

A new genetic-based approach for solving k -coverage problem in directional sensor networks

Abolghasem Alibeiki^a, Hodayun Motameni^{a,*}, Hosein Mohamadi^b

^a Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

^b Department of Computer Engineering, Azadshahr Branch, Islamic Azad University, Azadshahr, Iran

ARTICLE INFO

Article history:

Received 13 March 2020

Received in revised form 10 February 2021

Accepted 15 March 2021

Available online 20 April 2021

Keywords:

Directional sensor networks

Scheduling algorithms

Genetic algorithm

ABSTRACT

In recent years, the use of directional sensor networks (DSNs) has continued to rise increasingly, which is due to their extensive use in many situations. In such networks, the main problem is how to cover the targets distributed in a defined area and simultaneously prolong the network lifetime as much as possible. The reason of this problem is the limitation of both sensing angle and energy resource of sensors in such networks. This problem gets more complex in cases where targets need to be covered by more than one sensor direction (i.e., each target needs to be monitored for at least k times). This problem is generally known as target k -coverage problem which its NP-completeness has been already proved in the literature. The k -coverage problem can be considered in over-provisioned and under-provisioned environments. In both of these environments, especially in the latter, it is important to create a balanced coverage, as these environments do not have enough sensors to monitor all targets for k times. Thus, in this paper, we proposed a genetic-based algorithm to solve the problem in over-provisioned environments, then developed the proposed algorithm in another way to solve the problem in under-provisioned networks so that it uses the minimum number of sensors. many experiments were performed to test the efficiency of the proposed algorithm, and the obtained results showed the high capacity of the proposed algorithm in solving the k -coverage problem in both environments.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

In recent years, researchers have been increasingly attracted to visual sensor networks due to their wide applicability in real scenarios. Video-transmitting sensors are getting cheaper, hence more available in market; therefore, they are increasingly used in fields such as environment monitoring, surveillance systems, smart traffic controlling systems, etc. [10]. A visual sensor network consists of a set of sensors that monitor a set of targets in the network. The sensors in such networks are capable to control their own orientation. In a general view, visual sensor networks (VSNs) are divided into two classes: over-provisioned networks and under-provisioned networks. A VSN is over provisioned if it contains enough sensors to monitor all targets; otherwise, it is termed under provisioned. In an under-provisioned network, the target coverage must be maximized as much as possible regardless of the number of used sensors because the network does not necessarily have enough sensors to cover all targets. Surprisingly, providing coverage in an over-provisioned network is still more challenging than an under-provisioned one since in

the former, the number of sensors must be minimized and at the same time, the coverage needs to be maximized as much as possible [10,22].

The main task of a VSN is coverage which simply refers to collecting data from environment. Note that the goals defined for each application determine the type of data collected. Coverage problem is divided into two different categories: area coverage and target coverage [23]. While the former focuses on providing coverage for an entire area, the latter is mainly focused on providing coverage for only some defined points in the area of interest. Coverage can be also considered in three different types: simple coverage, k -coverage, and Q -coverage. In the simple coverage, each target is monitored by a single sensor. The problem that may arise in this coverage is that if the sensor monitoring a target gets exhausted, blocked by a perpetrator, or malfunctions, the target may lose its coverage. A solution to this problem is combining fault tolerance with the coverage task. To do this, each target in the network needs to be monitored by more than one sensor. In the literature, this problem has been recognized as k -coverage. Note that simple coverage is actually a certain type of k -coverage in which $k=1$. To improve the accuracy of coverage and make it more fault tolerant, it is desirable to have a higher value of k . The efficiency of the solution proposed for this problem

* Corresponding author.

E-mail address: motameni@iausari.ac.ir (H. Motameni).

depends on the number of sensors in each coverage set the fewer sensors used, the less energy consumed to do the coverage task. It can ultimately lead to extension of the network lifetime. As a result, k -coverage problem attempts to both minimize the number of sensor used in coverage process and have a fault-tolerant way of coverage. Remember that the k coverage problem can be generally considered in two environments: over provisioned and under provisioned. In over-provisioned networks, there are enough sensors to provide all targets in the network with desirable coverage for k times, while the under-provisioned ones lack enough sensors to do it. The aim of k -coverage problem in over-provisioned networks is to provide k -coverage for all targets using minimum number of sensors, whereas under-provisioned networks are aimed to provide a balanced coverage for all targets, which is due to lack of sensors in these environments. The Q -coverage is needed in networks where the coverage requirement of different targets is met using different numbers of sensors.

In this paper, we address the k -coverage problem in both over-provisioned and under-provisioned networks. To solve this problem we first proposed a GA-based algorithm capable of providing a desirable k -coverage for all targets existing in an over-provisioned network using minimum number of sensor directions. Afterward, we developed the proposed algorithm so that it can provide a balanced coverage in an under-provisioned network as well. Additionally, we developed a greedy-based algorithm capable of solving the k -coverage problem to apply as a criterion for evaluating the performance of the proposed algorithms. This study has the following contributions:

Proposing a GA-based algorithm capable of solving the k -coverage problem in both over provisioned and under provisioned networks; presenting an effective scheme for chromosome representation, which initially provides k -coverage for all the targets and attempts to provide a balanced coverage especially in under-provisioned networks (this is due to allocating appropriate values to critical targets); proposing a repair operator to assure feasibility of the child chromosome returned as the output of the crossover operation; and conducting several experiments to evaluate the proposed algorithms performance.

This paper is organized as follows. Section 2 discusses the related works; Section 3 explains the system model, used terminologies, and the problem addressed in this study; Section 4 introduces the proposed GA-based algorithm in detail; Section 5 reports the experimental results; and finally Section 6 concludes the study.

2. Related work

The aim of designing a sensor network is mainly monitoring and collecting data within a defined area. Target coverage is one of the most important aims since the sensor nodes in such networks are generally scattered in a random way. In a general view, target coverage is the task of monitoring a set of certain targets in a definite area using a set of sensors consuming the minimum amount of energy. One of the most popular approaches to solve the target coverage problem is the use of scheduling technique. Through scheduling the sensors distributed with a high redundancy in a given network, we can save their energy very efficiently. This technique puts some of the sensors in inactive mode (i.e., power saving mode) while keeping the others active in order to do the tasks they are supposed to do. This technique significantly expands the network lifetime because the inactive sensors need a negligible amount of energy; additionally, with frequently oscillating the sensors between their active and inactive modes, their batteries last for a longer time. In this paper, we are mainly focused on solving the target k -coverage problem

using the scheduling technique. In the following, some of the most outstanding studies conducted on this problem using the scheduling technique in WSNs and DSNs are discussed.

In [5], the authors modeled the target coverage problem as disjoint cover set and proved its NP-completeness. In [4], the previously-mentioned study was extended to non-disjoint cover sets in which each sensor can freely join multiple cover sets. In [13,14,19], LAs were also used to solve the target coverage problem in WSNs. They were applied to identify the active sensors capable of properly monitoring the area of interest. All papers discussed above have centered on solving simple target coverage problem; though, in real conditions, targets may need to be monitored by more than only one sensor. In literature, such problem is addressed as k -coverage problem. The authors in [2] attempted to propose a solution to the k -coverage problem in the mission-oriented mobile WSNs using the centralized and distributed approaches. The aim of authors in [2] was to select the minimum number of connected sensor nodes so that the targets can be completely covered by k -coverage. They proposed a connected k -coverage working set construction algorithm on the basis of the Euclidean distance in order to provide k -coverage all over the sensing region and at the same time decrease the number of active sensors as far as possible. In [8], both coverage and connectivity problems were addressed. To solve these problems, the authors introduced a GA-based scheme so that k -coverage can be provided for all targets and m -connectivity for each sensor. The above-mentioned studies offered efficient solutions to target coverage problem in WSNs; though, in DSNs, they have nothing to say. Such inefficiency is due to the limitation on the sensing angle of directional sensors. It has obliged researchers in this field to provide new algorithms to solve the target coverage problem specifically in DSNs.

Several researchers (see [18,19,21]) have proven the effective role of the scheduling technique in solving the target coverage problem in DSNs. MA and Liu [9], the pioneers of DSN research, used sensor deployment strategies in order to meet the requirements of area coverage. In [1], the target coverage problem was studied for the first time using directional sensors with the aim of monitoring maximum number of targets by means of minimum number of active sensors. In [3], the authors proved NP-completeness of the multiple directional cover set problem and proposed a number of heuristic algorithms to form non-disjoint cover sets each of which was able to monitor all the targets in the network. Gil and Han [6] proposed two algorithms, one based on greedy approach and the other one based on genetic algorithm, aiming at maximization of the number of constructed cover sets and management of critical targets. In [11,12,15,17], the use of LA was confirmed as an appropriate approach to solve NP-complete problems and several LA-based algorithms were designed to solve the target coverage problem in DSNs using the scheduling technique. furthermore in [16,20], LAs were used to offer a solution to the priority-based target coverage problem in order to select the appropriate sensor directions to meet the coverage quality requirements of each target existing in the network. In [22], the authors proposed some target-oriented heuristics to solve the target coverage problem in smart camera networks. The heuristics were capable of providing near-optimal coverage and they were shown more successful than the existing sensor-oriented heuristics in solving the same problem.

All studies mentioned above have been mainly focused on solving simple target coverage problem in networks where sensors possessed only a single power level. In recent years, Mohamadi et al. [18] proposed two greedy-based algorithms to solve the target coverage problem in networks comprising sensors with more than one power level (i.e., adjustable sensing ranges). They confirmed that with adjusting sensing ranges, the network

Table 1

Notations.

| Notation | Meaning |
|--------------|---|
| N | Number of sensors |
| M | Number of targets |
| k | The level of required coverage |
| s_n | A sensor, for all $n = 1, 2, \dots, N$ |
| t_m | A target, for all $m = 1, 2, \dots, M$ |
| l_i | Lifetime of sensor s_i |
| S | Set of sensors, $S = s_1, s_2, \dots, s_N$ |
| T | Set of targets, $T = t_1, t_2, \dots, t_M$ |
| $d_{i,j}$ | The j th direction of the i th sensor. (e.g., $d_{1,2}$ (sensor1, direction 2)) |
| $T(d_{i,j})$ | Refers to all the targets covered by sensor direction $d_{i,j}$ |
| w | The number of directions per sensor |
| C_t | The number of sensors that monitor target t_i |

lifetime could be significantly extended. Mohamadi et al. [18] also focused on solving the simple coverage problem where only a single sensor was responsible for covering one target. The main challenge is that targets in real conditions may need to be monitored by more than one sensor. Note that networks with higher coverage scope are normally more reliable since they have further resistance to sensor failures and erroneous sensor measurements that may take place during a monitoring operation. Reliability is a vital factor especially in networks without the opportunity of finding and replacing the failed sensors. As a result, to solve the target coverage problem in a more efficient way, there is a need for proposing new algorithm that can offer reliability and failure tolerance and at the same time maximize the network lifetime.

In [10], the authors introduced the balanced k -coverage problem with the aim of avoiding conditions in which only some of the existing targets are provided with k -coverage while the others have remained uncovered or slightly covered. To solve this problem, they proposed three solutions on the basis of Integer Linear Programming (ILP), Integer Quadratic Programming (IQP), and Integer Non-Linear Programming (INLP). Since they had high computational expense and NP-completeness, the authors introduced a new solution termed Centralized Greedy k -Coverage Algorithm (CGkCA) that was faster in its computational operations. Note that real conditions may involve situations in which some targets might be located in some spots that cannot be simply monitored or only a few or even a single sensor is capable of covering them. Such targets need to be covered first to hold a maximal coverage. As a result, the targets located in such areas must be prioritized carefully according to their situation, then a minimum number of sensors need to be chosen to monitor the targets in order of their priority. For this purpose, we used the concept of critical targets to determine the priorities for each target in the network. To simplify the solutions proposed in [10], avoid local optima, and take into consideration the critical targets, we propose a target-oriented GA-based algorithm capable of solving the target k -coverage problem using minimum number of active sensor directions.

3. Problem definition

Table 1 presents the notations used in this paper. This paper takes the following scenario into consideration. In a 2-D Euclidean field, a number of targets were placed at known locations. Each target in this field needs to be monitored for at least k times. In addition, a number of directional sensors were randomly scattered adjacent to the targets to satisfy their k -coverage requirement. Each directional sensor possesses a number of non-overlapping directions, and only one of its directions needs to be activated at each unit of time, which is recognized as working direction. In this study, we provide each directional sensor with a device to determine the working direction of each sensor from

among its available directions at each unit of time. A target is covered by a directional sensor if it is situated within both the sensing range and field of view of the selected sensor direction. Remember that in the k -coverage problem, each target needs to be covered simultaneously by multiple directional sensors so that its k -coverage requirement can be fully satisfied. Initially, the battery power of all sensors is the same, and it is generally assumed that the batteries are nonchargeable.

Problem. How to select a minimal subset of sensor directions capable of providing all targets with k -coverage.

Definition 1. A cover set comprises a subset of sensor directions that is able to provide k -coverage for all the targets in the network.

Definition 2. A target is critical if it is monitored by the minimum number of sensor directions.

To elaborate the problem in hand, let us give an example. Assume that we have the example network depicted in Fig. 1. It comprises five directional sensors and four targets; each target needs to be covered by two directional sensors. Remember that the environment of interest in our example is two dimensional, in which a directional sensor can monitor a target if the target is in both its sensing range and in field of view. As can be seen in Fig. 1.B, the sensor directions monitoring each target are as follows. $t_1 = \{d_{1,2}, d_{2,3}\}$, $t_2 = \{d_{1,2}, d_{2,1}, d_{4,3}\}$, $t_3 = \{d_{4,3}, d_{5,2}, d_{1,1}\}$ and $t_4 = \{d_{3,1}, d_{4,2}, d_{6,3}\}$. (e.g., $d_{1,2}$ stands for sensor 1 direction 2). To hold a simple coverage, a cover set can be constructed through selecting and activating three directional sensors $\{d_{1,2}, d_{4,3}, d_{3,1}\}$. However, if k -coverage is desirable (e.g., $k=2$), the cover set already constructed for the simple coverage needs to be added with some new directional sensors in a way to provide all the targets in the network with k -coverage (for example, $\{d_{1,2}, d_{2,3}, d_{4,3}, d_{5,2}, d_{3,1}, d_{6,3}\}$). As a result, to have the best cover set capable of providing k -coverage, it is very important to choose appropriate sensor directions.

4. Proposed GA-based algorithm

In this section, we first generally review GA; then, we introduce two target-oriented solutions to solve the target k -coverage problem in DSNs. According to [7], GA is a meta-heuristic approach that can be used to solve various optimization problems. Generally defining, GA mimics the processes take place during the natural evolution (on the basis of The Theory of Evolution proposed by Darwin). This algorithm makes use of three operations (i.e., selection, crossover, and mutation) in order to produce successive generations of solutions with higher quality. The main aim of GA is to achieve optimization, i.e., setting different parameters in such a way that the performance of an algorithm in solving a given problem can be improved. The key advantage of GA is rising the chance of reaching global optima and simultaneously avoiding local optima. GA initially produces a population of possible solutions. Each solution is signified by a set of genes, which is called chromosome or individual. For the purpose of each certain problem, a fitness function is defined aiming at testing the performance of each chromosome. Fitness of a chromosome depends on how well that chromosome performs its task in solving the problem in hand. After the initial population is produced, the algorithm performs three operations of selection, crossover and mutation. Through the first operation, the algorithm selects a set of possible solutions from among the initial population. Then, two selected chromosomes (called parents) are used to create two child chromosomes through crossover operation where the

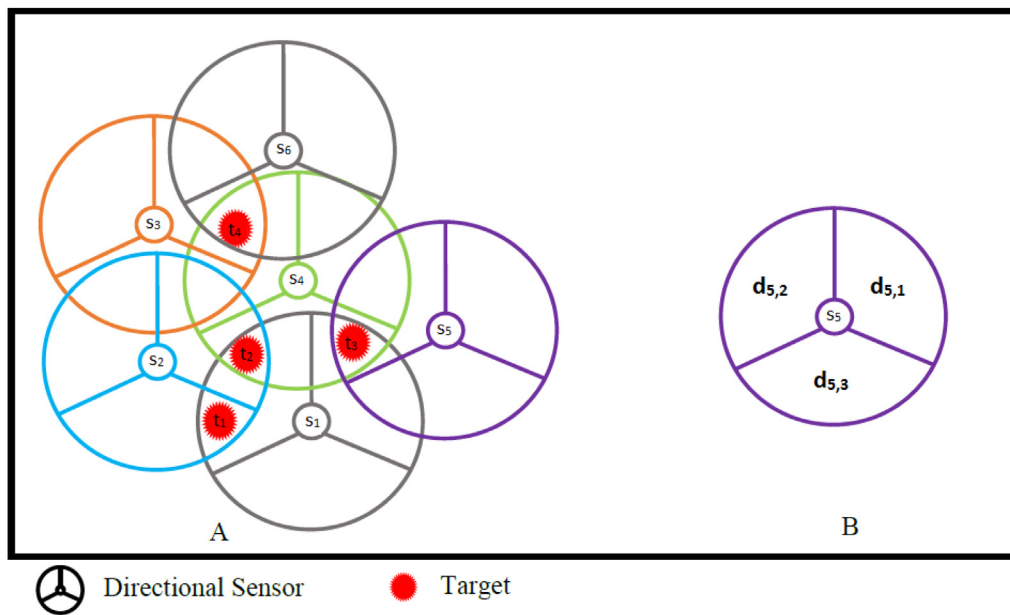


Fig. 1. (A) Example network with six directional sensors and four targets (B) A directional sensor with three directions.

genetic information is exchanged between the parent chromosomes. Next, the child chromosomes do the mutation operation to produce a better solution. When this operation is completed, the algorithm applies the fitness function again to measure the child chromosomes. The values of these chromosomes are compared to those of all the chromosomes that belong to previous generation. In the final step, from among the initial population and the population generated through the crossover and mutation operations, those with better fitness values are chosen as the coming generation.

4.1. GA-based algorithm in over provisioned networks

In this section, we are going to solve the k -coverage problem in DSNs. Generally, this problem can be considered in two different environments: over provisioned and under provisioned. The former refers to the networks containing enough sensors to provide all the targets with desirable k -coverage, while the latter refers to the networks that lack sufficient sensors to do the coverage task, thus there is a need for managing the coverage (which is recognized as imbalanced k -coverage problem). Note that k -coverage is important to both environments. In the under-provisioned networks, k -coverage is mainly aimed to provide a balanced coverage; in other words, we have to provide uniformly the k -coverage for all targets in the network. In the over-provisioned networks, we have to not only maximize the coverage, but also minimize the number of sensors in constructed cover sets so that energy consumption can be managed as much as possible. In the following, we propose a target-oriented GA-based algorithm capable of solving the k -coverage problem in both above-mentioned environments. Here, we first explain the details of the proposed algorithm in an over-provisioned environment; then, we will explain how it differs when it is applied to an under-provisioned one.

Here, we explain generally the entire process of constructing a cover set. In over-provisioned networks, the network operation falls into a number of rounds each of which results in a cover set capable of providing a desirable coverage for all the targets in the network. In each round of the algorithm, to construct an appropriate cover set, we make use of a GA-based algorithm that is explained in the following. When a cover set is returned as the

output of the algorithm, it is assigned with a working time value (a fixed time) that is then added to the network lifetime. After that, energy of the sensors is updated and the sensors with no energy are removed from the list of available sensors. The process of cover set construction continues until the available sensors are capable of providing all targets with desirable coverage in an over-provisioned network. The proposed GA-based algorithm involves mainly the following steps.

Representation: To solve a problem using a GA-based algorithm, one of the most significant steps is to model the problem in hand. That is, we need to show the available solutions in the problem space in the form of a chromosome. In this paper, we use an integer-based representation to model a chromosome. In this model, a chromosome stands for a cover set and each gene of a chromosome signifies a working direction of a sensor chosen to monitor a given target from among all sensor directions covering it. A 2-D array (matrix) is used to encode each chromosome. The matrix comprises M columns and k rows, where M denotes the number of targets existing in the network, whereas k represents the coverage required for all the targets. The reason we consider M for the length of the chromosome, which actually equals the number of targets in the network, is that literature shows designing target-oriented algorithms have led to a better performance compared to sensor-oriented ones in terms of solving the problem in hand [22] and also in random environments, the number of targets is usually smaller than sensors, which causes the proposed model to hold chromosomes with shorter length, hence using less memory. The problem addressed in this study is k -coverage where each target needs to be monitored by multiple sensor directions; this is why for each gene, the repeated values must be avoided. In other words, the set of sensor directions, which is chosen to provide a target with k -coverage and taken into account as the value of the gene corresponding to that target, must be non-repetitive. Additionally, note that in DSNs, at a given time, only a direction of a sensor is applicable to the formation of a cover set. As a result, after selecting a direction of a sensor to assign a value to the gene corresponding to a target, the other directions of that sensor cannot be used for the same task in case of the other targets. Modeling a chromosome in this way leads to the production of a valid chromosome, which refers to a chromosome that can provide full k -coverage for all the targets in

| Target Number | t_1 | t_2 | t_3 | t_4 |
|---------------|----------|----------|----------|----------|
| Gene Values | d_{12} | d_{12} | d_{43} | d_{31} |
| | d_{23} | d_{43} | d_{52} | d_{63} |

Fig. 2. An example chromosome for the network.

the network. When a chromosome is going to be created by the algorithm (through assigning a value to its genes); first, the most critical target needs to be marked out; afterwards, appropriate values are assigned to the genes of that target. In other words, the algorithm chooses a subset of sensor directions from among the ones monitoring the target to assign a value to each gene corresponding to it. When a sensor direction is chosen to assign a value to a gene corresponding to the critical target, the selected sensor direction cannot be chosen again to assign value to other genes of that target. The process of identifying a critical target and assigning values to its genes continues until the genes corresponding to all the targets are assigned with appropriate values. The reason of adopting such strategy is that some of the targets in a real network may be situated in such an area that only a single or few sensor directions can monitor them. For the purpose of maximizing the coverage, these targets (critical targets) need to be monitored before the others. It demands an algorithm to prioritize the targets concerning their coverage vulnerability. Then, it needs to choose minimum number of sensor directions in such a way to monitor the targets based on their priority.

Here, an example is presented to elaborate the proposed model as much as possible. As depicted in Fig. 1, let us assume that there are four targets and the coverage required to cover all the targets is $k = 2$. As a result, we use a matrix 2×4 to model each chromosome (see Fig. 2). Genes of each column represent the status of the target corresponding to that column. To produce a chromosome (by assigning appropriate values to its genes), the most critical target must be marked out first. In the example network, as target t_1 is monitored by minimum number of sensor directions, it is considered as the first critical target; thus, a value is assigned to its first-level gene. In other words, the algorithm selects a sensor direction from among those monitoring that target. The selected sensor direction is removed temporarily from the list of the sensor directions monitoring the critical target; then, the algorithm selects a sensor direction to assign a value to the second-level gene of that target. Remember that when a sensor direction is chosen to monitor the critical target, if the other directions of the same sensor are present in the list of the directions monitoring other targets, they will be eliminated. Let us assume that in the first step, to cover the critical target, sensor directions $d_{1,2}$ and $d_{2,3}$ are selected. In this condition, other directions of the selected sensors (i.e., $\{d_{1,1}, d_{1,3}, d_{2,1}, d_{2,2}\}$) are removed temporarily if they are in the list of sensor directions monitoring other targets. This strategy avoids the selection of more than one direction from a single sensor. The process of selecting a critical target and assigning value to its genes continues until the chromosome in hand is formed entirely.

Initial population: A set of chromosomes produced randomly, creates the initial population where each chromosome is denoted by a set of integer numbers. The initial population generation

algorithm is presented in Algorithm 1. In the present paper, we set the number of initial population to 60.

Fitness: On algorithm 2 line 4, in a general view, fitness value measures the quality of a chromosome according to the objectives defined in each case. Problems in this area of study are divided into two classes: minimization and maximization. In the former, the chromosomes with lower fitness function are desirable, while in the latter, those with higher fitness function are desirable. The algorithm proposed in this paper is mainly aimed to choose a minimum number of active sensor directions from among available ones in a way to provide k -coverage for all the targets in the network. Note that the cover sets containing fewer active sensor directions actually obtain a better fitness function. Therefore, the proposed algorithm attempts to find candidate solutions comprising minimum number of active sensor directions, which can ultimately lead to extension of the network lifetime. For example, the fitness function of the chromosome illustrated in Fig. 2 is 6.

$\text{Fitness}(\text{chromosome}) =$ The number of unique sensor direction in chromosome is considered.

Selection: On algorithm 2 line 6, in the process of selection, the algorithm chooses valid chromosomes using some methods including roulette-wheel selection, rank selection, and tournament selection. The chromosomes showing better (lower) fitness value have more chance to be chosen. We apply the roulette-wheel selection method to the algorithm proposed in this study. The selected chromosomes produce new child chromosomes through the crossover operation.

Crossover and mutation: On algorithm 2 line 7, to create the next generation, the crossover operation is applied to the two chromosomes randomly chosen from among current population. Literature contains various types of crossover operation such as uniform crossover, single-point crossover, and double-point crossover. To the present study, the single-point crossover is applied, in which the crossover point is randomly selected. Then, the two parent chromosomes get involved in exchanging their information (see Fig. 3). Note that in DSNs, at a given time, only one direction of a sensor can be manipulated. As a result, following the crossover operation, invalid chromosomes can be also created. Invalidity of a chromosome is confirmed if in assigning values to its genes, multiple directions of one sensor are employed. In addition, a repair operator is proposed in this study aiming at changing an invalid chromosome to a valid one. Once valid chromosome is generated, the mutation operation is started, in which a gene of the chromosome is randomly chosen and its value is changed. In other words, the value of a gene that is corresponding to a given target is changed to a value chosen from among the other sensor directions covering the target. In the mutation operation, two issues need to be considered well in order to avoid the production of invalid chromosome. First, when a sensor direction is selected to allocate the value to a gene, the value needs to be compared to the values of other genes of a

Algorithm 1: Generation-of-Initial-Population.

Input: 1) Number of targets, M , The required coverage, k .
 2) List of minimal adjusted sensor direction covering each target, $CoverageList$.
 3) Size of initial population $nPop$, i.e., number of chromosomes.

Output: Initial Population.

```

1:  $InitialPopulation \leftarrow \emptyset$ ;
2: Integer  $Chromosome[M]$ ;
3: for  $ii = 1$  to  $nPop$ 
4:   for  $jj = 1$  to  $M$ 
5:     find a critical target,  $T_c$ 
6:     for  $kk = 1$  to  $k$ 
7:        $Chromosome[ii, T_c, kk] = rand(CoverageList(T_c))$ 
8:     end for
9:   end for
10:  $InitialPopulation \leftarrow InitialPopulation \cup Chromosome$ 
11: end for

```

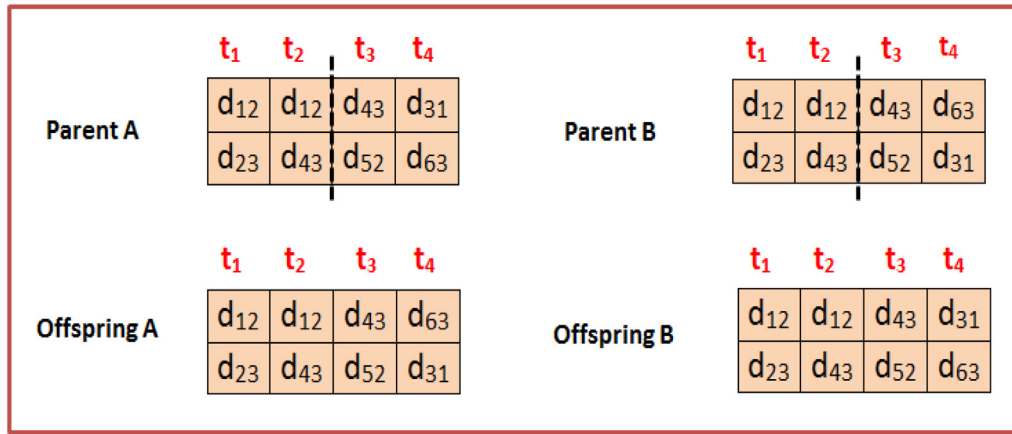


Fig. 3. An example of single point crossover operation.

chromosome. If the selected value does not cause the invalidity of the chromosomes, it is applied; otherwise, a new value is chosen. Second, when the mutation operation is going on, after changing the value of a gene, it is of a high significance to have various values for the genes that are corresponding to a given target. From the crossover and mutation operations, a set of chromosomes is resulted; each chromosome is denoted by a matrix. Each matrix is able to provide all targets with full k -coverage. As a result, in each iteration, validity of the chromosomes is not needed to be checked.

Repair operator: On algorithm 2 line 8, we propose a repair operator in the present paper to assure feasibility of the child chromosome returned as output of the crossover operation. To this end, this operator checks to assure that there is not more than one direction of a single sensor in a chromosome; if this happens, it changes the value of the gene of that chromosome. Note that values of the genes corresponding to one target need to be different with each other. When the child chromosome resulted from the crossover operation is returned, its validity is checked. If it is invalid (see Fig. 4), at first, the repair operator finds the list of all directions cannot be used as values of the genes of the chromosome. This list (also known as dominated list) holds the other directions of those sensors got already involved with one of their directions in the formation of the chromosome. In the next step, the values of the genes corresponding to the target in hand are added to the dominated list to avoid allocating the

same values to the genes corresponding to the target. Then, the list of all sensor directions covering the target (the one under the repair operation) is provided and termed as covering list. Afterward, to allocate a value to the gene of the target in hand, the difference between the covering list and the dominated list is found and one sensor direction is selected from among them. For instance, according to Fig. 4 two chromosomes were selected and crossover operation was done on them which resulted in an invalid chromosome. repair operator checks and corrects it.

Stopping criterion: On algorithm 2, the proposed algorithm comes to an end once maximum number of consecutive iterations is achieved.

4.2. GA-based algorithm in under provisioned networks

This section explains the way the proposed algorithm works in an under-provisioned network. In this new state, similar to the over provisioned, we make use of an integer-based representation to model the chromosomes (i.e., matrix $m \times k$). Each chromosome denotes a set of sensor directions selected to provide maximum balanced coverage. In this model, each gene of a chromosome stands for a sensor direction selected to cover a certain target, from among all sensor directions monitoring the target. Note that as the environment we work on is under provisioned, if there are not sufficient sensor directions to provide a desirable coverage for a certain target, we assign null-value to its gene.

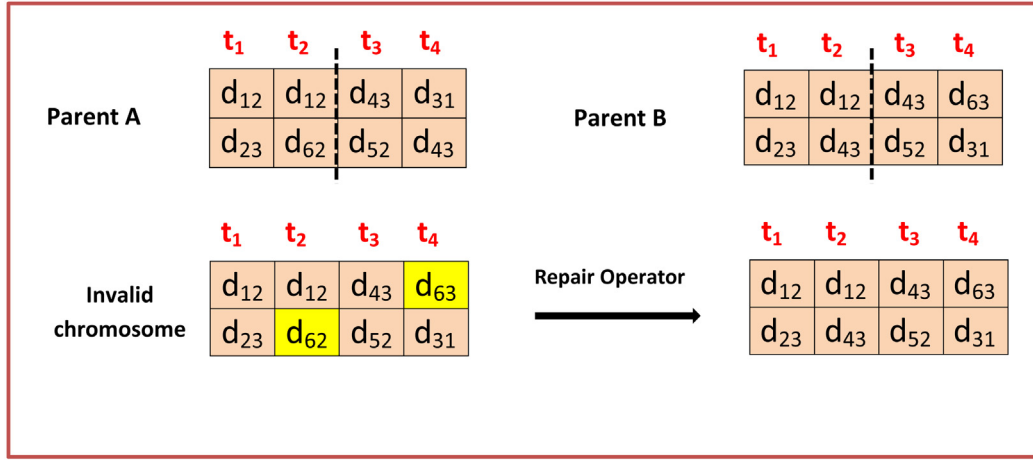


Fig. 4. An example of invalid chromosome from crossover operation.

Algorithm 2: GA-based k -Coverage Algorithm in over provisioned networks.

01. **Input:** Directional network $Net = (S, T)$, The level of required coverage, k
02. **Output:** A collection of appropriate sensor directions, C
03. Initialize population Pop;
04. Evaluate Pop;
05. While (not terminated)
06. $P_s = \text{Select}(\text{Pop})$;
07. $P_c = \text{Crossover}(P_s)$;
08. $P_c = \text{Repair}(P_c)$;
09. $P_m = \text{Mutate}(P_c)$;
10. $P' \leftarrow P_c \cup P_m$;
11. Evaluate P' ;
12. $\text{Pop} = \text{Survival}(\text{Pop}, P')$;
13. End while
14. C = the best chromosome return by the GA algorithm in Pop set;
(Minimum number of unique active sensor directions).
15. **return** (C)

In this networks, to have maximum balanced coverage, at the time of creating a chromosome, it is better to first select the targets covered by fewer sensor directions and allocate a value to their genes. To this end, similar to the algorithm proposed formerly for over-provisioned networks, we apply the concept of critical targets. This enables us to provide a desirable coverage for those targets covered less than others, hence having a maximum balanced coverage. After modeling the problem, we form randomly an initial population of the chromosomes. Similar to the over provisioned networks, in under-provisioned ones, we need a metric (an appropriate cost-function in GA) to make judgment on superiority of one solution to another one. To this end, we make use of an appropriate metric termed Balancing Index (BI) (see [10] for more details) as our cost-function. It takes into account both the fairness of coverage and maximization of the total coverage. In this study, higher value of BI shows a better coverage. Eq. (1) shows BI in a mathematical way.

$$BI = \frac{(\sum_{t=1}^M C_t)^2}{M * \sum_{t=1}^M C_t^2} * \frac{\sum_{t=1}^M C_t}{kM} \quad (1)$$

Here, C_t is the number of sensors that monitor target t_i and M denotes the total number of targets in the network. For example, as can be seen in the following figure, a 3-coverage problem

($k=3$) is taken into consideration. The BI was obtained for the two different conditions presented in Fig. 5(a). and (b).

$$BI = \frac{(2 + 2 + 2 + 2)^2}{4 * (2^2 + 2^2 + 2^2 + 2^2)} * \frac{(2 + 2 + 2 + 2)}{3 * 4} = 0.67$$

$$BI = \frac{(2 + 1 + 1 + 3)^2}{4 * (2^2 + 1^2 + 1^2 + 3^2)} * \frac{(2 + 1 + 1 + 3)}{3 * 4} = 0.48$$

Therefore, this metric is suitable for recognizing the optimal coverage. Higher BI results in a better coverage.

In algorithm 3 also the reproduction operations (selection, crossover, and mutation) are done until the algorithm meets its termination criterion or ($BI > t_q$). the parameter t_q is set to a number less than 1. Since the problem is to solve the coverage problem in under provisioned directional sensor networks, so the algorithm is not able to k -coverage targets and BI index does not reach 1. Therefore, in this algorithm, t_q is set to a number less than 1, for example 0.9 (0 to 0.99). When the parameter reaches this value, the algorithm returns the set as the coverage set. If t_q does not reach this number, the algorithm runs the same as Algorithm 2 and the final step, a set of sensor directions able of providing maximum balanced coverage is returned as the output of the algorithm.

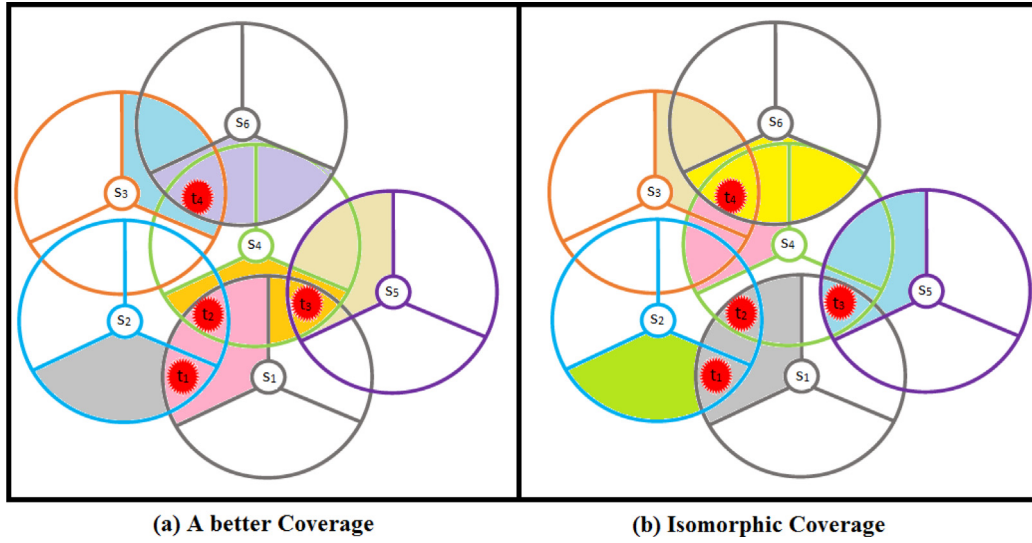


Fig. 5. Balancing Index.

Algorithm 3: GA-based k -Coverage Algorithm in under provisioned networks.

01.**Input:** Directional network $Net = (S, T)$, The level of required coverage, k
 02.**Output:** A collection of appropriate sensor directions, C
 03. Initialize population Pop;
 04. Evaluate Pop;
 05. Let t_q denote the dynamic threshold.(e.g., $t_q=0.9$).
 06. While (not terminated)
 06. $P_s = \text{Select}(\text{Pop})$;
 07. $P_c = \text{Crossover}(P_s)$;
 08. $P_c = \text{Repair}(P_c)$;
 09. $P_m = \text{Mutate}(P_c)$;
 10. $P' \leftarrow P_c \cup P_m$;
 11. Evaluate P' ;
 12. $\text{Pop} = \text{Survival}(\text{Pop}, P')$;
 13. Compute the balancing index (i.e., BI) based on **equation 1**
 14. **if** $BI > t_q$ **then**
 15. Exit;
 16. End while
 17. C = the best chromosome return by the GA algorithm in Pop set;(Maximum value of BI index).
 18. **return** (C)

5. Simulation results

This section evaluates the efficiency of the proposed algorithm in two states, under provisioned and over provisioned. To this end, different scenarios were made to test the effects of various parameters on the algorithm performance. In over-provisioned scenarios, we used the network lifetime as the comparison criterion, while in case of under-provisioned ones, this criteria was the balancing index. To take into consideration the algorithm efficiency more accurately, it was compared with a greedy-based algorithm recently proposed in literature [10]. To make a network, we randomly scattered several targets in a field of $500(m) \times 500(m)$, then we distributed a number of sensors with sensing range r and sensing angle $\frac{2\pi}{3}$ to monitor the targets. The number of targets, sensors and sensing range was set to 10,150 and 100 respectively by default. In addition, the k -value was fixed at 3. To have a high reliability in the obtained results, each scenario was run for 10 times and the average value was returned as the final

output of the algorithm. In the experiments done in this study, the population size was fixed at 50, and the rate of crossover and mutation was set to 0.2 and 0.05, respectively. The experiments 1 to 4 test the algorithm performance in over-provisioned state, whereas the experiment 5 tests it in the under-provisioned one.

Experiment 1. In this experiment, we studied the effect of k -value on the network lifetime. For this purpose, we changed the k -value from 1 to 5, with incremental step 1. The results shown in Fig. 6 indicated that an increase in k -value leads to reduction in the network lifetime. The reason was that with increasing k -value, it is needed to select more sensor directions to provide k -coverage for all targets in the network (see Fig. 7). It is reasonable to say that the more sensor directions are used, the more energy is needed to consume, hence decreasing the network lifetime. The results obtained from the proposed GA-based algorithm were compared with those of the greedy-based one, and the comparative results confirmed the superiority of the former over

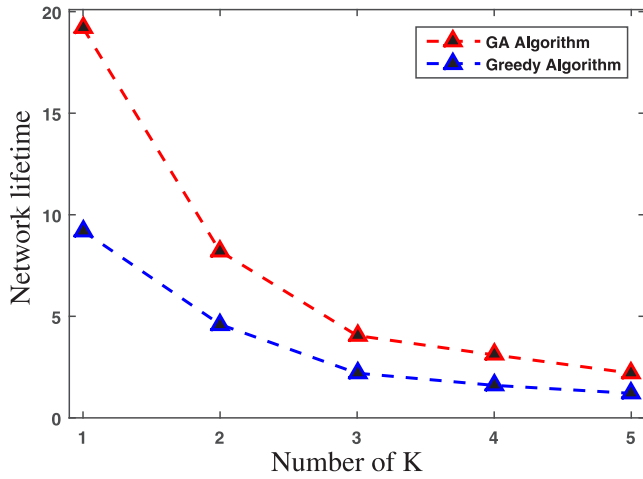


Fig. 6. Effect of the number of required coverage on the network lifetime.

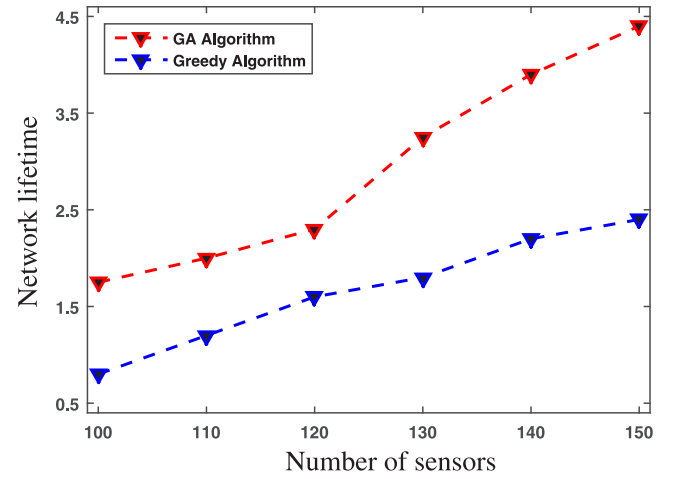


Fig. 8. Effect of the number of sensors on the network lifetime.

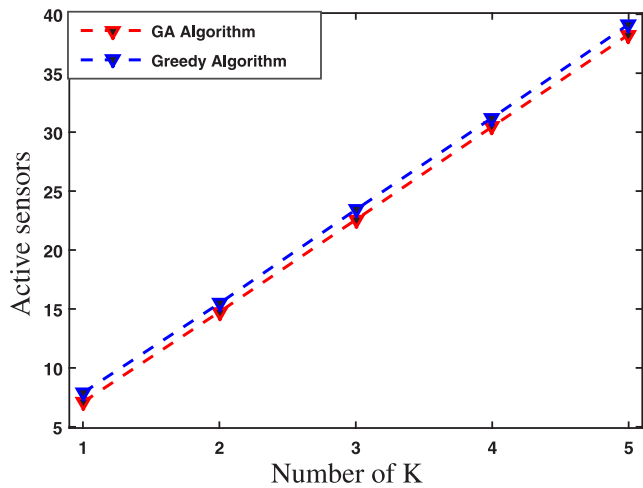


Fig. 7. Effect of the number of required coverage on the number of active sensors.

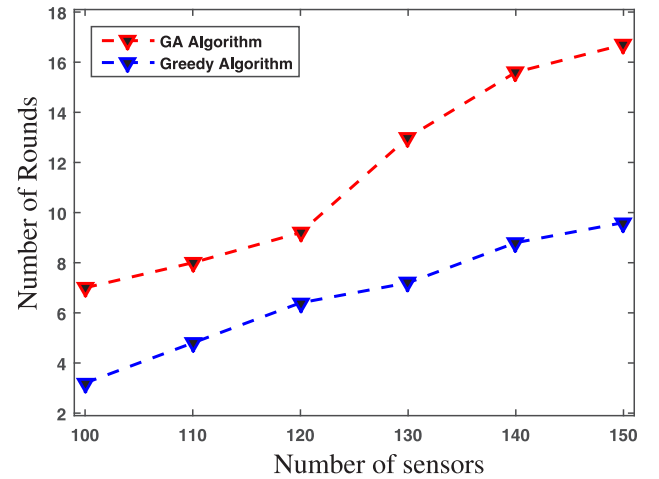


Fig. 9. Effect of the number of sensors on the number of rounds.

the latter in terms of prolonging the network lifetime. This superiority is mainly due to managing the critical targets; the issue that is not taken into account by the greedy-based algorithm.

Experiment 2. In this experiment, the number of sensors was varied between 100 and 150 with incremental step 10 in order to examine the effect of the number of sensors on the network lifetime. Based on the obtained results (see Fig. 8), with more sensors, the network lifetime prolonged further. The reason was that in this condition, the algorithm was able to run more rounds, which resulted in creating more cover sets, hence increasing the network lifetime (see Fig. 9). As confirmed by the results, in the condition of this experiment, the proposed GA-based algorithm was more successful than the greedy-based one in terms of extending the network lifetime.

Experiment 3. In this experiment, we investigated the effect of the number of targets on the network lifetime. To find this relationship, we reset the number of targets between 10 and 20, with incremental step 2. The results presented in Fig. 10 show that increasing the number of targets led to a decrease in the network lifetime. It was due to the fact that with increasing the targets, we needed more sensors to provide k -coverage for all the targets in the network. In this condition, more sensors were needed to construct the necessary cover sets (see Fig. 11); which finally diminished the network lifetime. The results presented in

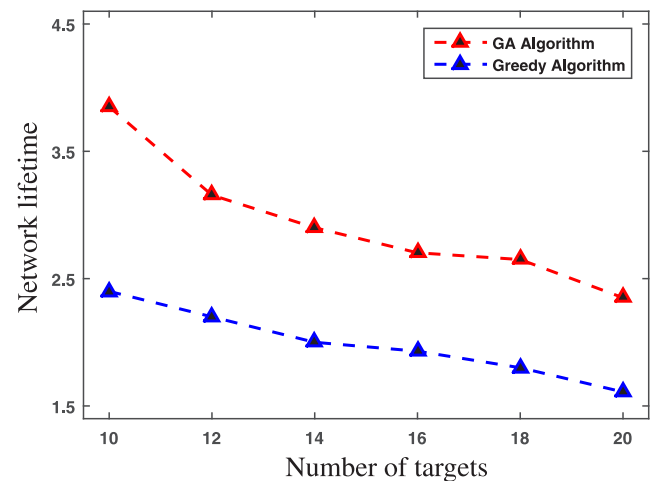


Fig. 10. Effect of the number of targets on the Network lifetime.

Figs. 10 and 11 show that the proposed algorithm achieved more network lifetime compared to the greedy-based one.

Experiment 4. In this experiment, we changed the sensing range from 80 to 120 with incremental step 10 in order to find

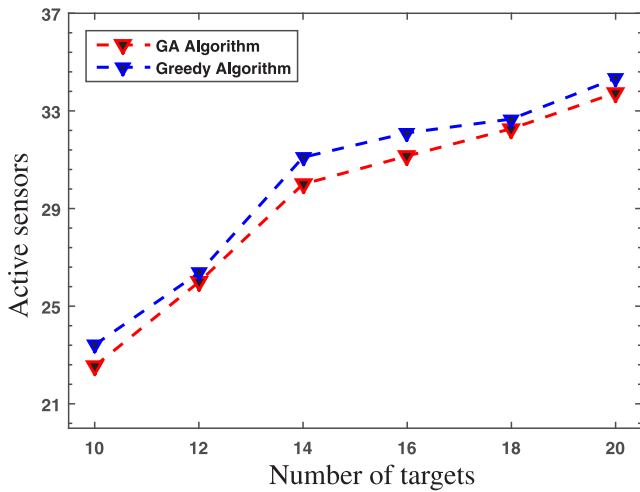


Fig. 11. Effect of the number of targets on the number of active sensors.

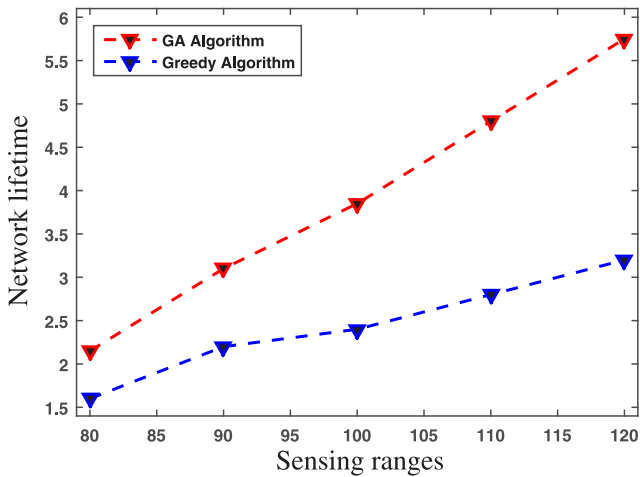


Fig. 12. Effect of sensing range on the network lifetime.

the relationship between sensing range of the sensors and the network lifetime. As shown by the results presented in Fig. 12, there is a direct relationship between these two parameters; that is, higher sensing range leads to a more prolonged network lifetime. The reason is that in this condition, sensors are able to cover more targets; thus, we need fewer sensor directions to construct cover sets that are capable of providing all the targets in the network with full coverage (see Fig. 13).

Experiment 5. In this experiment, we tested the efficiency of the algorithm in an under-provisioned environment. To this end, we set the number of sensors between 20 and 40 with incremental step 10 and k -value between 2 and 4, with incremental step 1. Here, the number of targets and the sensors' sensing radius were fixed at 10 and 100 respectively. Table 2 shows the balancing index obtained by different algorithms. As can be seen, the proposed algorithm is able to achieve a balancing index more close to the optimal value compared to the greedy-based one. In other words, the proposed algorithm is capable of making a higher balance in providing desirable coverage for all targets in the network. Table 3 shows the number of targets that have not been monitored at all. As confirmed by the obtained results, the proposed GA-based algorithm outperformed the greedy-based one in terms of expanding coverage on the field of interest as much as possible. Table 4 shows the number of targets provided with

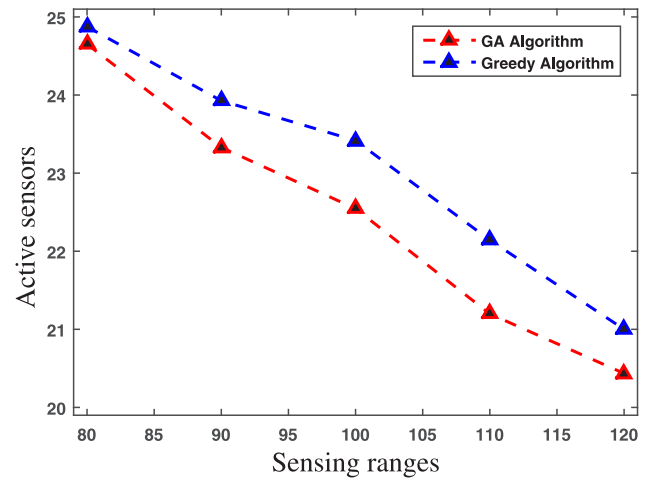


Fig. 13. Effect of sensing range on the number of active sensors.

Table 2
Effect on Balancing Index.

| Number of Sensor | N = 20 | | N = 30 | | N = 40 | |
|------------------|---------|---------|---------|---------|---------|---------|
| | GA | Greedy | GA | Greedy | GA | Greedy |
| K = 2 | 0.92539 | 0.63182 | 0.97556 | 0.81667 | 0.98999 | 0.9205 |
| K = 3 | 0.77772 | 0.56559 | 0.95332 | 0.71614 | 0.98041 | 0.82869 |
| K = 4 | 0.60510 | 0.45942 | 0.88952 | 0.57597 | 0.96770 | 0.63845 |

Table 3
Number of uncovered targets in Under-provisioned Network.

| Number of Sensor | N = 20 | | N = 30 | | N = 40 | |
|------------------|--------|--------|--------|--------|--------|--------|
| | GA | Greedy | GA | Greedy | GA | Greedy |
| K = 2 | 0.4 | 0.6 | 0.2 | 0.2 | 0.1 | 0.1 |
| K = 3 | 0.4 | 0.8 | 0.2 | 0.3 | 0.1 | 0.1 |
| K = 4 | 0.4 | 1 | 0.2 | 0.4 | 0.1 | 0.1 |

Table 4
Number of covered targets in Under-provisioned Network.

| Number of Sensor | N = 20 | | N = 30 | | N = 40 | |
|------------------|--------|--------|--------|--------|--------|--------|
| | GA | Greedy | GA | Greedy | GA | Greedy |
| K = 2 | 8.5 | 8.1 | 9.7 | 9.3 | 9.7 | 9.8 |
| K = 3 | 5.6 | 5.5 | 8.5 | 8.3 | 9.7 | 9.4 |
| K = 4 | 3.4 | 3.3 | 6.6 | 6.2 | 8.7 | 8.2 |

desirable coverage. The results indicated that in both algorithms, the more sensors distributed, the more targets received their desirable coverage. Generally, different experiments conducted in this study resulted in the fact that the proposed algorithm outperformed the greedy-based one in terms of achieving a close-to-optimal balancing index value and having fewer uncovered targets in the network of interest.

6. Conclusion

In this study, we addressed the k -coverage problem in both over-provisioned and under-provisioned environments, and to solve this problem, we proposed a GA-based scheduling algorithm. The proposed algorithm in the over-provisioned networks was able to form cover sets containing minimum number of sensor directions in a way to provide the targets in a DSN with full k -coverage, while in the under-provisioned networks, it was capable of providing a maximum balanced coverage for all the targets in the network. To solve the problem in hand using GA, we had some innovations, including the use of a suitable chromosome representation to model the problem, considering the

critical targets while producing the chromosomes, and designing a repair operator capable of changing invalid chromosomes to valid ones. several experiments were conducted to check the performance of the proposed algorithms in both environments and the results were compared to those of a greedy-based one proposed already in literature. The final results showed that the proposed algorithms were more successful than its rival (the greedy-based one) in terms of solving the problem in hand. In other words, the proposed algorithm is capable of achieving not only a higher network lifetime in over provisioned environments, but also a higher balancing index in under-provisioned ones. This is mainly due to managing the critical targets in the network.

CRediT authorship contribution statement

Abolghasem Alibeiki: Conceptualization, Software, Formal analysis, Investigation, Data curation, Writing - review & editing, Writing - original draft. **Homayun Motameni:** Validation, Resources, Visualization, Writing - review & editing. **Hosein Mohamadi:** Conceptualization, Methodology, Writing - original draft, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] J. Ai, A. Abouzeid, Coverage by directional sensors in randomly deployed wireless sensor networks, *J. Comb. Optim.* 11 (2006) 21–41.
- [2] H.M. Ammari, On the problem of k-coverage in mission-oriented mobile wireless sensor networks, *Comput. Netw.* 56 (2012) 1935–1950.
- [3] Y. Cai, W. Lou, M. Li, X.-Y. Li, Energy efficient target-oriented scheduling in directional sensor networks, *IEEE Trans. Comput.* 58 (2009) 1259–1274.
- [4] M. Cardei, M.T. Thai, L. Yingshu, W. Weili, Energy-efficient target coverage in wireless sensor networks, in: *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 2005*, pp. 1976–1984.
- [5] M. Cardei, J. Wu, M. Lu, Improving network lifetime using sensors with adjustable sensing ranges, *Int. J. Sens. Netw.* 1 (2006) 41–49.
- [6] J.-M. Gil, Y.-H. Han, A target coverage scheduling scheme based on genetic algorithms in directional sensor networks, *Sensors* 11 (2011) 1888–1906.
- [7] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co. Inc., 1989.
- [8] S.K. Gupta, P. Kuila, P.K. Jana, Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks, *Comput. Electr. Eng.* 56 (2016) 544–556.
- [9] H. Ma, Y. Liu, On coverage problems of directional sensor networks, *Lect. Notes Comput. Sci.* 3794 (2005) 721–731.
- [10] S.M.B. Malek, M.M. Sadik, A. Rahman, A on balanced k-coverage in visual sensor networks, *J. Netw. Comput. Appl.* 72 (2016) 72–86.
- [11] H. Mohamadi, A. Ismail, S. Salleh, A learning automata-based algorithm for solving coverage problem in directional sensor networks, *Computing* 1 (2013) 1–24.
- [12] H. Mohamadi, A. Ismail, S. Salleh, Utilizing distributed learning automata to solve the connected target coverage problem in directional sensor networks, *Sensors Actuators A* 198 (2013) 21–30.
- [13] H. Mohamadi, A. Ismail, S. Salleh, Solving target coverage problem using cover sets in wireless sensor networks based on learning automata, *Wirel. Pers. Commun.* 75 (2014) 447–463.
- [14] H. Mohamadi, A. Ismail, S. Salleh, A. Nodehi, Learning automata-based algorithms for finding cover sets in wireless sensor networks, *J. Supercomput.* 66 (2013) 1533–1552.
- [15] H. Mohamadi, A. Ismail, S. Salleh, A. Nodehi, Learning automata-based algorithms for solving the target coverage problem in directional sensor networks, *Wirel. Pers. Commun.* 73 (2013) 1309–1330.
- [16] H. Mohamadi, S. Salleh, A.S. Ismail, A learning automata-based solution to the priority-based target coverage problem in directional sensor networks, *Wirel. Pers. Commun.* 79 (2014) 2323–2338.
- [17] H. Mohamadi, S. Salleh, A.S. Ismail, S. Marouf, Scheduling algorithms for extending directional sensor network lifetime, *Wirel. Netw.* 21 (2) (2015) 611–623, <http://dx.doi.org/10.1007/s11276-014-0808-9>.
- [18] H. Mohamadi, S. Salleh, M. Norsyarizad Razali, Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges, *J. Netw. Comput. Appl.* 46 (2014) 26–35.
- [19] H. Mohamadi, S. Salleh, M.N. Razali, S. Marouf, A new learning automata-based approach for maximizing network lifetime in wireless sensor networks with adjustable sensing ranges, *Neurocomputing* 153 (2015) 11–19, <http://dx.doi.org/10.1016/j.neucom.2014.11.056>.
- [20] M.N. Razali, S. Salleh, H. Mohamadi, Solving priority-based target coverage problem in directional sensor networks with adjustable sensing ranges, *Wirel. Pers. Commun.* 95 (2017) 847–872.
- [21] A. Singh, A. Rossi, A genetic algorithm based exact approach for lifetime maximization of directional sensor networks, *Ad Hoc Netw.* 11 (2013) 1006–1021.
- [22] H. Zannat, T. Akter, M. Tasnim, A. Rahman, The coverage problem in visual sensor networks: A target oriented approach, *J. Netw. Comput. Appl.* 75 (2016) 1–15.
- [23] C. Zhu, C. Zheng, L. Shu, G. Han, A survey on coverage and connectivity issues in wireless sensor networks, *J. Netw. Comput. Appl.* 35 (2012) 619–632.



Abolghasem Alibeiki 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 2002. 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.