle between the target and the sensor, moving with animals, falling into water, etc. Consequently, coverage needs to be incorporated by the concept of fault tolerance [14]. Fault tolerance coverage is used to ensure that the required coverage of all targets is satisfied even in case some sensors are discarded from the network. One of its solutions is to cover each target with more than one sensor. Accordingly, K-coverage and Q-coverage methods were proposed to solve the problem.

Another reason that leads us to the Q-coverage scenario is that sensor networks are used in military environments, industrial plants, refineries, etc., where the targets are not of equal importance. For example, in a refinery, the temperature of the heat tank is more important than the temperature of the staff rest room and needs to be covered with more sensors. On the other hand, covering all targets with the same number of sensors is not cost-effective due to network overhead and additional costs; thus, different targets need to be covered with different numbers of sensors. All of these reasons motivated us to consider covering targets with different requirements and to study *Q*-coverage in DSNs.

1.4. Our contribution and organization of this paper

Solving the *Q*-coverage problem is great importance in DSNs. Although the problem has been discussed in traditional sensor networks by a number of researchers, it has not been addressed much in DSNs; thus, it is considered to be still an appealing field for research.

In this study two GA-based algorithms proposed to solve the target *Q*-coverage problem in environments. A model was suggested for the chromosome, which could provide all coverage requirements for the objectives defined in an over-provisioned environment, and the maximum balanced coverage in an under-provisioned one. The *QBI* metric was developed to solve the problem in situations where targets have different coverage requirements. In addition, a repair operator was developed to ensure that the child chromosomes are valid after crossover and mutation operators. Finally, using a greedy algorithm, the effectiveness of the developed algorithms is evaluated.

The remainder of this paper is structured as follows: Section 2 briefly reviews the work already conducted to solve the coverage problem and extend the network lifetime. Section 3 describes the network model, problem formulation and performance metrics. Section 4 presents the developed algorithms with an overview of Genetic Algorithm (GA). Section 5 compares the developed algorithms with the recently proposed greedy based algorithms in order to evaluate their performance. After all, Section 6 provides the conclusion and further work.

2. Related Work

Several methods have been proposed to solve the coverage problem and extend the network lifetime. Among all, sensor scheduling is regarded as a common method for maintaining the coverage and managing the energy consumption of sensors. According to this method, the sensors participate in a number of cover sets each of which is activated for a certain period of time. As reported by previously-conducted studies, the existing algorithms for scheduling the sensors differ only in how the sensors are selected to create the cover sets [16].

2.1. Sensor-oriented and target-oriented approaches

Different methods have been used to solve the coverage problem: sensor-oriented and target-oriented methods [17]. In the first method, at each step, the sectors that can cover the maximum number of uncovered targets are selected. According to this method, the overlapping regions of the adjacent cameras must also be determined to eliminate the redundant coverage. While repeating the algorithm at each step, the selection process is done just for the sectors of those sensors that can cover the maximum number of targets. However, one of the drawbacks of such methods can be the fact that at each stage, only a series of sensors may be selected. In the second method, at first, special attention is paid to the targets.
In DSNs, when targets and sensors are randomly distributed, there may be targets covered by a smaller number of sensors. These targets are called critical targets [19]. In an over-provisioned environment, in every cover set, all targets must be covered and their coverage requirements must be fully satisfied. Therefore, identifying and prioritizing the critical targets and selecting them when creating cover sets will lead to an increase in the quantity of cover sets and an extended network lifetime. By definition, the network lifetime is the length of time during which a network works properly [11]. Critical targets play an important role in prolonging the network lifetime and these targets need to be covered first.
The difference between the sensor-oriented and target-oriented methods in solving the coverage problem is further illustrated by an example. As represented by Fig. 1 it is assumed that the sensors (s_1, s_2, s_3, s_4) and the targets (t_1, t_2, t_3, t_4, t_5) are deployed in the environment. Based on the sensor-oriented method in Fig. 1.b, sensor s_2 can cover three targets t_4, t_3, t_2 with one of its sectors. Then, s_4 can cover targets t_3 and t_4 with one of its sectors, but since these targets are already covered by sensor s_2, then s_4 is not required to be active. Sensor s_3 is activated to cover target t_5 and the sensor cannot cover critical target t_1. The Fig. 1.a, is based on the target-oriented method. According to this approach, critical target t_1 is initially covered with one of the sectors of sensor s_2 due to the priority of the coverage of critical targets. Target t_2 is covered by sensor s_1, and targets t_3 and t_4 are covered by s_4, and finally target t_5 is covered by sensor s_3. All the targets will be covered by using such an approach.

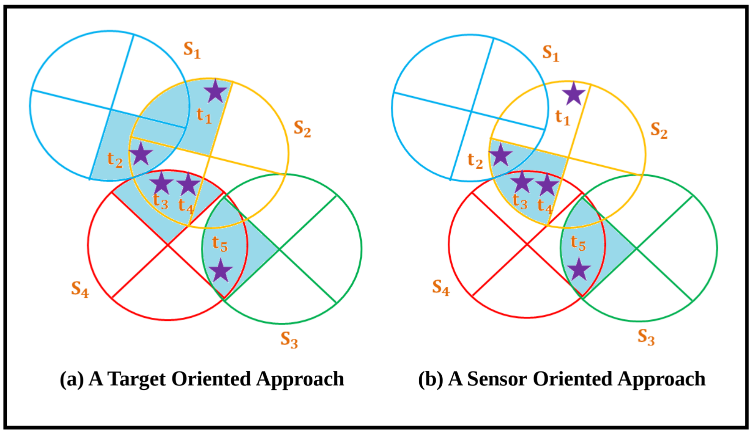


Figure 1: The superiority of target-oriented method over sensor-oriented method in solving the coverage problem

2.2. Related studies

One of the objectives of this study is to investigate the methods proposed to solve the target coverage in DSNs. Based on the number of sensors required for target coverage, the target coverage problem in DSNs can be divided into three general categories: simple coverage, *K*-coverage, and *Q*-coverage.

In simple coverage category, the authors in [18] are the pioneers in studying the coverage problem in DSNs. First, they developed a model called the maximum coverage with the minimum sensors, in which it is intended to provide the maximum coverage for targets with the least number of active sensors. The problem was then formulated with Integer Linear Programming (ILP), and two centralized and distributed greedy algorithms were proposed to solve the problem, and the algorithms were proved to be computationally efficient. In [19, 20], to further extend the network lifetime, two methods of sensor scheduling and adjusting sensing ranges were used. In [19], two greedy algorithms were proposed to create cover sets. The algorithms were implemented in several steps regarding the problem conditions, and a cover set was created at each step. The authors also paid special attention to the critical targets (targets covered by sensors that consume less energy) in the proposed algorithms.

In [20], the Maximum Network Lifetime with Adjustable Ranges (MNLAR) problem was investigated, in which the sensors are selected based on the appropriate direction and sensing ranges in order to create a cover set. The best choice is made depending on the intended target is in which sensing range and in which sector. In [21], the authors used an NSGAII-based algorithm to solve the problem of sensor scheduling and target coverage, with the idea that evolutionary algorithms are suitable to multi-objective optimization in the field of WSNs. To solve the problem, two parameters, the number of sensors and the target coverage, are considered. In general, in multi-objective optimization problems, the objectives must be optimized simultaneously, because they may be in conflict with each other; if only one objective is optimized, the other objectives may be weakened and, consequently, the overall quality of the solution may be compromised.

There are several studies conducted on K-coverage category, in [16], an energy-efficient heuristic method was proposed to solve the K-coverage problem in the homogeneous sensor networks. The proposed method is mainly focused on the battery lifetime of the sensor and avoiding the activation of redundant sensors. Due to the fact that battery lifetime is limited and the critical targets need more management, the method prevents unnecessary consumption of the sensor energy, thereby increasing the lifetime of the network. Moreover, researchers have been increasingly attracted to solve the K-coverage problem in DSNs. In [22], the authors studied the K-coverage problem in DSNs with adjustable sensing range. To this end, two learning automata-based algorithms were proposed, which were equipped with pruning rules to achieve the most optimal solution. The pruning rules prevent the addition of more than one direction of each sensor to a cover set, the presence of multiple sensing ranges of a sensor in a cover set, and the choice of additional directions of the sensors.

Some studies have been recently conducted to investigate K-coverage in the visual sensor networks in under-provisioned state. In [14], some methods were provided to balance the network coverage when the directional sensor nodes are not sufficient and each target requires at least k sensors for coverage. The authors primarily expanded the existing ILP for simple coverage in the K-coverage mode to balance coverage in the network. In addition, the Integer Quadratic Programming (IQP) and the Integer Non-Linear Programming (INLP) methods were proposed. Then, due to high time complexity of these methods in large-scale networks, the Centralized Greedy K-coverage Algorithm (CGKCA) was also proposed. According to the latter method, the Balancing Index (BI) was introduced to achieve a balanced coverage for targets in the network. The authors in [23] presented a learning automata-based algorithm to solve K-coverage in under-provisioned networks. They used the BI relation to achieve a balanced coverage for targets in the network. The algorithm was intended to select the minimal number of sensors in each cover set while maintaining the coverage balance for all targets. To this end, the algorithm was further equipped with pruning rules to increase the algorithm speed by eliminating the incorrect choices.

In [15], two genetic-based algorithms were proposed to solve the K-coverage problem in DSNs. The authors considered both over-provisioned and under-provisioned environments for the problem. In the over-provisioned environment, the algorithm satisfies the coverage requirements of the targets with the least number of sensors. On the other hand, in the under-provisioned environment, based on the Balancing Index (BI), it attempts to provide a maximum balanced coverage for the targets. Compared to [15], in a more general view, the present paper considers the target Q-coverage problem. In addition, BI was developed in a way to be applicable to the Q-coverage problem. To evaluate the performance of the proposed algorithm, different parameters such as Distance Index, Q-Balancing Index, Coverage Quality, and Power Consumption were taken into consideration in the simulations executed in this study.

The last category is Q-coverage. There are several studies conducted on Q-coverage in WSNs. In [8], authors examined Q-coverage in sensor networks in which the energy consumption of each sensor depends on the number of targets covered by that sensor. They proved that the problem was NP-complete and then they formulated the optimization problem by applying the linear programming techniques. With regard to the combinational complexity of such optimization (which makes it more elaborated), a column generation-based approach was proposed, which broke down the original formula into sub-original formulations and solved them alternately. In [24], after proving that Q-coverage problem is NP-complete, a greedy algorithm was proposed to produce non-disjoint cover sets based on the priority of the sensors in terms of the remained battery lifetime. In [25], first, the NP-Hardness of the problem was proved; then, a greedy algorithm was proposed, in which, at each step of implementing the algorithm, the sensors are selected for membership in cover set based on the maximum coverage of the targets that are not yet covered. If two sensors are similar in this property, the selection process is done based on the highest residual energy level of the sensors. In [26], a GA-based algorithm was proposed to solve the Q-coverage problem, in which each target may be covered by one or more sensors, and the sensors are divided into several cover sets. The algorithm is mainly designed to maximize the network lifetime while satisfying the coverage limitations. The proposed algorithm also avoids redundancy and uses the available resources to maximize the network lifetime. In [27], three various versions of Q-coverage were introduced and they were provided by a column generation solution consisting of a master problem and auxiliary problem, which are repeated alternately to get the optimal solution. There is not much research conducted on Q-coverage despite its importance in DSNs. In [28], the heterogeneous coverage in visual sensor networks was investigated, where each target has a different predetermined coverage requirement. In their research, the solution previously proposed by other researchers along with the ILP for simple problem and K-coverage was changed to solve Q-coverage problem. The authors proposed an IQP formula to minimize the Euclidean distance between the obtained and required vectors. They presented the prioritized IQP and reduced-variance IQP for situations where there are not enough sensors to cover. They succeeded to solve the prioritized and group wise balanced coverage by using it. At last, considering that both ILP and IQP are inefficient to solve large-scale problems due to their computational complexity, a Sensor Oriented Greedy Algorithm (SOGA) was used to solve the problem with several different problem evaluation functions.

The algorithm proposed in the present paper has some advantages that are explained as follow. The authors in [28] applied the greedy-based algorithm that may be trapped in the local optimization and may not achieve the global optimization. In this regard, the present paper uses GA-based to solve the problems. The superiority of GA over other models is that multiple-solution search technique is used and it can achieve high-quality optimal or near-optimal solutions during a reasonable time [29]. Compared to [28], the current paper uses the target-oriented method, according to which, first, the targets are searched instead of the sensors, and special attention is paid to the critical targets when creating the cover sets.

3. Network model, problem formulation and performance metrics

We define our DSN model and describe the target Q-coverage problem that we solve in this study. Then, a method is described to determine that target t_{k} is covered by sensor s_{i} and sector j. The notations are defined in the relevant sections, listed in the Table 1 for simplicity.

3.1. Network model

This section examines a DSN that is consisted of m targets $T = \{t_1, t_2, \ldots\ t_{\mathrm{m}}\}$ and n directional sensors $S = \{s_1, s_2, \ldots\ s_{\mathrm{n}}\}$ to cover the targets. All the targets are put at known locations within a 2-D Euclidean plane, and the directional sensors are distributed in a random way in the vicinity of the targets. The sensors in the model presented in this study are directional in their field of view (FoV) that is generally defined as the extent of the observable/sensing area that can be covered

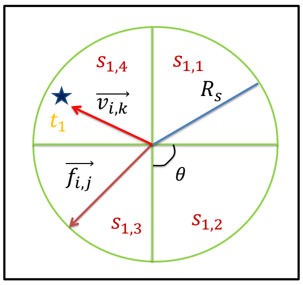


Figure 2: Directional sensor and its coverage parameters in DSN

at any given direction, or the sector of observable area at any given direction by the sensors. FoV is in fact determined by its sector. The sensing area or sector of a sensor is presented as a sector within a circle lying on a 2-dimensional plane (see Fig. 2). Normally, the sector of a sensor is described using two parameters, i.e., θ and R_s. In the following, the parameters that are used to the formalization of various aspects of a sensor node are described (see also Fig. 2):

- (x_i, y_i): This parameter stands for the position of sensor s_i within the Cartesian coordinate system.
- θ: Angle of View (AoV), that refers to the maximum sensing/coverage angle of a sensor in any direction.
- R_s: This parameter stands for a sensor's maximum coverage range beyond which it the sensor is not able to detect any target.
- $\overrightarrow{f_{i,j}}$: It represents a unit vector passing through the middle of a sector, which denotes the sensor s_i orientation toward sector d_j.
- $\overrightarrow{v_{i,k}}$: This parameter signifies a vector in the direction from sensor s_i toward target t_k.

The assumptions considered in the proposed model are based on homogeneity of the sensors. Each target in this model has a different and predetermined coverage requirement, which equals to 1, 2, 3, and 4. In addition, it is assumed that each sensor has finite orientations or sectors and the sectors are disjoint, which are represented as w. For instance, in Fig. 2, a sensor that holds $\frac{\pi}{2}$ of FoV is able to select one out of the four mutually non-overlapping sectors. If all the sectors

are combined, a complete circular view of a sensor's entire sensing region would be generated.
The user of the proposed model is able to verify the cover ability of a target t_k by a given sensor s_i. Furthermore, they can explore the corresponding sector to which the target belongs. The steps for sensor s_i and target t_k can be executed as described below [30]:

1. The angle $\psi_{\mathrm{i,t}}$ between sensor orientation $\overrightarrow{f_{\mathrm{i,j}}}$ of sector d_j and the target vector $\overrightarrow{v_{\mathrm{i,k}}}$ should be computed first (see Eq. (1)).

$$\psi_{it} = \cos^{-1}\left(\frac{\overrightarrow{v_{it}} \cdot \overrightarrow{f_{ij}}}{|\overrightarrow{v_{it}}||\overrightarrow{f_{ij}}|}\right) \tag{1}$$

2. After that, it should be checked whether the target vector $\overrightarrow{v_{\mathrm{i,k}}}$ falls in the FoV of the camera s_i through checking the constraint, $\psi_{\mathrm{i,t}} \leq \frac{\theta}{2}$ or through the use of the inner product.
3. At the final step, target $\mathrm{t_k}$ should be checked to verify whether it is positioned in the sensing range of the sensor through checking $|\overrightarrow{v_{\mathrm{i,k}}}| \leq R_\mathrm{s}$.

In case a target could pass all the three steps mentioned above, then we can say that sensor s_i covers the target t_k within sector d_j. Through applying the above steps to every sector d_j of sensor s_i and every target t_k, a subset of targets can be created for each sensor s_i, which consists of the targets that can be monitored by sensor s_i within sector d_j. It is recognized that a negligible amount of energy is consumed to change the sector, which is overlooked in the calculations. All sensors fall into one of the three modes: active, idle and passive. The Table 1 introduces the notations used in this paper.

3.2. Problem formulation

In the following, a formal definition of the problems is presented.

Given:

1. A set of targets to be covered, $T = \{t_1, t_2, ...t_\mathrm{m}\}$.
2. A set of sensors, $S = \{\mathrm{s}_1, \mathrm{s}_2, ...\mathrm{s}_\mathrm{n}\}$, each of which can be oriented in one of w possible disjoint sectors.
3. A set of positive integers, K=$\{k_1, k_2, ... k_\mathrm{m}\}$, where k_i is the required coverage of target t_i for all targets in T.

Table 1: List of Notations

Notation	Definition
n	Number of sensors
m	Number of targets
w	Number of sectors per sensors
s_i	The ith sensor $1 \leq i \leq n$
t_k	The kth target $1 \leq \mathrm{k} \leq \mathrm{m}$
S	Set of sensors $\{s_1, s_2, \ldots\ s_n\}$
T	Set of targets $\{t_1, t_2, \ldots\ t_m\}$
$d_{i,j}$	jth sector of ith sensor
D	Set of the sectors of all the sensors $D = \{d_{i,j} \mid i = 1, \ldots, n\ , j = 1, \ldots, w\}$
k_t	Total coverage required by the target t
φ_k	Total coverage achived by the target t

4. A set of disjoint (or non-overlapping) sectors, $D = \{d_{\mathrm{i,j}} | (i = 1, \ldots, n), (j = 1, \ldots, w)\}$, a set of all possible sectors.

Problem 1: The problem addressed in the current paper is how to create a cover set in an over-provisioned system with the sectors in D (find a subset C of D) so that the coverage requirement of all the targets in T could be met and, at the same time, the total number of active sensors could be minimized.

Problem 2: How to create a cover set within an over-provisioned environment, with the sectors in D (find a subset C of D) so that the balanced coverage in the network could be maximized.

3.3. Discussion on problem statement

We can solve a problem instance only if there is at least one set of sector (coverage set) to meet all the coverage requirements. Both types of sensor environments (i.e., under-provisioned and over-provisioned) can be redefined regarding the parameter of solvability. For instance, in Fig. 3, there are four sensors and three targets, (t_1, t_2, t_3) with coverage requirements of 2, 1, and 1, respectively. This problem is solvable; a potential solution is cover by $((s_{2,2},\ s_{1,1}),\ s_{4,3},\ s_{3,1})$. Let's look at Fig. 3 again, if target coverage requirement changed to 2, 2, and 2; the problem is unsolvable. In case the coverage requirements remain unsatisfied, two options are available: either achieving a prioritized coverage, or achieving a

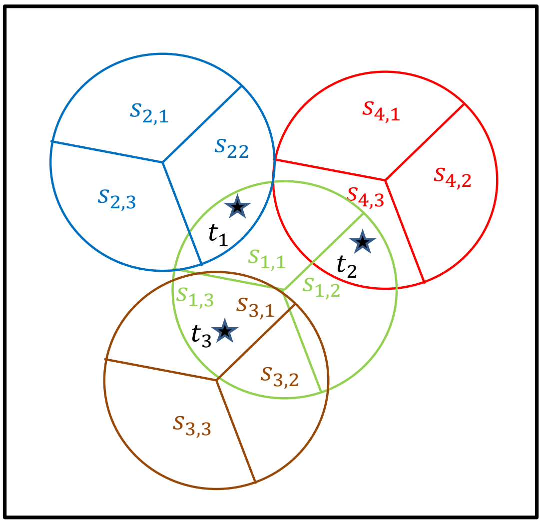


Figure 3: Simple example of unsolvable and solvable scenario

group-wise balanced coverage [28]. In the former option, the targets could be set with priority; maximum coverage should be provided to the targets with higher priorities; however, for the other targets, the coverage requirement may remain unsatisfied. In general, the targets with higher coverage requirement are of higher importance. In the group-wise balanced coverage, a set of sensors can be activated, which can minimize the variances of achieved coverage in each coverage group.

3.4. Performance metrics

In this paper, five metrics (which are defined below) are used to compare the performance of all approaches.

3.4.1. Distance Index (DI):

For the purpose of the current study, the *DI* metric presented in [28] was used. In Eq. (2) , k_t is coverage required for target t, φ_t is coverage achieved for the target t. The higher value of this metric indicates a better coverage for the network.

$$DI = \frac{\sum_{t=1}^{m} k_t^2 - \sum_{t=1}^{m} (k_t - \varphi_t)^2}{\sum_{t=1}^{m} k_t^2} \tag{2}$$

The maximum value of this metric is equal to one [19].

3.4.2. Q-Balancing Index (QBI):

The *QBI* metric was developed to achieve balanced coverage in the network based on the coverage requirement of the targets. According to Eq. (3), higher *QBI* value (closer to 1) results in more balanced coverage in the network based on the target coverage requirement. The maximum value of this metric is equal to 1; it can be obtained when the required coverage of all targets is provided.

$$QBI = \frac{(\sum_{t=1}^{m} \varphi_t)^3}{\sum_{t=1}^{m} (\varphi_t)^2} \times \frac{(\sum_{t=1}^{m} (k_t)^2)}{(\sum_{t=1}^{m} k_t)^3} \tag{3}$$

3.4.3. Coverage Quality (CQ):

The *CQ* metric presented in [28] was used in this study. It is important to note that the distance between the targets and the sensors affects the quality of the coverage in the environment. As the distance between the targets and the sensors increases, the quality of the coverage decreases. The number and distance of sensors covering a target affect the amount of *CQ*. The value of this parameter is displayed for a target with *cq* and is obtained as follows.

$$cq_{(i,j,k)} := \begin{cases} 1 - \left(\frac{|\overrightarrow{v_{i,k}}|}{R_s}\right)^2 & \text{if } t_k \text{ is covered by } s_{ij}, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\overrightarrow{v_{i,k}}$ is the vector that points from the sensor s_{i} to the target t_{k}, R_{s} is the sensing range of the sensor node, and $s_{\mathrm{i,j}}$ is the sector j of the sensor s_{i}. The total coverage quality for a solution is obtained from the coverage quality obtained for all targets in the solution. *CQ* is obtained using Eq. (5) as follows:

$$CQ = \sum_{i=1}^{n} \sum_{j=1}^{w} \sum_{t=1}^{m} qc(i,j,t) \quad (s_{\mathrm{i}} \textit{ is activated in sector } j) \tag{5}$$

The larger value indicates a better *CQ*.

3.4.4. Power Consumption (PC):

To determine the amount of energy consumption in a network, the states in which the sensor is placed must be considered. For a better comparison, the model presented in [28] was used in this study. According to the proposed model, a sensor can be in three modes: active, idle, or sleep. In the active mode, where the sensor consumes the most energy, the sensor is responsible for monitoring the targets in the environment and processing the collected data. In the idle mode, the sensor consumes power only to the extent that its operating system and processor can remain active. Finally, the sensor is passive and has the lowest energy consumption in the sleep mode. Eq. (6) is used to calculate the amount of energy consumption in a sensor [28].

$$\begin{aligned} p_{\mathrm{c}} = &\left(\mathrm{Count(sleep\ sensor)}\right) \times p_{\mathrm{s}} \\ &+ \left(\mathrm{Count(Active\ sensor)}\right) \times p_{\mathrm{a}} \\ &+ \left(\mathrm{Count(Idle\ sensor)}\right) \times p_{\mathrm{i}} \end{aligned} \tag{6}$$

In this regard, the amount of energy consumption in the active mode is equal to p_{a}; in the idle mode, it is equal to p_{i}; and in the sleep mode, it is equal to p_{s}. The values of p_{a}, p_{i}, p_{s} are chosen based on what suggested in [28], that is, $Pa = 5.268W$, $Pi = 1.473W$, and $Ps = 0.058W$.

3.4.5. Activate Sensor (AS):

To evaluate this metric, the number of activated sensors at each step is counted. When a set of sensors is selected and activated, the sensors of that set consume all or part of their energy. The fewer sensors activated in one step, the better the chance of extending network lifetime. Therefore, when comparing two algorithms, the algorithm in which a smaller number of sensors is selected at each stage is considered more efficient.

4. Proposed Algorithms

4.1. Overview of the Genetic Algorithm

To solve the problem in hand, this study uses the GA whose main application is optimization. GA is a metaheuristic approach that is intended to imitate the natural evolution process and evolve better solutions by applying genetic operations and creating successive generations. Each solution in this algorithm is represented by a simple set of genes, called the chromosome (individual). In GA, the chromosomes are evaluated by a fitness function to determine their value. The fact that GA increases the chance of achieving a global optimum as well as prevents the optimal local selection is regarded as its prominent advantage. GA is a population-based algorithm that works on the whole chromosomes instead of focusing on a single point in the search space or one chromosome. After creating the initial population in GA, three fundamental operations, namely selection, crossover, and mutation, are applied to the population. In the selection phase, valuable chromosomes are selected by the selection technique to perform the recombination process. The crossover and mutation phases are then used to discover the solution space. In the crossover phase, two chromosomes are selected as parents to generate two offsprings, through which the genetic information of their parents is exchanged. The resultant offspring are mutated to develop a better solution [31, 32].

4.2. Algorithm for over-provisioned network

This section provides a target-oriented GA-based algorithm to solve the *Q*-coverage problem in DSNs. The output of the algorithm are cover sets. At the beginning of the algorithm, the network is configured to create the cover set and the variables are then initialized. In lines 3, the set of solutions is quantified. In line 4, set *D* includes all the sectors in the network. In line 5, the initial population of GA is generated based on the specific constraint (as explained earlier). In line 6, the initial population is evaluated based on the Fitness-Over function. In line 7, the counter value of the repetition algorithm is determined in order to create a cover set. Lines 8 to 18 are the main body of GA, which is repeated by MAX. Each operation specified in this loop is described in the next section. Line 19 is dedicated to the output of the algorithm. In the following, the phases mentioned in the pseudocode are explained.

4.2.1. Representation

Chromosome modeling is the first step to solving a problem through GA. This paper uses an integer representation for chromosome modeling as a solution to the problem space. According to this model, each gene on the chromosome represents the sensor sector, and each chromosome implies a cover set. Each chromosome is represented by a two-dimensional matrix, in which the number of rows denotes the maximum coverage required by the targets and the number of columns indicates the number of targets in the network. In the matrix, the number of non-blank rows below a target indicates the quantity of coverage required by that target. Due to difference in the coverage requirement of each target, some genes may not have value in this matrix. The values in a column are the sensors that cover the target with a given sector. To better understand the proposed model, the Fig. 4 presents a chromosome with 3 targets; the maximum number of coverage required for the targets is equal to 3. As it is indicated in the figure, the targets are initialized based on the required coverage. For example, target t_1 has just three coverage, while target t_2 has one coverage. When a value is added to a chromosome as a gene, it should be noted that each directional sensor can participate in the chromosome with only one sector (direction). The chromosome is valid if this condition is met in its creation process.

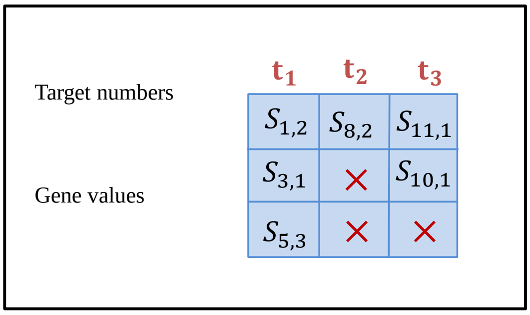


Figure 4: An example chromosome for the network.

To create the initial population, a set of chromosomes is randomly generated. First of all, this paper attempts to cover the critical targets in the system while initializing the genes on a chromosome. By definition, the targets that are covered by a minimum number of sensors are referred to as critical targets.

Algorithm 1 : Target-oriented GA-based algorithm in over provisioned networks

Input: Directional sensor network *Net = (S, T)*
Output: Cover sets with appropriate sensor sectors
$SOL \leftarrow \varnothing$
$D = \{ s_{i,j} \mid 1 \leq i \leq n,\ 1 \leq j \leq w \}$
Initialize population (pop)
Fitness-Over (pop)
  $k \leftarrow 0$
While $k \leq MAX$ **do**
  $p_s = Select\ (pop)$
  $p_c = Crossover\ (p_s)$
  $p_c = Repair\ (p_c)$
  $p_m = Mutation\ (p_c)$
  $p_m = Repair\ (p_m)$
  $p' \leftarrow p_c \cup p_m$
  $pop = Survival\ (pop, p')$
  Fitness-Over (pop)
  $k{+}{+}$
End While
  $SOL \leftarrow BEST\ (pop)$
return *SOL*

4.2.2. Fitness function

A defined criterion is needed to evaluate and compare the chromosomes reliably. The evaluation is the process of assigning a number to a chromosome depending on how value it is. GA uses a fitness function to evaluate and compare the chromosomes. By setting the parameters related to the function properly, it would be possible to achieve the optimal or near-optimal solution in the search space. In the first algorithm, the fitness function (Fitness-Over) is equal to the number of sensors that create a cover set (chromosome), and the fitness functions consisting of fewer sensors are considered to be better solutions regarding the algorithm. It is worth mentioning that only one sector of a sensor can be present in a chromosome. The fitness function of the chromosome given in Fig. 4 is equal to 6.

4.2.3. Selection

Selection is typically the first operator used to manipulate the population and applied to the chromosomes. The operator selects the proper chromosomes and allows them to have a better fitness than the others so as to reproduce more frequently. There are several methods for performing the selection phase in GA, such as roulette-wheel, rank selection, and tournament [31, 33]. The Roulette-wheel is the method applied in this algorithm; therefore, the chromosomes that have a better fitness function have a better chance of being selected. The selected chromosomes use the crossover operator to generate new offspring.

4.2.4. Crossover

The crossover operator is used to combine the genetic information of two parents in order to produce new offspring, which is regarded as one of the methods for generating new solutions from the existing population. To generate the offspring, two parents must be selected first. The offspring inherit some of the genes from the first parent and inherit the rest from their other parents. Various methods have been proposed for crossover, such as single-point, two-point, uniform crossover, etc., [32, 33].

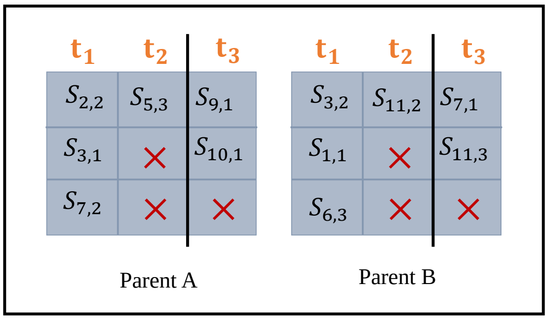


Figure 5: Representation of two parents for performing the crossover operator

This study applies the single-point method to generate offspring. Based on this method, a point is randomly selected and the two parent chromosomes exchange their information to generate two offspring. After the crossover operation, the offspring who have a multi-sector sensor are not valid and need to be modified. The repair operator was also introduced for chromosome correction, which will be explained later. Section A is provided to further clarify Fig. 5 As can be seen in this figure, there are two chromosomes on which a single-point operator is applied as a parent.

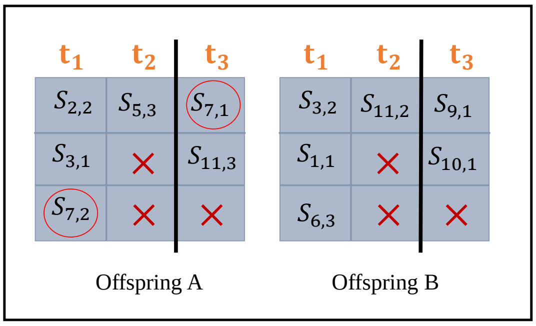


Figure 6: Representation of the two offspring generated after the crossover operation, the invalid offspring is on the left and the valid offspring is on the right.

The result of the operation is shown in Fig. 6 In offspring A, there is Sensor $s_3$with two sectors: 3 and 4, which must be replaced with one of the two genes with the correct value using the repair operator so as to produce a valid chromosome.

4.2.5. *Mutation*

Mutation operation helps GA avoid getting trapped in the ”Local Optimum”. The operator may cause major changes to the produced chromosome. In this operation, a gene is randomly selected from a chromosome and then its value is changed [33]. Once its value changes, the limitation mentioned in the proposed solution must be met and this change should not invalidate the chromosome; otherwise, another value for the gene should be replaced using the repair operator.

4.2.6. *Repair Operator*

This paper developed a repair operator to assure the chromosomes are valid after crossover and mutation operations. The operator checks the values of genes associated with each target, ensuring that there is no multi-directional sectors from a sensor on a chromosome at the same time. The process is repeated for all genes on the chromosome. Whenever a gene with an invalid value is detected, the gen value needs to be modified to the allowable value. The process continues till all genes on the chromosome get a valid value.

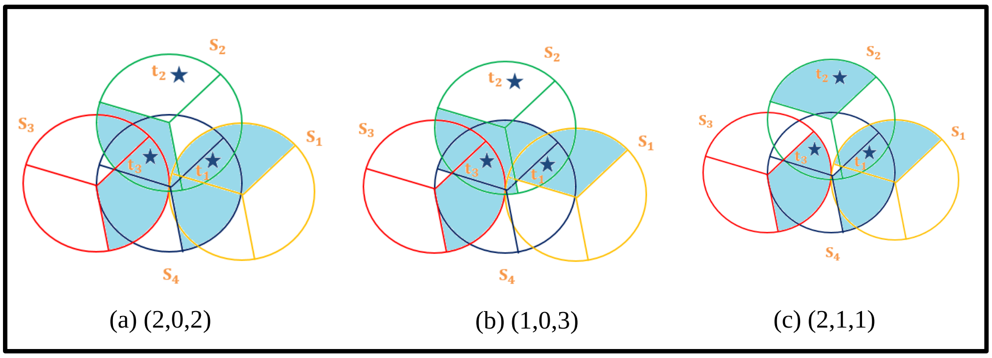


Figure 7: An example of QBI

4.2.7. *Stopping Criterion*

The stopping criterion for the algorithm in each round is the number of times specified in advance. At each time, when the repetition is done, a suitable solution (a cover set) is returned as the output of the algorithm.

4.3. *Algorithm for under-provisioned network*

As previously noted, when the network is in an under-provisioned mode, there are not sufficient sensors in the network to satisfy the coverage requirement of all the targets. In this situation, it is not intended to satisfy the coverage requirement of the targets; the coverage is provided based on the targets priority or a balanced coverage is established for the targets with the same requirements. With the aim of achieving maximum balanced coverage in such networks, this study modified the metric presented in [14, 23] in such a way that it could be applied to Q-coverage problems. The new developed metric (Q-Balancing Index (QBI)), helps algorithm 2 to evaluate chromosome and determine their superiority over each other. In Eq. (3), m indicates the number of targets in the network, k_t is the total number of sensors required to cover target t, and φ_t is the total number of sensors that cover the target t. The Fig. 7 is considered to illustrate the Eq. (3). Let us assume there are four sensors and three targets in network and the sensing angle of each sensor to be equal $\frac{\pi}{4}$. Three coverage modes are shown. If the required coverage of the targets in the network (t_1, t_2, t_3) is assumed to be equal to (3, 1, 2), respectively, the amount of achieved coverage in Figures a, b, and c are (2, 0, 2), (1, 0, 3) and (2, 1, 1), respectively. The QBI value for each of the states in the above figure is given as follows:

$$QBI = \frac{(4)^3}{8} \times \frac{14}{(6)^3} = \frac{896}{2160} = 0.51 \qquad (2,0,2)$$

$$QBI = \frac{(4)^3}{10} \times \frac{14}{(6)^3} = \frac{896}{2160} = 0.41 \qquad (1,0,3)$$

$$QBI = \frac{(4)^3}{6} \times \frac{14}{(6)^3} = \frac{896}{2160} = 0.69 \qquad (2,1,1)$$

In this relationship, larger obtained values result in more balanced coverage and also better answers will be achieved. Therefore, according to the obtained values, the value of 0.69 and coverage of (2, 1, 1) will be the best option.

Algorithm 2 : Target-oriented GA-based algorithm in under provisioned networks

Input: Directional sensor network *Net = (S, T)*
Output: Cover set with appropriate sensor sectors
$SOL \leftarrow \varnothing$
$D = \{ s_{i,j} \mid 1 \le i \le n,\ 1 \le j \le w\}$
Initialize population (pop)
Fitness-Under (pop)
  $p_s = Select\ (pop)$
  $p_c = Crossover\ (p_s)$
  $p_c = Repair\ (p_c)$
  $p_m = Mutation\ (p_c)$
  $p_m = Repair\ (p_m)$
  $p' \leftarrow p_c \cup p_m$
  $pop = Survival\ (pop, p')$
  Fitness-Under (pop)
  $SOL \leftarrow BEST\ (pop)$
return *SOL*

Algorithm 2 is presented to solve the problem in under provisioned networks. The algorithm uses the Fitness-Under evaluation function, which evaluates chromosomes based on *QBI*. This algorithm is executed in one step and the output of the algorithm is a cover set.

4.4. Time complexity

To understand the time complexity of the proposed algorithms, the different parts of the genetic algorithm need to be well considered. The proposed genetic algorithm consists of three main phases: the creation of initial population, evaluation of the fitness function, and reproduction. In the reproduction phase, the selection, crossover, mutation and repair operators are applied. The chromosomes are randomly encoded as two-dimensional matrices with m rows and n columns with integer numeric values. It takes approximately $O(n \times m)$ time to encode each chromosome and $O(n \times m \times N_{\text{pop}})$ to generate an initial population of N_{pop}. The fitness function is evaluated in both over-provisioned and under-provisioned during $O(n \times m)$ period of time. In this study, the single-point method is used to perform the crossover operator and create offspring, which is done during $O(N_{\text{pop}})$. In mutation, a gene value is selected from the valid values related to a target; then, the selected value replaces one of the values of the target in the chromosome. This operation can be performed in the constant time of $O(1)$. An invalid child may be created after applying mutation to a chromosome; thus, the chromosome validity needs to be checked. Therefore, the mutation operator can be performed in $O(n \times m)$ time period.

5. Simulation Results

This section describes several experiments conducted to evaluate the performance of the proposed algorithms. For the purpose of this study, the sensors and targets were randomly distributed in the environment, and the coverage requirements of the targets were predetermined non-uniformly and randomly equal to 1, 2, 3, and 4. In this paper, it is assumed that each sensor has finite orientations or sectors and the sectors are disjoint. The experiments were performed in such a way that the network could run in both under-provisioned and over-provisioned environments. The developed algorithms were compared with the algorithm presented in [28]; five metrics were used to compare the performance of the algorithms. The experimental results show the effects of changing the number of sensors, the number of targets, and the value of sensing range relative to the five metrics. For more reliability, each test scenario in this paper was performed 20 times. For simulation purposes, MATLAB R2014a was used on a system with RAM 4 GB, Intel i3 processor 1.7 GHz in Windows 7 platform. Table 2 lists the parameters and test conditions.

The experiments were performed in three separate groups, and in each group, five experiments were carried out. To this end, the experiments were performed

Table 2: Simulation Parameters

Parameters	Values
Monitored region	500 m × 500 m 1000 m × 1000 m
Number of directional sensors	20-180
FOV	90 (deg)
Number of targets	10-150
Deployment	Random
Sensing radius	30-130
Comm. radius	2 × Sensing radius
Target coverage requirement	1, 2, 3, 4 random and Non-uniform

in two distinct environments. The first and second groups were conducted in a 500*500 environment, while the third one was performed in a 1000*1000 environment. In the first group, the number of sensors is variable, while the sensing range and the number of targets are assumed to be constant, and the effect of the number of sensors on metrics is the output of the experiments. The second groups of experiments were conducted to examine the effect of the number of targets on the metrics. In these experiments, the amount of sensing range and the number of sensors are constant. In the third group, the number of sensors and the number of targets are constant, while the value of the sensing range is variable, and the output of the experiments is the effect of sensing range on the metrics.

5.1. ***Experiments***

5.1.1. ***Group 1.***

In this group, the number of sensors was increased from 50 to 150, with incremental step of 20. The number of targets was fixed at 130 and the sensing range at 100.

Experiment 1. This experiment was intended to investigate the effect of increasing the number of sensors on QBI. Note that as the number of sensors increases, the network switches from under-provisioned to over-provisioned in which there are sufficient sensors to satisfy the coverage requirement of all targets; as a result, φ_t increases. In Eq. (3), after increasing φ_t, the amount of $(\sum_{t=1}^{m} \varphi_t)^3$ and $\sum_{t=1}^{m}(\varphi_t)^2$ increased, but the amount of $(\sum_{t=1}^{m} \varphi_t)^3$ increased more. Therefore, the value of *QBI* got closer to the optimal value 1. Almost from the amount of

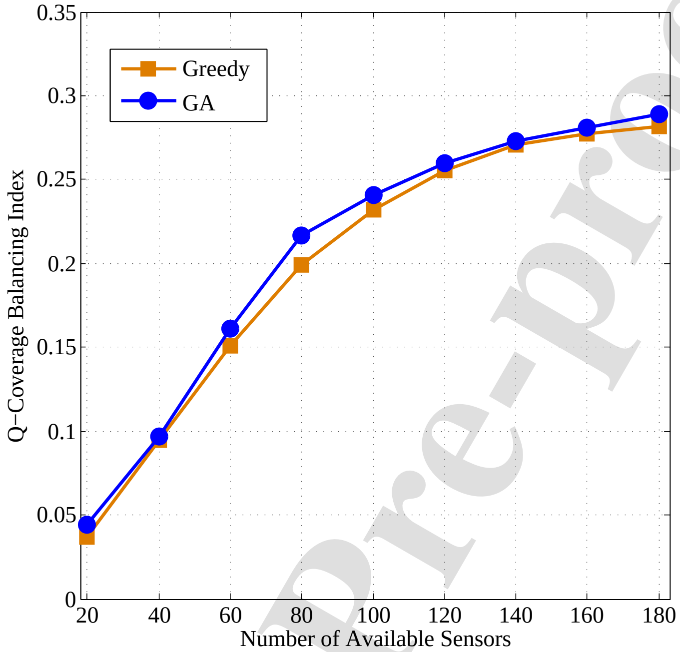


Figure 8: Effect of increasing the number of sensors on the Q-coverage balancing Index

$n = 140$ (number of sensors), QBI becomes saturated. This was because the coverage requirement of targets was satisfied. As indicated by Fig. 8, the proposed GA performed better than the greedy algorithm.

Experiment 2. This experiment was aimed at investigating the impact of increasing the number of sensors on Distance Index (DI). As the number of sensors increased, the network changed from under-provisioned to over-provisioned state and the coverage requirements of the targets were satisfied. As indicated in Fig. 9, the DI value increased and approached the value $(DI = 1)$. From one point onwards, increasing the number of sensors had no effect on the network since the coverage requirement of the targets had been already met. The reason can be expressed as follows: in Eq. (7), when the achieved coverage of each target increases, the amount of φ_t increases for $t = 1, 2, \ldots, m$. As a result, the fraction $\frac{\sum_{t=1}^{m}(k_t-\varphi_t)^2}{\sum_{t=1}^{m}k_t^2}$ in Eq. (7) decreases while DI increases. When φ_t becomes close to

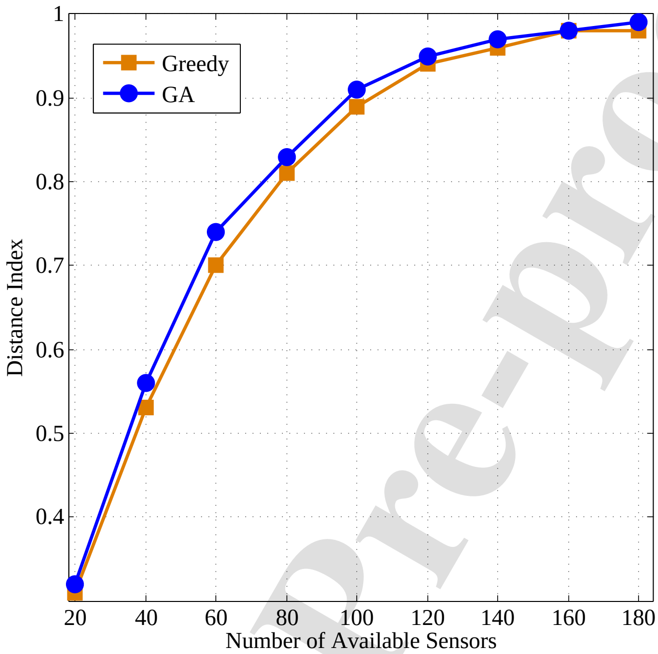


Figure 9: Effect of increasing the number of sensors on the distance index

k_t at a point and as φ_t is bounded by k_t, the distance index becomes saturated to value 1. In other words, when all the targets receive their coverage requirement, the value of DI becomes 1. In the following, Eq. (7) is presented:

$$DI = 1 - \frac{\sum_{t=1}^{m}(k_t - \varphi_t)^2}{\sum_{t=1}^{m} k_t^2} \tag{7}$$

Experiment 3. In this experiment, the number of sensors was varied to investigate its effect on Coverage Quality (CQ). As the number of sensors increased, the network changed from under-provisioned mode to over-provisioned mode. By increasing the number of sensors, the coverage requirements of the targets were met. Increasing the number of sensors reduced the distance between the targets and the sensors; as a result, according to Eq. (5), the coverage quality improved. After a certain point, coverage quality became saturated. The reason can be stated as follows: additional sensors do not need to be activated once the coverage requirements of sensors are already met. The obtained results are shown in Fig. 10.

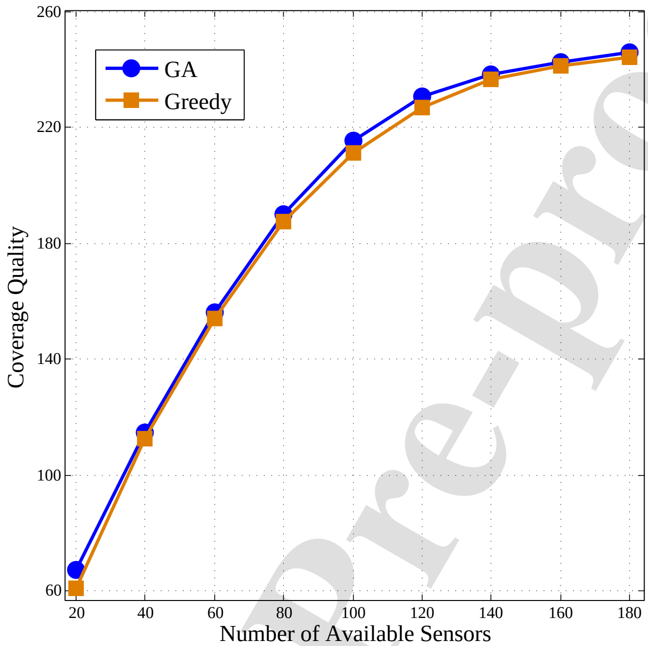


Figure 10: Effect of increasing the number of sensors on the coverage quality

Experiment 4. In this experiment, the impact of increasing the number of available sensors on the number of active sensors was investigated. When there are not enough sensors in the network, the network is in under-provisioned mode and the number of active sensors is small. As the number of sensors was increased, the network gradually entered the over-provisioned mode, and the number of selected sensors was also increased to satisfy the target coverage requirements. From a point onwards, as the number of available sensors was increased and the network became an over-provisioned system, the number of active sensors changed slightly and became fixed after a certain number of available sensors (almost $n = 40$). The cause can be stated as follows: sensors are good enough to meet all coverage requirements, and the network turns into the over-provisioned mode. Remember that in DSNs, there is a relationship between the number of active sensors and the network lifetime. That is, by increasing the number of active sensors, energy consumption increased, and the network lifetime will be decreased. As stated in

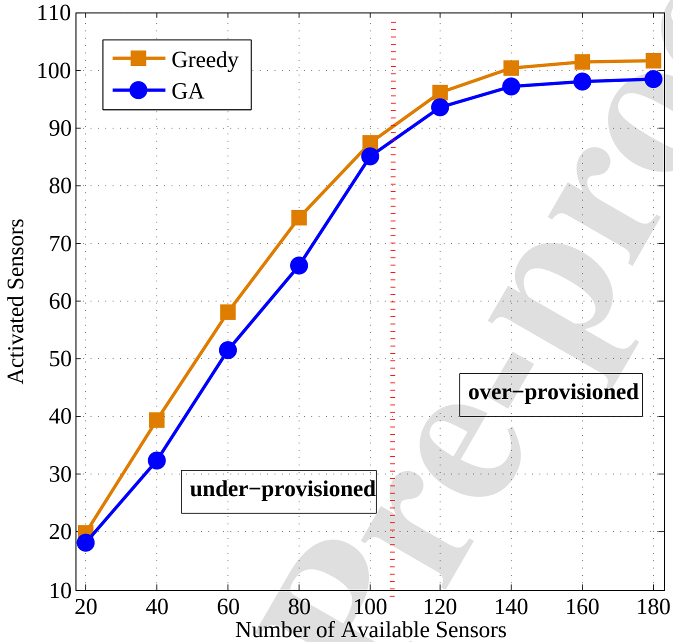


Figure 11: Effect of increasing the number of sensors on activated sensors

Fig. 11, the number of active sensors in GA was smaller than that of the greedy algorithm. Therefore, GA can prolong the network lifetime.

Experiment 5. This experiment examined the effect of increasing the number of sensors on the amount of energy consumption in the network. When the number of active sensors in the network is small, the network is in the under-provisioned mode, which necessitates the activation of more sensors. By increasing the number of sensors, the network shifts to an over-provisioned mode. When newly-added sensors become immediately activated in an under-provisioned mode, power consumption in the network increases gradually. From a point onwards (almost $n = 140$), increasing the number of sensors in the over-provisioned mode did not have much impact on the number of active sensors and energy consumption. The reason is that the targets, in that condition, have achieved their coverage requirements and increasing the number of sensor has no impact on covering the targets. The results are displayed in Fig. 12. GA was found more successful in terms of

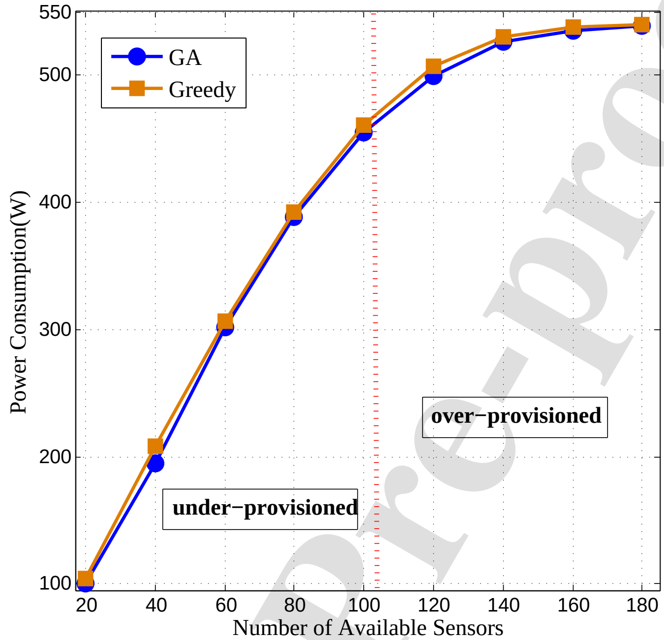


Figure 12: Effect of increasing the number of sensors on the power consumption

extending network lifetime, because it consumes less energy.

5.1.2. ***Group 2.***

In this group of experiments, the number of targets was increased from 50 to 150, with incremental step of 20. The number of sensors was fixed at 130, and the sensing range at 100.

Experiment 1. This experiment was conducted to know how the increased number of targets affects the *QBI* value. The network becomes under provisioned as the number of targets increases and the number of sensors remains constant. The reason is that, in this condition, more sensors are required to satisfy the target coverage requirement. Remember that in Eq. (3), the number of targets increased, the amount of φ_t decreased. Consequently, the value of $(\sum_{t=1}^{m}\varphi_t)^3$ and $\sum_{t=1}^{m}(\varphi_t)^2$ decreased, but the amount of $(\sum_{t=1}^{m}\varphi_t)^3$ decreased even more. The results are depicted in Fig. 13. The *QBI* value decreased and became farther

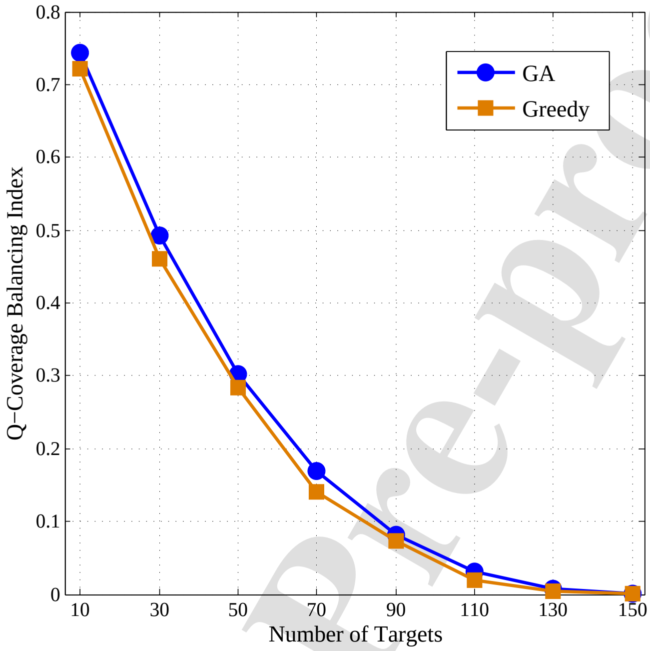


Figure 13: Effect of increasing the number of targets on the Q-coverage balancing index

from the ideal value $(QBI = 1)$ by increasing the number of targets and shifting to under-provisioned mode. Fig. 13 implies the superiority of GA as compared to the greedy algorithm.

Experiment 2. This experiment was performed to investigate the impact of increasing the number of targets on DI. As it is shown in Fig. 14, as the number of targets increased, the network switched from over-provisioned into under-provisioned mode. In this case, the distance between the required coverage vector of the targets and their achieved coverage vector increased. According to Fig. 14, as the number of targets increased, the curve became farther from the normal state $(DI = 1)$. Looking at Eq. (2) and changing it to Eq. (7), we find that as the number of targets increases, both $\sum_{t=1}^{m} k_t^2$ and $\sum_{t=1}^{m}(k_t - \varphi_t)^2$ increased, too. Note that the amount of increase for $\sum_{t=1}^{m}(k_t - \varphi_t)^2$ was less than or equal to the amount of increase for $\sum_{t=1}^{m} k_t^2$. Therefore, with the increase of the number of targets, the

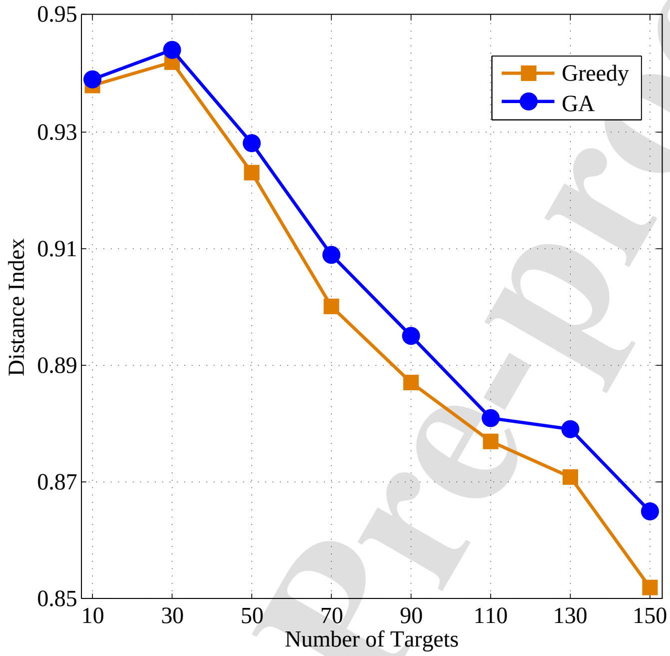


Figure 14: Effect of increasing the number of targets on the distance index

fraction $\frac{\sum_{t=1}^{m}(k_t-\varphi_t)^2}{\sum_{t=1}^{m}k_t^2}$ increased, whereas the *DI* value decreased.

Experiment 3. This experiment examines the impact of increasing the number of targets on *CQ*. Remember that in Eq. (5), all targets participate in the amount of *CQ*. As the number of targets increased, density of targets and sensors in the environment increased and the distance between the sensors and the targets decreased; consequently, additional targets contributed to the *CQ* value, and the value increased. The graph in Fig. 15 illustrates the results.

Experiment 4. In this experiment, the effect of the number targets on the number of active sensors was investigated. As the number of targets in the network is increased, more sensors must be activated to cover the targets. In this condition, the network shifts to the under-provisioned mode from the over-provisioned mode. With increasing the number of targets, the number of active sensors increased. From one point onwards (almost $m = 100$), increasing the number of

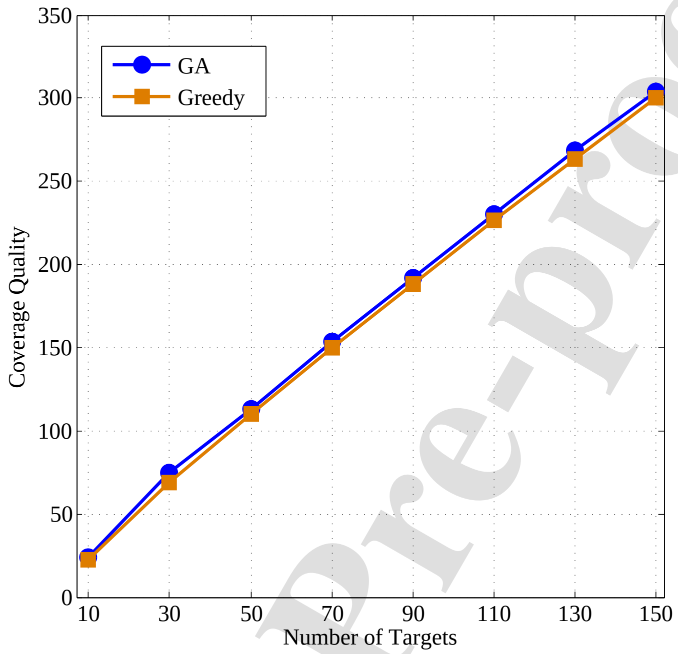


Figure 15: Effect of increasing the number of targets on the coverage quality

targets showed no more effect on the number of active sensors, because there was no sensor left to be activated. Fig. 16 displays the obtained results.

Experiment 5. The purpose of this experiment was to investigate the effect of the number targets on the energy consumption of sensors in the network. By adding new targets, the network shifted to the under provisioned mode from the over-provisioned mode. As the number of targets was increased, the number of active sensors increased, too; as a result, the amount of energy consumption increased. From one point onwards, the amount of energy consumption did not change much; the reason was that no sensor remained to be activated. Remember that the more active sensors in a network, the higher the power consumption and decreases network lifetime. The results are depicted in Fig. 17.

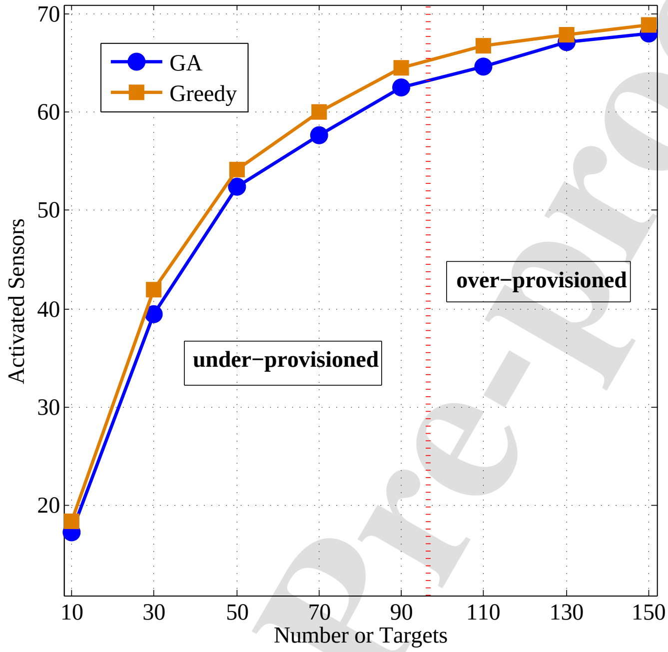


Figure 16: Effect of increasing the number of targets on the active sensors

5.1.3. ***Group 3.***

To perform the experiments of this group, the number of sensors and the number of targets were set to 300 and 20, respectively, and the sensing range was set to 50-200, with incremental step of 25. As mentioned earlier, a 1000*1000 field was used to carry out the experiments in this group.

Experiment 1. This experiment was aimed at investigating the effect of increasing the sensing range on the *QBI*. As the sensing ranges of the sensors in a network increase, more targets may be covered by the sensors and the coverage requirement of targets could be satisfied. In this condition, the network switches from under-provisioned to over-provisioned mode. In Eq. (3), after increasing φ_t, the amount of both $(\sum_{t=1}^{m} \varphi_t)^3$ and $\sum_{t=1}^{m} (\varphi_t)^2$ increased, too; however, the amount of $(\sum_{t=1}^{m} \varphi_t)^3$ increased more. Consequently, the value of *QBI* increased. Note that of approximately the amount of $R_s = 100$ (sensing range), *QBI* is bounded by (0.25). The graph in Fig. 18 indicates that the *QBI* approaches the ideal value $(QBI = 1)$ as a result of increasing the sensors' sensing range.

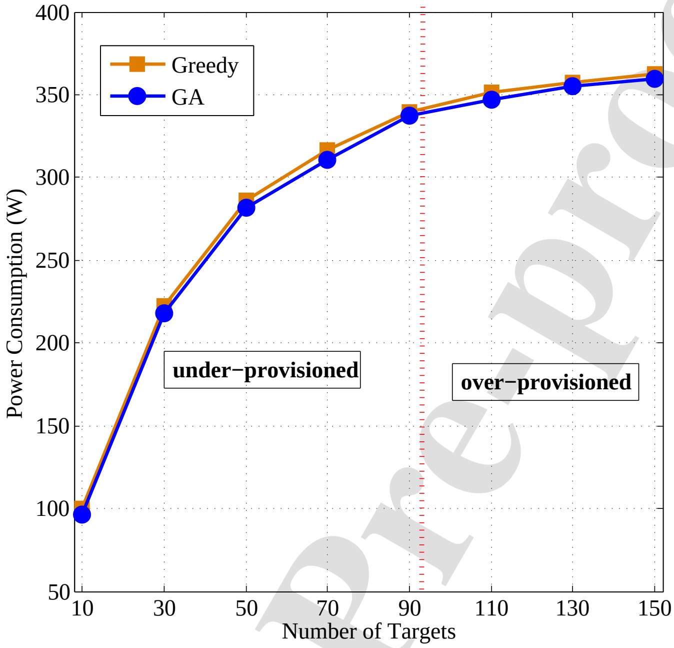


Figure 17: Effect of increasing the number of targets on the power consumption

Experiment 2. This experiment was conducted to examine the effect of increasing the sensing range on *DI*. As it is widely known, when the sensing range of a sensor increases, more targets may be covered by the sensor; consequently, more coverage can be achieved by less sensors. Increasing the sensing range of the sensors causes the network to shift to the over-provisioned mode. According to Fig. 19, as the sensing range is increased, the *DI* value increases and approaches the optimal value $(DI = 1)$. Remember in Eq. (7), when the amount of φ_t increases for $t = 1, 2, \ldots, m$, the fraction $\frac{\sum_{t=1}^{m}(k_t-\varphi_t)^2}{\sum_{t=1}^{m}k_t^2}$ decreases, *DI* increases. φ_t becomes near to k_t at a point, and φ_t is bounded by k_t. As a result, *DI* becomes saturated to value 1.

Experiment 3. This experiment aimed to investigate the impact of changing the sensing range on *CQ*. As it is widely accepted in the literature, the sensors and targets are assumed to be located in a fixed environment, and as the sensing

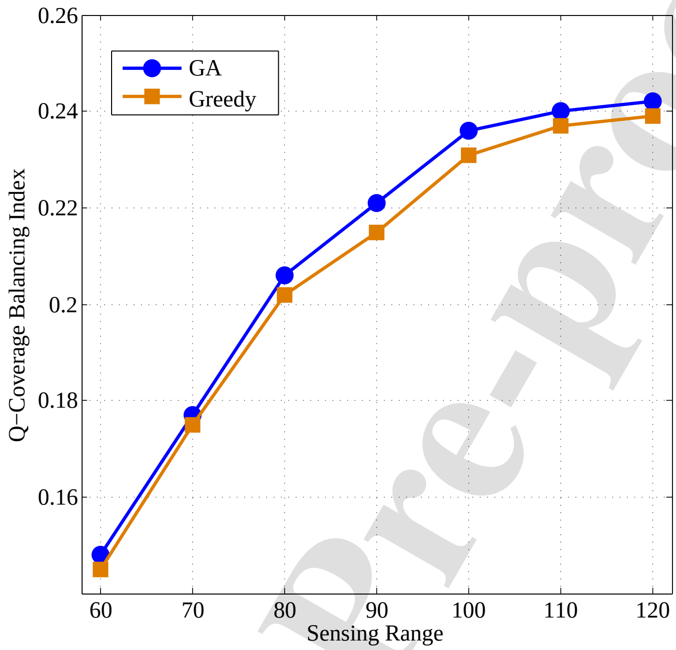


Figure 18: Effect of increasing the sensing range on the Q-coverage balancing index

range of the sensors increases, the distance between the targets and the sensors does not change; however, in this situation, more targets could be covered by a sensor. According to Eq. (5), and as indicated in Fig. 20, the CQ value increases with increasing the sensing range. But in this case, CQ becomes saturated after all targets achieve their coverage requirements; there is no need to increase the sensing range when the requirements have been already met. From now on, CQ becomes saturated.

Experiment 4. In this experiment, the effect of increasing the sensing range on the number of active sensors was investigated. The results are shown in Fig. 21. By increasing the sensing range, each sensor could cover more targets; as a result, the network was switched to over-provisioned mode, the quality of target coverage in the network was improved, and the coverage requirements of the targets were entirely met. In this condition, the number of active sensors decreased, and the network run with fewer sensors. Increasing the number of active sensors in DSN

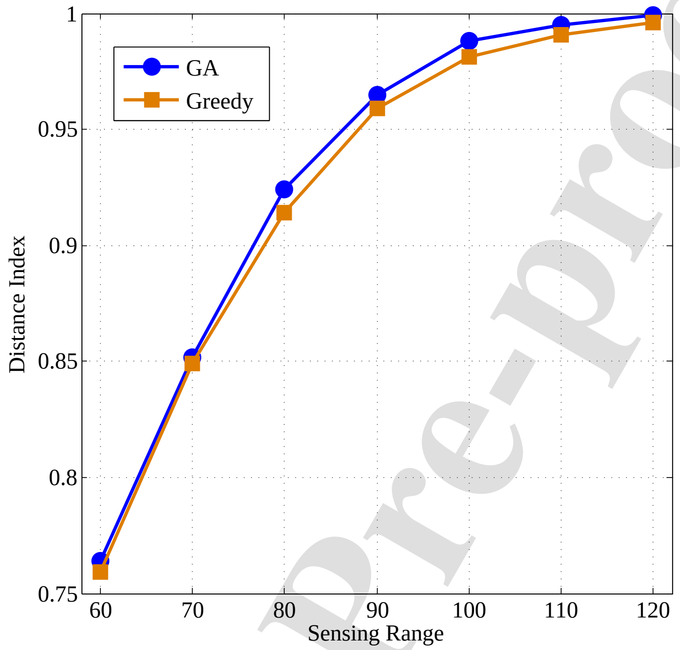


Figure 19: Effect of increasing the sensing range on the distance index

causes the total network lifetime to be increased.

Experiment 5. As shown in Fig. 22, the purpose of this experiment was to investigate the effect of the sensing range on the energy consumption of sensors in the network. Remember the results of the fourth experiment of this group; more targets are covered by increasing the sensing range of the sensors. There is no need to activate more sensors when the coverage requirements of all the targets are already met and the network is in the over-provisioned mode. This experiment showed that by reducing the number of active sensors in the network, the amount of energy consumption decreased, too. As the power consumption in the network decreases, the network lifetime increases.

5.2. Discussion on simulation results

In this paper, a sensor-oriented greedy algorithm [28] and developed target-oriented GA-based algorithms were compared with each other, and the superiority

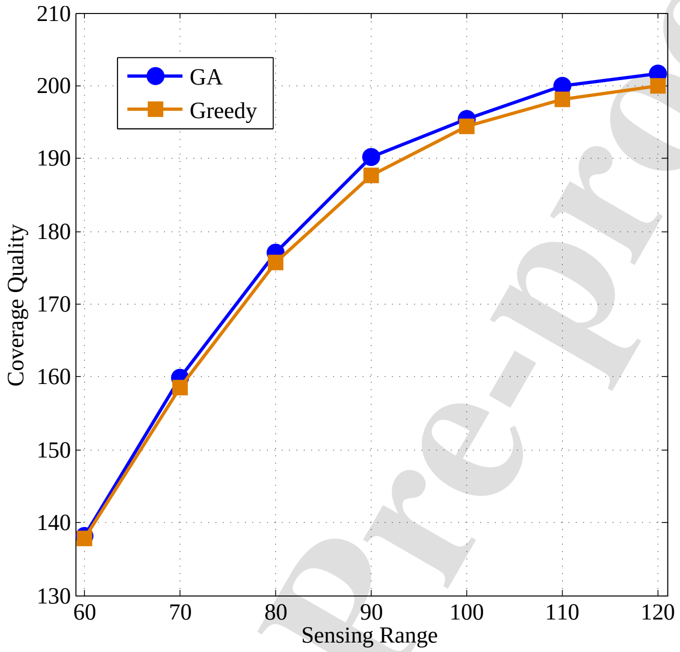


Figure 20: Effect of increasing the sensing range on the coverage quality

of the developed GA-based algorithm was confirmed for the following reasons: Greedy-based algorithms are able to find coverage sets for maximizing the network lifetime in real time. It is noticeable that the performance of greedy-based algorithms is extremely dependent on the closeness of the initial candidates to the optimal solution and the scheme can result a local minimum due to its heuristic search [34]. Remember that GA is one of the evolutionary algorithms, which is modeled according to the characteristics of the problem in hand. These algorithms have been applied by many researchers to various issues, including the optimization of sensor networks. GA is a population-based search method that moves towards the optimal global answer by selecting, repeating, and creating a set of candidate solutions and then evaluating them.

GA operate in a non-deterministic way and perform stochastic population-based heuristic search-based exploratory searches. It is possible to achieve different results in independent performances, and there is no guarantee that optimal results will be achieved. Good results can be expected when the ”global” search space

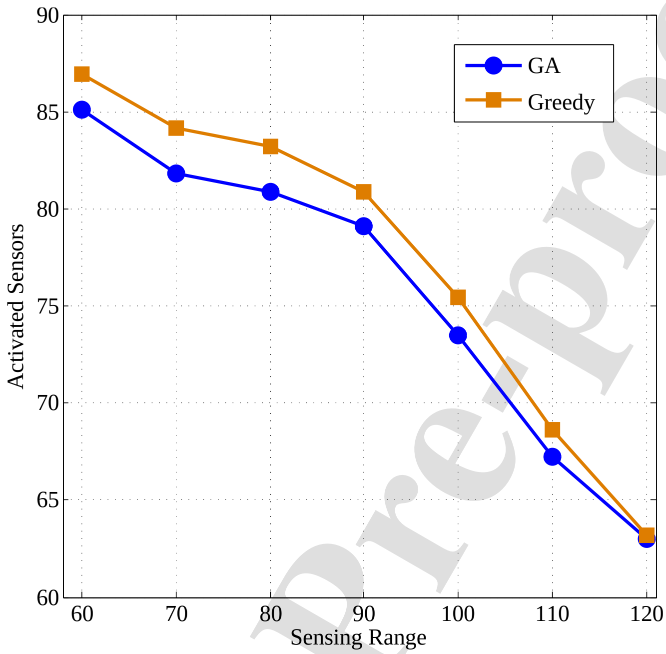


Figure 21: Effect of increasing the sensing range on the active sensors

is examined globally. By using evolutionary algorithms and high iterations in the global search space, we will have a relatively higher computational cost compared to the greedy algorithm. In optimization problems that have a large search space (e.g., coverage optimization in sensor networks), achieving a global optimization (or even a state near to it) has a high computational cost, but it is worth it [35]. Critical targets are targets that covered by a small number of sensors. If a target is not covered in over-provision environment, the network lifetime will be expired. In the target-oriented GA-based algorithm, the critical targets first selected and managed to ensure that all targets are covered. This is ignored in the developed greedy algorithm.

6. Conclusion and further work

This paper investigated the target Q-coverage problem in DSNs and developed two target-oriented GA algorithms to solve the problem. Both over-provisioned

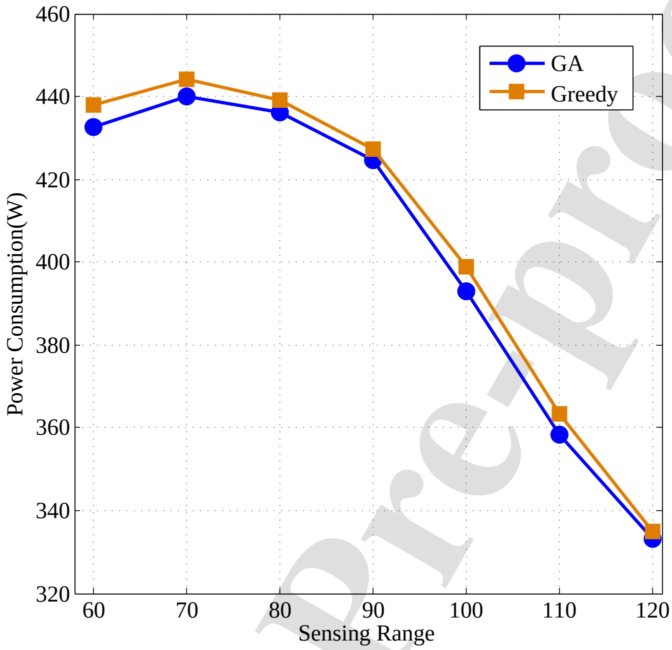


Figure 22: Effect of increasing the sensing range on the power consumption

and under-provisioned environments were studied in order to evaluate more accurately the network performance in real-world conditions. In case of the over-provisioned environment, an target-oriented GA algorithm was developed to select the minimum number of sensor sectors, which could satisfy the coverage requirement of all the targets. In case of the under-provisioned environment, an target-oriented GA algorithm was proposed to achieve a maximum balanced coverage the network. Furthermore, the present paper offers some initiatives to facilitate problem solving. For example, a new model was developed for chromosome which can hold an unequal number of targets requirements in the *Q*-coverage model. In this model, to detect invalid chromosomes and make them valid after crossover and mutation operators, the repair operator was developed. Critical targets were considered and the *QBI* metric was developed to solve the problem.

Moreover, this study included several experiments performed to evaluate the performance of the proposed algorithms. The algorithms performance was compared with that of a recently-proposed sensor-oriented greedy algorithm. Five metrics

(Distance Index, Q-Balancing Index, Coverage Quality, Power Consumption and Activate Sensor) were developed to evaluate the performance of the algorithms. The results confirmed the capability of the algorithms developed in this paper to solve the Q-coverage problem in both environments.

However, coverage management in DSNs is still an unlocked research challenge. A part of our future work is to use the adjusting feature for sensors in the under-provisioned environment to achieve the maximum balanced coverage for all the targets in the network. Existing studies have shown that adjusting the directions of sensor nodes in a network leads to better coverage of targets and conservation of sensors' energy, hence prolonging the network lifetime.

References

[1] Yick, J., B. Mukherjee, and D. Ghosal, Wireless sensor network survey. Computer Networks, 52(12) (2008) 2292-2330.

[2] Lautner, D., et al., WaaS (Wireless-as-a-Sensor): Conception, design and implementation on mobile platforms. 88 (2018) 65-73.

[3] Atzori, L., A. Iera, and G. Morabito, The Internet of Things: A survey. Computer Networks, 54(15) (2010) 2787-2805.

[4] Akyildiz, I.F., et al., Wireless sensor networks: a survey. Computer Networks, 38(4) (2002) 393-422.

[5] Ma, H. and Y. Liu. On coverage problems of directional sensor networks. in International Conference on Mobile Ad-Hoc and Sensor Networks (2005) Berlin, Heidelberg: Springer Berlin Heidelberg.

[6] Guvensan, M.A. and A.G.J.A.H.N. Yavuz, On coverage issues in directional sensor networks: A survey, 9(7) (2011) 1238-1255.

[7] Sangwan, A. and R.P. Singh, Survey on Coverage Problems in Wireless Sensor Networks. Wireless Personal Communications, 80(4) (2015) 1475-1500.

[8] Gu, Y., et al., QoS-aware target coverage in wireless sensor networks, 9(12) (2009) 1645-1659.

[9] Zhao, Q. and M.J.I.A.t.o.n. Gurusamy, Lifetime maximization for connected target coverage in wireless sensor networks, 16(6) (2008) 1378-1391.

[10] Cardei, M. and D.-Z.J.W.n. Du, Improving wireless sensor network lifetime through power aware organization, 11(3) (2005) 333-340.

[11] Sangwan, A. and R.P.J.W.P.C. Singh, Survey on coverage problems in wireless sensor networks, 80(4) (2015) 1475-1500.

[12] Cardei, M., et al. Energy-efficient target coverage in wireless sensor networks. in Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (2005) IEEE.

[13] Li, Y. and S.J.J.o.C.O. Gao, Designing *K*-coverage schedules in wireless sensor networks, 15(2) (2008) 127-146.

[14] Malek, S.M.B., et al., On balanced *K*-coverage in visual sensor networks, 72 (2016) 72-86.

[15] Alibeiki, A., et al., A new genetic-based approach for solving *K*-coverage problem in directional sensor networks, 154 (2021) 16-26.

[16] Manju, et al., Target *K*-coverage problem in wireless sensor networks, 23(2) (2020) 651-659.

[17] Zannat, H., et al., The coverage problem in visual sensor networks: A target oriented approach, 75 (2016) 1-15.

[18] Ai, J. and A.A.J.J.o.C.O. Abouzeid, Coverage by directional sensors in randomly deployed wireless sensor networks, 11(1) (2006) 21-41.

[19] Mohamadi, H., et al., Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges, 46 (2014) 26-35.

[20] Alibeiki, A., et al., A new genetic-based approach for maximizing network lifetime in directional sensor networks with adjustable sensing ranges, 52 (2019) 1-12.

[21] Mottaki, N.A., et al., Multi-Objective Optimization for Coverage Aware Sensor Node Scheduling in Directional Sensor Networks, 4(1) (2021) 43-52.

[22] Javan Bakht, A., H. Motameni, and H. Mohamadi, A learning automata-based algorithm for solving the target K-coverage problem in directional sensor networks with adjustable sensing ranges. Physical Communication, 42 (2020) 101156.

[23] Javan Bakht, A., et al., A learning automata-based algorithm to solve imbalanced K-coverage in visual sensor networks. Journal of Intelligent and Fuzzy Systems 39(3) (2020) 2817-2829.

[24] Chaudhary, M. and A.K. Pujari. Q-coverage problem in wireless sensor networks. in International Conference on Distributed Computing and Networking. (2009) Springer.

[25] Liu, H., et al. Energy-efficient algorithm for the target Q-coverage problem in wireless sensor networks. in International Conference on Wireless Algorithms, Systems, and Applications. (2010) Springer doi:10.1007/s11227-011-0646-9.

[26] Arivudainambi, D., et al. Energy efficient sensor scheduling for Q-coverage problem. in 2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (2017) IEEE.

[27] Singh, A., A. Rossi, and M.J.E.O. Sevaux, Matheuristic approaches for Q-coverage problem versions in wireless sensor networks. 45(5) (2013) 609-626.

[28] Al Zishan, A., et al., Maximizing heterogeneous coverage in over and under provisioned visual sensor networks. 124 (2018) 44-62.

[29] Kumar, A., A.K.J.I.J.o.A.S. Dhingra, and E. Research, Optimization of Scheduling Problems: A genetic algorithm survey. 1(1) (2012) 11-25.

[30] Jing, Ai, et al., Coverage by directional sensors in randomly deployed wireless sensor networks. J. Combin. Optim. 11 (1) (2006) 21–41.

[31] Sivanandam, S. and S. Deepa, Genetic algorithms, in Introduction to genetic algorithms. Springer (2008) 15-37.

[32] Asorey-Cacheda, R., et al., A survey on non-linear optimization problems in wireless sensor networks, 82 (2017) 1-20.

[33] Goldberg, D.E., Genetic algorithms (2006) Pearson Education India.

[34] Mohamadi, H., S. Salleh, and A.S.J.W.p.c. Ismail, A learning automata-based solution to the priority-based target coverage problem in directional sensor networks. 79 (3)(2014) 2323-2338.

[35] Elivelton O.Rangel, Daniel G.Costa, Angelo Loula, On redundant coverage maximization in wireless visual sensor networks: Evolutionary algorithms formulti-objective optimization, 82 (2019) 105578.

Nemat allah mottaki is currently pursuing the Ph.D. degree in Computer Software at Department of Computer Science. He is working on several researches related to evolution algorithms and computer networks.

Homayun Motameni received B.S. degree in Computer Engineering-Software engineering in Shahid Beheshti of Tehran University and M.S. degree in Computer Engineering-Machine Intelligence from Islamic Azad University-Science and Research Branch in 1995 and 1998, respectively. He received Ph.D. degree in Computer Engineering-Software Engineering from Islamic Azad University-Science and Research Branch in 2007. His current research interests include Evolution Algorithms, Petri Net, software systems modeling and evaluation using Petri Net, and machine learning.

Hosein Mohamadi is currently pursuing the Ph.D. degree in Computer Science at Faculty of Computing, Universiti Teknologi Malaysia. He is working on several researches related to computer networks, especially on wireless networks, learning systems, and soft computing.

Nemat allahMottaki

Homayun Motameni

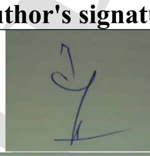
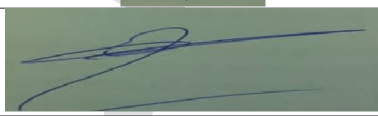
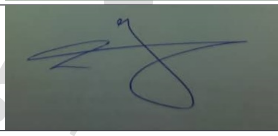
Hosein Mohamadi

Conflicts of Interest Statement

Manuscript title: A genetic algorithm-based for solving the target Q-coverage problem in over and under provisioned Directional Sensor Networks

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership,employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial intrest (such as personal or professional relationships, affiliations, knowledge or beliefs) in .the subject matter or materials discussed in this manuscript

:Author names

Author's name(typed)	**Author's signature**	**Date**
Nemat allah Mottaki		2020/3/25
Homayun Motameni		2020/3/25
Hosein mohamadi		2020/3/25