Pervasive and Mobile Computing 52 (2019) 1–12

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

A new genetic-based approach for maximizing network lifetime in directional sensor networks with adjustable sensing ranges

Abolghasem Alibeiki [a], Homayun 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 30 May 2018
Received in revised form 1 August 2018
Accepted 25 October 2018
Available online 9 November 2018

Keywords:
Directional sensor networks
Cover set formation
Scheduling algorithms
Genetic algorithm

ABSTRACT

In recent years, the directional sensor networks have been attractive to researchers due to their wide and different applications. These networks normally contain a number of self-configurable directional sensors holding adjustable spherical sectors with limited angle. One of the most significant problems in such networks is how to monitor the targets scattered in these networks using sensors with adjustable sensing range and, at the same time, maximize the network lifetime. This problem is recognized as Maximum Network Lifetime With Adjustable Ranges; it has been already proved as an NP-complete problem. As an efficient solution to this problem, the present paper proposes a target-oriented GA-based algorithm that can form cover sets comprising sensors with appropriate directions and sensing ranges in a way to desirably monitor all targets in the network. We examined the efficiency of the proposed algorithm by comparing its obtained results with those of a greedy-based one introduced recently in literature. The comparative results confirmed the efficient performance of the proposed algorithm and also its superiority over the greedy-based algorithm in terms of extending the network lifetime.

1. Introduction

Recently, directional sensor networks (DSNs) have attracted more and more the researchers' attention because of wide applicability and great capacity they showed in collecting data from a variety of environments. A DSN contains a set of directional sensors (such as infrared, ultrasound, and video sensors) doing the task of monitoring the targets that exist in the network. These sensors are able to control their own orientation. A directional sensor is capable of sensing several directions; however, at each unit of time, it senses only a single direction. Regarding directional characteristics of sensors, this paper centers on sensing abilities only, and communication capacities are overlooked. Using different ways, we can extend the sensing capabilities of the directional sensors [1,2]. First, a number of directional sensors of a similar type can be positioned on one sensor node in such a way that each of the sensors be responsible for monitoring a different angular sector. In [3], we can find a practical instance of this arrangement, where a single node carries four ultrasonic sensors for the purpose of getting ultrasonic signals from any direction. In the second way, each sensor node can be well matched with a mobile device in order for the nodes to move around. Third, each sensor node can be equipped with a device that is capable of switching the direction of the sensor in order to enable it to sense in all directions, albeit not simultaneously. The activated direction

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

https://doi.org/10.1016/j.pmcj.2018.10.009
1574-1192/

of a sensor at any unit of time is termed 'working direction'. Similar to a number of previously-conducted studies [4,2,5], the present research takes into consideration the third approach explained above and assumes that different possible directions in which a sensor may work do not have any overlap and a sink is positioned in the communication range of each sensor. Thus, the connectivity problem is not discussed in this paper.

One of key tasks of a DSN is providing coverage (i.e., gathering data from a given field). The type of data gathered in each network is dependent on the objectives specified for each application [6]. From a general view, the problem of coverage falls into two categories: area coverage and target coverage. The former provides coverage for an area as a whole, while the latter does it for some specific points defined within the field of interest [7]. The present paper attempts to find a solution to the target coverage problem in networks where the sensors possess more than one sensing range (i.e., multiple power levels); the aim is to prolong the network lifetime as much as possible. The network lifetime is the amount of time during which the sensors are able to continue their monitoring task. To solve this problem in DSNs, a number of challenges need to be taken into consideration. First, because of the limited sensing angle the directional sensors normally have, they are not capable of sensing entirely a circular region; this causes more complexity of the problem compared with wireless sensor networks. Second, sensors have normally non-rechargeable limited-power batteries, which makes it really difficult to work especially in harsh and/or remote environments. As a result, it is important to design novel algorithms considering the use of power saving techniques in order to more efficiently solve the target coverage problem in DSNs. In general, there are two popular techniques to save power in sensor networks: (i) scheduling the state of sensors and (ii) adjusting their sensing range [8].

The scheduling technique takes the advantage of the redundant sensor deployment occurred in such networks. It puts appropriate sensors into several cover sets each of which is capable of covering all the targets in the network. Furthermore, it determines the amount of time each cover set is able to continue its operation. Next, the constructed cover sets are activated one by one for a defined duration of time. While a cover set is operating, the sensors of the other cover sets are kept inactive. This way, the network lifetime increases significantly not only because the inactive sensors consume only a negligible amount of energy, but also since frequently oscillating of the battery between active and inactive modes causes it to last for a longer period [2]. To extend the network lifetime, adopting the adjusting technique is another proper suggestion already offered in some other studies [9–11]. Its main task is saving the energy of a sensor when it is covering the neighboring targets. The reason is that with an increase in the distance between a sensor and its corresponding target, the energy consumption of the sensor rises, too [7]. Recently, literature has witnessed a number of efficient scheduling algorithms proposed to address the target coverage problem in networks containing sensors with only one sensing range. Though, the adjusting technique, which can be normally applied to sensors with multiple sensing ranges, has not received adequate attention from researchers in this field of study. Accordingly, the present paper proposes an efficient algorithm to solve the target coverage problem and, at the same time, maximize the network lifetime through the use of both scheduling and adjusting techniques.

Here we present two examples to explain the performance of the two above-mentioned techniques and to show the reason we were motivated to use a combination of them in one algorithm. To illustrate the efficiency of the scheduling technique, we provide an example network in Fig. 1. Let us assume that Fig. 1 illustrates a DSN containing four sensors (i.e., $s_i (1 \le i \le 4)$) and three targets (i.e., $t_m (1 \le m \le 3)$). The figure also shows the sensor possesses three directions (i.e., $d_{i,j} (1 \le j \le 3)$ signifies the directions of sensor s_i). The possible cover sets we can make in this example are as follow. $C_1 = \{d_{1,3}, d_{2,2}\}$, $C_2 = \{d_{2,1}, d_{3,1}, d_{4,3}\}$ and $C_3 = \{d_{1,3}, d_{3,1}, d_{4,3}\}$. According to a classical assumption, let us assume that each sensor stays active for 1 unit of time. If one of the cover sets, for example C_1, stays activated for the whole period of the sensors' battery life, all of the targets can be under a full coverage for 1 unit of time. Thus, the network lifetime cannot be further extended since sensors that still have energy are not able to cover all the targets. Whereas if the three above-mentioned cover sets get activated one by one for 0.5 units of time, the network lifetime extends 1.5 units. As a result, it can be concluded that the scheduling technique can have a great contribution to maximization of the network lifetime.

To illuminate the efficiency of the adjusting technique, let us have another example presented in Fig. 2. Fig. 2.A and .B show the sensing ranges of each directional sensor when fixed to level 1 and 2, respectively. In this paper, $(d_{i,j}, a)$ refers to sensor direction $d_{i,j}$ when it is activated at level a. Suppose that the batteries are able to keep sensors active for 1 unit of time at power level 1 and 0.5 units at power level 2. If power level 1 is considered, there will be a single feasible cover set (i.e., $(d_{1,2}, 1), (d_{3,1}, 1), (d_{4,3}, 1)$) and the total network lifetime will be equal to 1. While, if power level 2 is considered, there will be more cover sets and a network lifetime of 1.5 unit of time can be achieved (consider, for example, $\{(d_{1,3}, 2), (d_{2,2}, 2)\}$, $\{(d_{3,3}, 2), (d_{4,1}, 2), (d_{2,1}, 2)\}$, and $\{(d_{1,3}, 2), (d_{3,3}, 2), (d_{4,1}, 2)\}$ are activated for 0.5 units of time each). Note that if the sensor directions and their sensing range are used appropriately, the network lifetime can be considerably extended.

In this paper, we addressed the problem of maximizing network lifetime in DSNs, which was called MNLAR (Maximum Network Lifetime with Adjustable Ranges) in [4], using the following scenario. Let us assume that m targets with certain locations are positioned in the network; then n directional sensors with multiple sensing ranges are also distributed randomly close to the targets in order to cover them. At the beginning of the algorithm operation, all sensors have a defined battery lifetime, and each of them can be in either active or inactive mode. At each unit of time, each active sensor is able to cover the targets positioned in the field of view of one of its non-overlapping directions. When a sensor is activated, its energy consumption depends on its sensing range applied to the cover set. Note that the inactive sensors cover no target, hence consuming only a negligible amount of energy. The MNLAR problem is aimed to schedule the sensors' activity so that the network lifetime can be maximized. In this process, it is important to note that during lifetime of the network, each target needs to be covered by the working direction and sensing range of at least one activated sensor.

It is worth noting that the MNLAR problem addressed in this paper has been already proved as an NP-complete problem [4]. Thus, finding near optimal solutions with lower computational complexity is of a high significance. Furthermore,

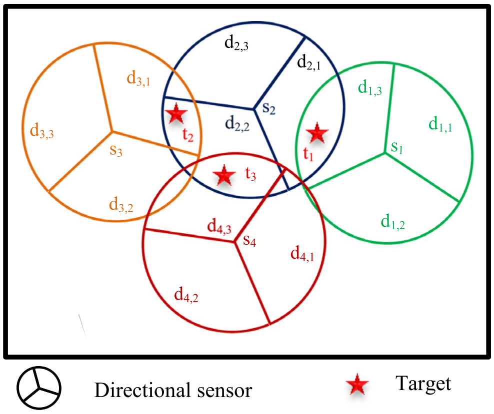

Fig. 1. Example network with four directional sensors and three targets.

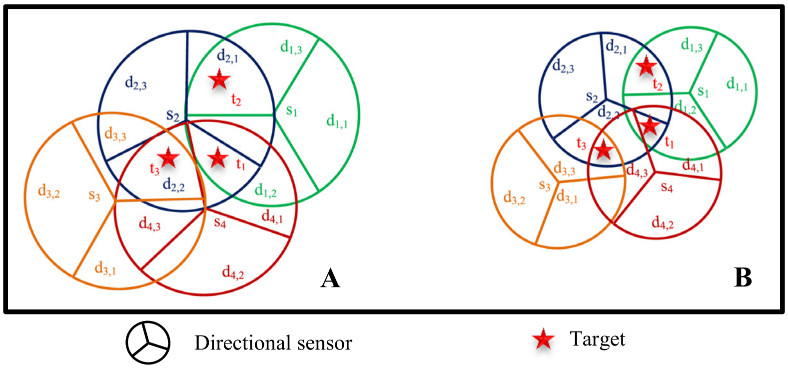

Fig. 2. Example network with four directional sensors, three targets and two power levels.

there is a need for an optimization technique that uses a reasonable space of memory and computational resources in order to obtain desirable results. Therefore, a metaheuristic approach can work well in achieving an approximate solution of the MNLAR problem.

1.1. Our contribution and organization of the paper

In this study, we propose a GA-based algorithm that combines two power saving techniques (i.e., scheduling and adjusting) in order to not only solve the target coverage problem, but also maximize the network lifetime. In the following, the main contributions of this paper are presented.

- We present a GA-based approach to solve the MNLAR problem, which provides a full coverage on a set of targets and maximizes the network lifetime through selecting the sensors with the most appropriate working directions and sensing ranges.
- We design a target-oriented chromosome representation in a way to efficiently encode the problem in hand.
- We make use of an effective fitness function aiming to find the solutions with least energy consumption.
- We propose a repair operator capable of changing invalid chromosomes to valid ones.
- We run several simulations on the proposed algorithm and compare the obtained results with those of an existing algorithm.

This paper is organized as follows: Section 2 briefly reviews the studies related to solving target coverage problem. Section 3 introduces MNLAR problem in DSNs. In Section 4, we propose a GA-based algorithm for solving the problem. In Section 5, the performance of the proposed algorithm is evaluated through several experiments. Finally, Section 6 contains the conclusion.

2. Related work

In sensor networks, a key issue is coverage that refers to gathering various types of data from a given environment. The present study only centers on the target coverage problem. Reviewing the literature, we can find several studies that attempt to solve the target coverage problem in WSNs based on both power saving techniques (see [12,13,9,10]). However, due to the limited sensing angle of directional sensors, the algorithms proposed for WSNs cannot be applied to DSNs. Here is a review of some brilliant papers that have applied the scheduling technique to the solution of the target coverage problem in DSNs.

Ai and Abouzeid [14] were one of the pioneers in investigating the target coverage problem in DSNs. They were centered on providing maximum coverage using minimum active sensors to monitor maximum number of targets. Several algorithms were offered to solve the problem. In [1], the authors defined the multiple directional cover set problem and then proved its NP-completeness. Several heuristics were proposed to create non-disjoint cover sets each of which capable of covering all the targets existing in the network. The researchers in [15] introduced a greedy-based algorithm and a GA-based algorithm to solve the target coverage problem. In [16–19], several algorithms based on learning automata were also proposed to solve this problem. The learning automata was also used in [11,20] to solve the priority-based target coverage problem where each target had a different coverage quality requirement and needed to be monitored by more than one sensor. In [21], the target coverage-aware clustering method was proposed for DSNs. Their proposed algorithm confirmed the importance of considering issues such as exploited number of targets and their neighbors, the sensors' distance from the sink, and their residual energy in selecting cluster heads and the gateways. They succeeded to decrease the number of activated sensors, hence increasing the network lifetime significantly. In [22], the joint problem of the maximizing both sensing coverage quality and network lifetime was addressed in order to find an efficient way to monitor heterogeneous targets in Smart City applications. They found the optimal coverage algorithms inapplicable to large Smart City application networks that have too many sensor nodes and targets. In such conditions, greedy algorithms with probabilistic sensing quality measurements have shown near-optimal performance.

To solve the network lifetime maximization problem, the authors in [2] made use of a column generation scheme to propose a model in which a two-level strategy (involving a GA and an integer linear programming (ILP) approach) was used to find a solution to the problem. The ILP approach plays only two roles: either escaping from the local optima or providing optimality for current solution. In [5], two versions of lifetime maximization problem were addressed. First, it was assumed that predefined sensing directions were given, while sensing directions could be devised freely in the second version of the problem. In this version, a polynomial-time algorithm was designed to shape sensing directions capable of maximizing the network lifetime. To solve both versions of the problem, they proposed a column generation algorithm. The obtained results indicated that addressing the second version significantly improved the network lifetime, while computational effort was found comparable for both problem versions. In [23], the maximization of network lifetime was also addressed specifically in camera sensor networks under the partial and full target coverage model. In the model of full target coverage, all the targets in the network were assumed to be monitored throughout the network lifetime, while in the partial model, the targets were assumed with weights based on their value. In this sense, only those targets that held a sum of weights higher than a predefined threshold were assumed coverable during the entire lifetime. To solve this problem, the authors proposed three heuristics whose performance in terms of extending network lifetime were confirmed.

In [24], the authors categorized the heuristics formerly proposed in literature to solve the target coverage problem into two classes: sensor-oriented and target-oriented. They then proposed several target-oriented heuristics in order to solve the same problem specifically in visual sensor networks. They confirmed the target-oriented heuristics outperformed the sensor-oriented ones in terms of solving the mentioned problem. In [25], 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 singly covered. To solve this problem, they proposed several algorithms. All papers mentioned above were centered on solving the target coverage problem where one sensor possessed only one power level and they did not pay attention to adjustability of sensing ranges of sensors, which could significantly save their energy when each sensor was responsible for covering only its adjacent targets. As a result, to have an efficient solution to the target coverage problem, new algorithms are needed to be designed capable of taking the advantages of both power saving techniques so that the network lifetime can be well maximized.

In [4], the researchers proposed two greedy-based algorithms to solve the target coverage problem in DSNs where the sensors held multiple sensing ranges. They applied both scheduling and adjusting techniques to the maximization of network lifetime. The process of cover set construction in these greedy-based algorithms were as follows. In each stage of the algorithm, from among the available minimal adjusted sensors, the one that had more contribution to the cover set formation process was selected. Then, the list of uncovered targets was updated; additionally, the list of available minimal adjusted sensors was updated by eliminating the sensors that already participated in the cover set formation process. This process continued until all targets were provided with a full coverage. The authors tested their algorithms performance through conducting some experiments and the obtained results showed that a combination of the scheduling and adjusting techniques could significantly extend the network lifetime. The proposed heuristics showed their high capacity in solving the target coverage problem; although, their performance was too much dependent on closeness of the initial candidates to optimal solution; this could led the scheme to local minimum due to its heuristic search. In the present paper, we propose a target-oriented GA-based algorithm to find an effective solution to this problem.

3. Problem definition

The scenario taken into account in this study involves several targets distributed at known locations within a given area. In addition, a number of directional sensors with adjustable sensing ranges were placed in the vicinity of the targets in

Table 1
Notations.

Notation	Meaning
n	Number of sensors
m	Number of targets
w	Number of directions per sensor, $w \geq 1$
a	Number of alternative power levels, $a \geq 1$
s_i	A sensor, for all $i \in \{1, \ldots n\}$
t_k	A target, for all $k \in \{1, \ldots m\}$
l_i	Lifetime of sensor s_i
$d_{i,j}$	Jth direction of ith sensor
D	Set of the directions of all the sensors
S	Set of sensors, $\{s_1, \ldots s_n\}$
T	Set of targets, $\{t_1, \ldots t_m\}$
$(d_{i,j},\ a)$	Refers to sensor direction $d_{i,j}$ activated at level a.
$T_{(d_{i,j},\ a)}$	Refers to all the targets covered by sensor direction $d_{i,j}$ when it is set at level a.

the network to provide a ceaseless coverage on them. In other words, directional sensors hold multiple predefined sensing ranges; all targets positioned in the scope of these ranges are covered with a predefined power level. Furthermore, each directional sensor possesses a number of directions; however, at each unit of time, only one of them works (recognized as working direction). A directional sensor monitors all targets positioned not only within its operating range, but also in the field of view of its working direction. Sensors in such networks have a limited energy resource and they can be activated at a limited number of sensing ranges (power levels). The amount of energy consumed by each sensor depends on which one of its sensing ranges is selected for the cover set formation process; the greater sensing range, the more energy is consumed [9]. All the sensors in a network are alike in their initial battery power and the energy consumed for each one of their sensing ranges. Since the power levels gradually extend the sensing ranges of the devices, for each sensor direction $d_{i,j}$ and each level $a > 1$, we have $T_{(d_{i,j},\ b)} \subseteq T_{(d_{i,j},\ a)}\ \forall\, b \in \{1,\ \ldots,\ a-1\}$. Moreover, the adjusted sensor direction $(d_{i,j},\ a)$ is defined minimal for target t_j if $t_j \in T_{(d_{i,j},\ a)}$ and either $a = 1$ or $t_j \notin T_{(d_{i,j},\ b)}\ \forall\, b \in \{1,\ \ldots,\ a-1\}$. Table 1 presents the notations used in this paper (they are similar to the notations used in the original paper [4] that addressed the MNLAR problem for the first time).

Note that to switch between directions of one sensor, a negligible amount of energy is consumed, which is ignored in the present paper as it is ignored in work carried out in (see [4]). This paper uses a positive parameter Δ^a for each power level a (sensing range) in order to model various battery consumptions [9]. Here Δ^a stands for the ratio between battery consumption at level a and level 1 that is the weakest level, hence the cheapest one. For instance, if $\Delta^a = 2$, then level a consumes two times more energy than level 1. Obviously, $\Delta^1 = 1$. In addition, the total battery power is normalized on the energy consumption of level 1; in other words, a sensor battery keeps the battery active for 1 unit of time if it is set to level 1.

Problem. How to divide the available sensors into possible cover sets so that each cover set can provide coverage for all targets in the network and, simultaneously, prolong the network lifetime as much as possible. In other words, the main challenge of this process is the way we can assign appropriate sensors with one of their directions and sensing ranges to each cover set in a way to not only provide a desirable coverage, but also maximize the network lifetime.

Here is presented an example network (see Fig. 3.A) consisting of four targets, and five directional sensors to monitor the targets existing in the network. Each sensor holds three directions and two sensing ranges (see Fig. 3.B). The minimal adjusted sensors covering each target are $t_1 = \{(d_{3,1},\ 2), (d_{1,2},\ 2)\}$, $t_2 = \{(d_{5,2},\ 1), (d_{4,1},\ 2)\}$, $t_3 = \{(d_{5,3},\ 2), (d_{1,1},\ 2), (d_{4,3},\ 2)\}$, $t_4 = \{(d_{2,1},\ 1), (d_{4,2},\ 2)\}$. The problem is how to form the best cover set. Let us assume that $C_1 = \{(d_{3,1},\ 2), (d_{5,2},\ 1), (d_{1,1},\ 2), (d_{2,1},\ 1)\}$ and $C_2 = \{(d_{1,2},\ 2), (d_{4,1},\ 2), (d_{1,1},\ 2), (d_{4,2},\ 2)\}$ are two cover sets that can potentially be constructed to run in this network. As mentioned earlier, the cover set with less energy consumption is more desirable; thus, to this example network, cover set C_1 can be applied.

4. Proposed GA-based algorithm

This section proposes a target-oriented GA-based algorithm as a solution to the MNLAR problem in DSNs. The network operation of this algorithm (whose pseudo code is presented in Algorithm 1) is done in a number of rounds; the output of each round is a cover set that is able to monitor all targets existing within the network. Each round comprises two phases: initialization and cover set formation. In the former, we configure the network in a way to construct a cover set. In the latter, using GA, we select an appropriate cover set from among the possible ones. In the following, both phases are explained in detail.

4.1. initialization

In this phase, the input parameters are determined. Here, we first set the residual lifetime of each sensor s_i. Set *Covers* comprises the cover sets together with their activation time. Parameter *Lf* displays the maximum network lifetime. The set $Covers$ and the Lf value are returned as the solution and the maximum network lifetime, respectively. After initializing the essential parameters, we check whether a cover set can be constructed or not. If yes, the cover set construction phase gets started. For more detailed information about the construction process of a cover set, see Section 4.2. After the cover set formation phase is done, a cover set containing sensors with appropriate directions and sensing ranges is returned as output. Next, the maximum activation time the constructed cover set can provide is calculated by the algorithm; this value is then added to the total network lifetime. After that, the residual energy of the sensors existing within the formed cover set is computed concerning which one of the sensing ranges of the sensors are applied to the process of cover set formation. The algorithm then eliminates the sensors that have no residual energy from the list of available sensors. To calculate the activation time, we consider the adjusted sensor of C_{cur} that minimizes $\frac{L_i}{\Delta^a}$; let us call it $(s_h,\ b)$. We set $Wt = \frac{L_{sh}}{\Delta^b}$; this ensures a feasible activation time for each $(s_i,\ a) \in C_{cur}$. The cover set construction phase continues its operation until all the targets are under a desirable coverage. At the final step, the constructed cover sets together with their respective activation times are returned as the output of the algorithm.

4.2. Cover set formation

In this phase, we propose a GA-based algorithm capable of forming cover sets consisting of sensors with appropriate sensing ranges. To do this efficiently, our algorithm starts its operation with randomly creating an initial population. Then, the reproduction operation, which involves selection, crossover, and mutation, is done on the initial population and at the end of the operation, a set of appropriate chromosomes is selected as the coming generation. This process is repeated until we can find an appropriate cover set. In the following, the algorithm is explained in detail.

Algorithm 1 : Proposed Algorithm

```
Input: Directional network Net = (S, T), number of power levels p
Output: A cover set with appropriate minimal adjusted sensor directions
Set initial lifetime of each sensor s_i
Covers ← Ø
Lf ← 0
while ⋃_{s_i∈S} T(s_i, p) ≡ T do
   Initialize population Pop;
   Evaluate Pop;
   While (not terminated)
     P_s = Select (Pop);
     P_c = Crossover (P_s);
     P_c = Repair (P_c);
     P_m = Mutate (P_c);
     Evaluate P';
     P = Survival (P, P');
   End while
Wt =max feasible activation time for C_cur
Lf ← Lf + Wt
Update the residual energy of sensors in the cover set
   Covers ← Covers ⋃{(C_cur, Wt)}
end while
return (Covers, Lf)
```

Representation: In the present study, to model a chromosome as a solution of the problem space, we make use of an integer-based representation. Each chromosome in this model stands for a cover set and each gene of the chromosome shows which sensor with which direction and sensing range is selected to participate in the formation of a cover set. Each chromosome is encoded by a vector of the length m; the parameter m signifies the number of targets within the network. The value allocated to a gene is selected from among the minimal adjusted sensor directions that monitor the target corresponding to that gene. To form a chromosome (through allocating a value to its gene), the algorithm first marks out the most critical target and allocates a value to its corresponding gene; that is, the algorithm chooses a minimal adjusted sensor direction from among the ones monitoring the target. When allocating values to genes of a chromosome, two issues are of importance: first, at each unit of time, only one direction of a single sensor participates in the cover set formation process, and second, only one sensing range of a single sensor can be applied to the cover set being constructed. As a result, after selecting a minimal adjusted sensor direction to assign a value to a gene corresponding to a target, the other directions

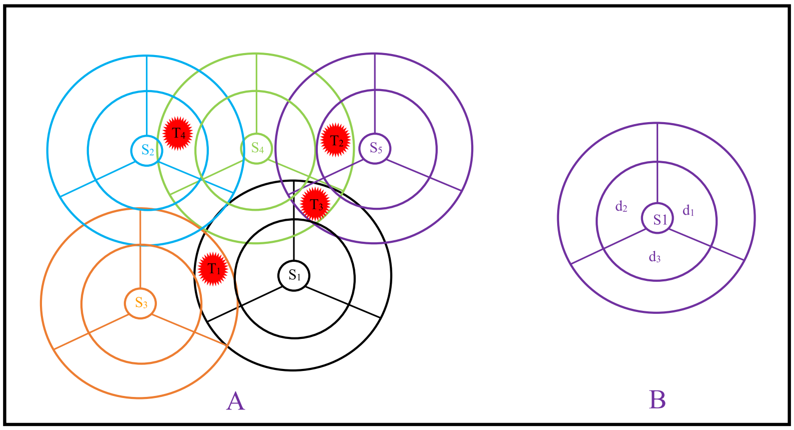

Fig. 3. (A) Example network with five adjustable directional sensors and four targets. (B) An example directional sensor with three directions and two sensing ranges.

Target Number	t_1	t_2	t_3	t_4
Gene Value	$(S_{3,1}, 2)$	$(S_{5,2},1)$	$(S_{1,1},2)$	$(S_{2,1},1)$

Fig. 4. An example chromosome for the network.

and other sensing ranges of the same sensor cannot be used in the under-construction cover set. If a chromosome is modeled this way, it will be valid. By definition, a valid chromosome is one capable of providing coverage for all targets in the network. The process of finding a critical target and allocating a value to its corresponding genes goes on until the genes of all targets hold appropriate values. Such strategy is adopted since some targets in real-life applications might be located in areas where they can be covered by only one or a few minimal adjusted sensor directions. To maximize the coverage, these targets (known as critical targets) must be covered first. Therefore, the algorithm needs to determine the priority of the targets according to their vulnerability.

Bear in mind that there are fewer targets than sensors in the random deployment environments. In such an environment, our model can shorten the chromosome length. Here is presented an example to elaborate the model proposed in this paper (see the network depicted in Fig. 3). As the number of targets is fixed at 4, the length of each chromosome is set to 4, too. In other words, each gene shows the status of one target. To create a chromosome, first the most critical target is identified. Let us assume that target t_1 is the first critical target since the total energy of the minimal adjusted sensor directions covering it is less than that of other targets. Therefore, a value is allocated to its gene. That is, the algorithm chooses a minimal adjusted sensor from among the ones covering the same target. The selected minimal adjusted sensor is then temporarily eliminated from the list of the minimal adjusted sensors that cover the critical target. Note that once a minimal adjusted sensor is selected to cover the critical target, if the other directions and sensing ranges of the same sensor are presented in the list of the minimal adjusted sensors covering the other targets, they will be removed. This way, we can avoid the selection of more than one direction and one sensing range of a single sensor. The process of identifying a critical target and allocating a value to its gene continues until the chromosome in hand is completely produced. Fig. 4 shows an example chromosome.

Initial population: To have the initial population, we randomly produce a set of chromosomes each of which is represented by a vector. Remember that while creating each chromosome, we take into consideration the issues mentioned above.

Fitness: Generally, in evolutionary algorithms, fitness function is applied extensively because of its explicit and implicit impacts on direction of the search in solution space. If this parameter is properly set for a problem, there will be available sufficient information about the search direction, and also proper candidate solutions can be distinguished from improper solutions. The present study considers the fitness function as the amount of energy the sensors in the cover set consume. This helped us identify the cover set consuming less amount of energy. Note that the energy consumption is dependent on which sensing range is selected from among other sensing ranges of the sensors participating in the cover set. Smaller sensing ranges are more favored for constructing a cover set because they consume less energy in doing the coverage task. Therefore, the chromosomes (solutions) with less amount of energy consumption are more desirable; that is, this paper addresses a minimization problem. For this purpose, the energy consumption of the cover set needs to be computed considering which sensors with which one of their sensing ranges are selected for the formation of the cover set in hand. For instance, the fitness function of the chromosome depicted in Fig. 4 is 3.

Selection: During the selection operation, valid chromosomes are selected by means of some methods, including roulette-wheel selection, rank selection, and tournament selection. The chromosomes that hold better fitness function are more likely

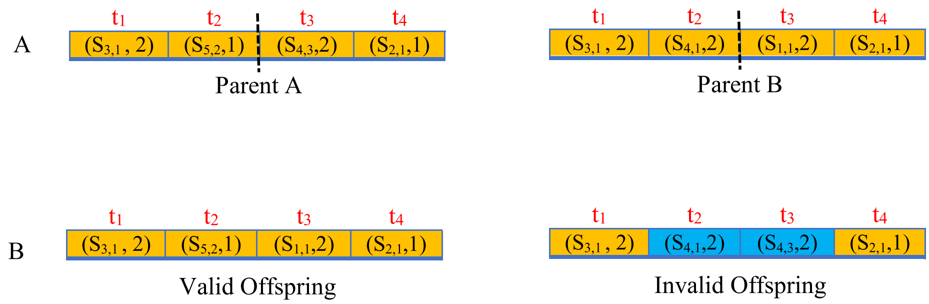

Fig. 5. An example of single-point crossover operation.

to be chosen. The scheme proposed here applies the method of roulette-wheel selection. The selected chromosomes make use of the crossover operation to generate new child chromosomes.

Crossover and mutation: The crossover is an operation that results in production of at least one new offspring solution. In this stage, first, two parents need to be chosen. Then, the newly produced offspring solution will inherit a number of genes from its first parents and the rest of genes are inherited from its second parents. Literature contains studies introducing various types of crossover such as single-point crossover, two-point crossover, uniform crossover, and so on. Here, we use a single-point crossover operator to produce a pair of child chromosomes from two chromosomes selected as parents. To do this, the crossover point is chosen in a random way, the parent chromosomes then exchange the information they hold. Although the sensors in our problem hold multiple directions and sensing ranges, at each unit of time, only one direction and one sensing range from each sensor can be applied to the construction of the cover set in hand. As a result, after crossover operation, it is possible to have invalid chromosomes. A chromosome is invalid if at the time of allocating value to its genes, more than one sensing range or more than one direction of a single sensor are used.

To have a deeper understanding of the crossover operation, have a look at Fig. 5. Fig. 5A shows two chromosomes selected as parents. Additionally, we marked the crossover point. Fig. 5B presents the offspring resulted from the crossover operation. As explained earlier, it was probable that the resulted offspring was either valid or invalid. As shown in the figure, one of the offspring is valid and the other one is invalid. The reason of the latter's invalidity is the selection of two directions simultaneously from a single sensor in construction of a cover set. To fix the invalid chromosomes that might be produced during the crossover and mutation operations, we propose a repair operator. When a valid chromosome is generated, the mutation operation gets started through which a gene of the chromosome is randomly selected and its value is changed. That is, the value of a gene corresponding to a given target is changed to a new value selected from the other minimal adjusted sensor directions monitoring the target. Remember that after selecting a new value for the mutation operation, it is important to check validity of the chromosome. For this purpose, when a minimal adjusted sensor direction is selected to allocate value of a gene, the selected value needs to be compared to those of the other genes of the chromosome. If the selected value does not cause invalidity of the chromosome, it is applicable; otherwise, the algorithm has to select another value.

Repair operator: This paper proposes a repair operator assuring the feasibility of the child chromosome produced through the crossover operation. This operator takes into account the value of the gene that corresponds to each target in the network. In each value, three different parameters are determined: the sensor number, the direction number, and the sensing range level of that sensor. After that, it is important to check whether the other directions and/or other sensing ranges of that sensor are applied to the allocation of value to genes corresponding to other targets. The same procedure is performed for all genes. In case any invalid value is found, the value of the gene needs to be fixed. Note that if invalid values are found for more than one gene, then we need to mark out the most critical gene and make a change to its value. This process goes on until all genes get valid.

Stopping criterion: The termination criterion of each round of the algorithm is to reach predefined maximum number of iterations. In this condition, an appropriate solution (a cover set) is returned as the algorithm output.

4.3. Time complexity

In this section, we measure the complexity of the proposed algorithm. In the GA-based algorithm proposed in this paper, the basic operations are creation of initial population, evaluation of the fitness, reproduction (involving selection, crossover, repair, and mutation operation). In the present study, a chromosome is encoded using an integer vector of the length m in a random way. As a result, it takes O(m) time to encode each of the chromosomes and O($N_{pop} \times m$) time to form the initial population of size N_{pop}. We can compute the fitness value of a chromosome in O(m) time. This paper uses the Roulette-wheel selection process that holds time complexity O(N_{pop}). A single-point crossover operation is applied, which takes O(m) time to produce two offspring. The mutation operation involves the random selection of a gene corresponding to a given target, then allocation of a new value to the gene. In this condition, the mutation operation can be done in a constant time, i.e., in O(1) time.

However, in our study, when a new value is to be assigned to a gene, it needs to be checked whether or not it causes invalidity of the offspring; therefore, the mutation operation is performed in O(m) time. Another operation we use in the proposed algorithm is the repair operator whose responsibility is to check the validity of offspring in each iteration of the algorithm. This operation is done in O(m^2) time. Remember that the crossover, repair, and mutation operations are iteratively performed until the termination criterion is met, and following each iteration, the fitness value of the newly-produced chromosome is computed in order to determine one of the parent or child chromosomes will survive in the upcoming generation.

5. Simulation results

In this section, we present the results of the experiments carried out to evaluate the performance of the proposed algorithm. To simulate the algorithm, we used MATLABR2012b on a system with an Intel i7 processor, 3.4 GHz CPU, 4 GB RAM and Microsoft Windows7 as a platform. The criterion used to evaluate the algorithm's performance was the network lifetime. Each experiment was customized in a way to test the impact of a different parameter on the network lifetime. In this study, to configure a DSN, m targets were scattered within an area of size $500(m) * 500(m)$; next, n sensors with identical directions and sensing ranges were deployed close to the targets in order to cover them. Each sensor applied to these experiments had multiple directions and sensing ranges; however, at each given time, only one direction and one sensing range of each activated sensor could be used for doing the coverage operation. The by-default parameters of the experiments were set as follow. Each sensor held one unit of energy. The number of sensors and targets were set to 100 and 10, respectively. In addition, for each sensor, its radius was fixed at $100(m)$, the number of its directions was set to 3, and the number of sensing range was 4.

To find the relationship between the number of sensing ranges and the network lifetime, the number of sensing ranges of each directional sensor was set to a number between 1 and 4 with incremental step 1, and the number of sensors, targets, and sensing radius were kept as fixed by default. As the results demonstrated, there was a direct relationship between the two parameters; on the other words, an increase in the number of sensing ranges caused the network lifetime to further extend, but it was true only to a specific extent. When the number of sensing range was set to 1, the algorithm showed its highest energy consumption rate. It was due to the fact that with more sensing ranges, the algorithm attempted to choose those sensors that had smaller sensing ranges. This diminished the energy consumption of the cover set. In the following experiments, the number of sensing range was set to 4. To have more reliable results, we run each simulation scenario for 10 times, and for each scenario, the average network lifetime was computed. Note that in each experiment, we test the impact of a parameter through increasing its value. In this situation, the larger set of that parameter was attempted to include all features of the previous scenario; that is, the previous environment were retained and only extra features were added to that. The reason was to have a consistency in our evaluation. The results obtained from the proposed algorithm were compared to those of a greedy-based algorithm [4] introduced recently to literature. In our proposed algorithm, the probability of crossover and mutation was set to 0.2 and 0.5, respectively, and the population size was fixed at 60.

Experiment 1. This experiment was carried out to test the relationship between the number of sensors and the network lifetime. To this end, the number of sensors was fixed at 60–100 with incremental step 10. The results depicted in Fig. 6 showed that with an increase in the number of sensors, the network lifetime increased, too. The reason was that when more sensors are available in the network, each target can be covered by more sensors. Therefore, the algorithm was able to construct more cover sets each of which had a more extended activation time. As confirmed by the comparative results, the proposed GA-based algorithm outperformed the greedy-based one concerning the maximization of the network lifetime. This superiority was due to managing the critical targets when the cover sets were being formed. The obtained results showed that GA constructed the fewest cover sets. It should be noted that it was not considered as a problem because the aim of the proposed algorithm was only maximization of the network lifetime. Thus, in the process of cover set construction, the main challenge of this algorithm was selecting the sensor directions that had minimum sensing range. This led to the formation of cover sets with more extended activation time.

Experiment 2. This experiment was aimed to investigate how the number of targets could affect the network lifetime. To do this, the number of targets was set to 6–14 with incremental step 2. As the obtained results in Fig. 7 show, increasing the number of targets caused a decrease in the network lifetime. The reason was that when more targets existed, more sensors were required to provide their coverage. This caused the sensors to run out of their energy sooner.

Fig. 8 shows the direct relationship between the number of targets and the energy consumption of the cover sets in the network. In other words, the more targets we had in the network, the more energy the cover sets consumed. The reason was that naturally more sensors were needed to monitor more targets. The results confirmed that GA consumed more amount of energy compared to the other algorithm since GA employed more sensors of the network and, on the other hand, it constructs more cover sets.

Experiment 3. This experiment was conducted to test the effect of the sensing range on the network lifetime. For this purpose, the sensing range was fixed at 80-120(m) with incremental step 10(m). As shown by the results presented in Fig. 9, an increase in the sensing range led to a rise in the network lifetime. It was because of the fact that in this condition, the sensors had chance to cover more targets, hence requiring fewer sensors to provide all targets in the network with desirable coverage. When the results of the proposed GA-based algorithm were compared with those of the greedy-based one, it was confirmed that the former was more successful than the latter in terms of maximizing the network lifetime.

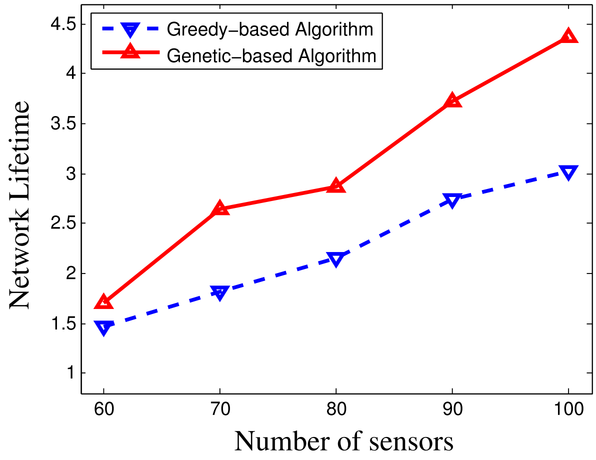

Fig. 6. Effect of the number of sensors on the network lifetime.

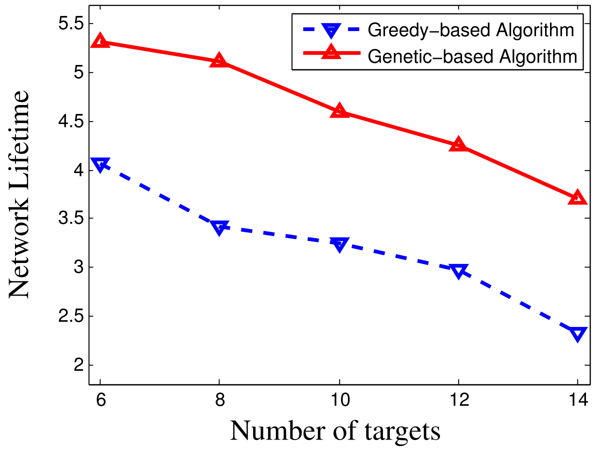

Fig. 7. Effect of the number of targets on the network lifetime.

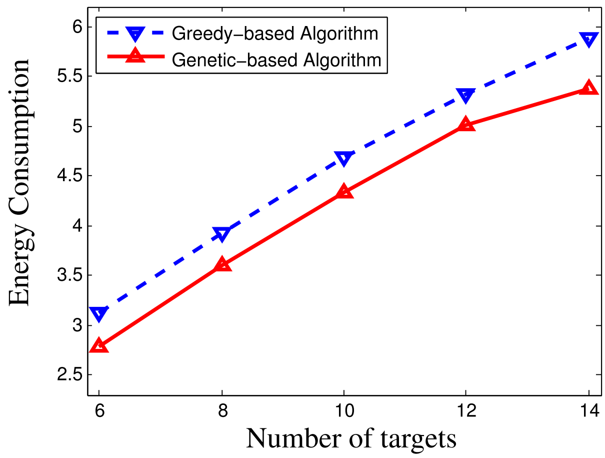

Fig. 8. Effect of the number of targets on the energy consumption.

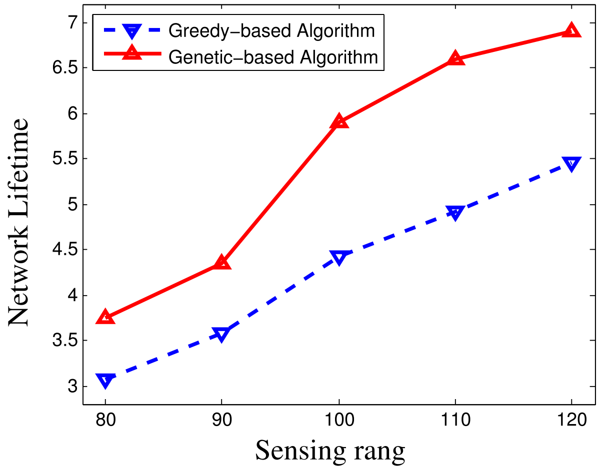

Fig. 9. Effect of sensing range on the network lifetime.

6. Conclusion

In this study, we addressed the target coverage problem in networks containing sensors with multiple directions and sensing ranges, aiming to maximize the network lifetime. To solve this problem, we proposed a target-oriented GA-based algorithm combining two power saving techniques, i.e., scheduling and adjusting. The main aim was to select the most appropriate directions and sensing ranges to form cover sets capable of monitoring all the targets in the network and, simultaneously, to maximize the network lifetime. In this study, we had some innovations, including the application of a suitable chromosome representation to modeling the problem in hand, management of the critical targets in the chromosome production process, proposing a new fitness function, and introducing a repair operator to fix the validity of offspring. In addition, to test how efficiently the algorithm works, a number of experiments were carried out and the obtained results were compared to those of a greedy-based algorithm proposed recently in literature. The comparative results showed that the proposed algorithm outperformed the greedy-based one concerning the construction of cover sets that consumed the least amount of energy, which resulted a significant extension of the network lifetime.

References

[1] 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.
[2] A. Singh, A. Rossi, A genetic algorithm based exact approach for lifetime maximization of directional sensor networks, Ad Hoc Networks 11 (2013) 1006–1021.
[3] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, D. Culler, An analysis of a large scale habitat monitoring application, in: Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 214–226..
[4] H. Mohamadi, S. Salleh, M.N. Razali, Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges, J. Netw. Comput. Appl. 46 (2014) 26–35.
[5] A. Rossi, A. Singh, M. Sevaux, Lifetime maximization in wireless directional sensor network, European J. Oper. Res. 231 (2013) 229–241.
[6] M. Amac Guvensan, A. Gokhan Yavuz, On coverage issues in directional sensor networks: A survey, Ad Hoc Networks 9 (2011) 1238–1255.
[7] 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.
[8] M. Cardei, J. Wu, M. Lu, Improving network lifetime using sensors with adjustable sensing ranges, Int. J. Sens. Netw. 1 (2006) 41–49.
[9] R. Cerulli, R. De Donato, A. Raiconi, Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges, European J. Oper. Res. 220 (2012) 58–66.
[10] 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.
[11] 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.
[12] H. Mohamadi, A.S. Ismail, S. Salleh, A. Nodhei, Learning automata-based algorithms for finding cover sets in wireless sensor networks, J. Supercomput. 66 (2013) 1533–1552.
[13] H. Mohamadi, A.S. 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] J. Ai, A.A. Abouzeid, Coverage by directional sensors in randomly deployed wireless sensor networks, J. Combin. Optim. 11 (2006) 21–41.
[15] J.-M. Gil, Y.-H. Han, A target coverage scheduling scheme based on genetic algorithms in directional sensor networks, Sensors 11 (2011) 1888.
[16] H. Mohamadi, A.S.B.H. Ismail, S. Salleh, A learning automata-based algorithm for solving coverage problem in directional sensor networks, Computing 95 (2013) 1–24.
[17] H. Mohamadi, A.S. 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.
[18] H. Mohamadi, A.S. 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.

[19] H. Mohamadi, S. Salleh, A.S. Ismail, S. Marouf, Scheduling algorithms for extending directional sensor network lifetime, Wirel. Netw. 21 (2015) 611–623.
[20] 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.
[21] S. Sharmin, F.N. Nur, M.A. Razzaque, M.M. Rahman, Target coverage-aware clustering for directional sensor networks, in: Proceedings of International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), 2017, pp. 116–120.
[22] S. Sharmin, F.N. Nur, M.A. Razzaque, M.M. Rahman, A. Almogren, M.M. Hassan, Tradeoff between sensing quality and network lifetime for heterogeneous target coverage using directional sensor nodes, IEEE Access 5 (2017) 15490–15504.
[23] A. Singh, A. Rossi, M. Sevaux, Heuristics for lifetime maximization in camera sensor networks, Inform. Sci. 385–386 (2017) 475–491.
[24] 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.
[25] S.M.B. Malek, M.M. Sadik, A. Rahman, On balanced k-coverage in visual sensor networks, J. Netw. Comput. Appl. 72 (2016) 72–86.