

A new target-oriented imperialist competitive-based algorithm to solve target coverage problem in directional sensor networks

Mahshad Saadati (✉ saadatimahshad@gmail.com)

Islamic Azad University

Mojtaba Mousavi fard

Islamic Azad University

Hosein Mohamadi

Islamic Azad University

Research Article

Keywords: Directional sensor networks, Cover set formation, Scheduling algorithms, Imperialist competitive algorithm

Posted Date: February 23rd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-862054/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A new target-oriented imperialist competitive-based algorithm to solve target coverage problem in directional sensor networks

Mahshad Saadati · Mojtaba Mousavi fard · Hosein Mohamadi

Received: date / Accepted: date

Abstract Today, directional sensor networks (DSNs) have received a great deal of attention. A DSN is composed of several self-configurable directional sensors with adjustable spherical sectors of limited angles, which can provide coverage on several targets distributed randomly within a defined area. One of the most significant problems associated with this type of networks, which has been already proved as an NP-hard problem, is monitoring the maximum number of targets by means of minimum number of sensors. Another aspect of the problem is how to extend the lifetime of such networks concerning the limited power source of the sensors. An appropriate solution to this problem is the use of scheduling technique in environments with densely-distributed sensors. In this paper, we proposed an imperialist competitive-based scheduling algorithm capable of providing near-optimal coverage through determining the priority of the targets that can be monitored by fewer sensors. To evaluate the proposed algorithm performance, the obtained results were compared to those of a target-oriented greedy-based algorithm already proposed in the literature. The final results showed the acceptable competency of the proposed algorithm in terms of solving the problem in hand.

Keywords Directional sensor networks, Cover set formation, Scheduling algorithms, Imperialist competitive algorithm

Mahshad Saadati
Department of Computer Engineering, Aliabad Katoul Branch, Islamic Azad University,
Aliabad Katoul, Iran

Mojtaba Mousavi fard
Department of Computer Engineering, Aliabad Katoul Branch, Islamic Azad University,
Aliabad Katoul, Iran

Hosein Mohamadi
Department of Computer Engineering, Azadshar Branch, Islamic Azad University, Azadshar,
Iran
E-mail:

1 Introduction

In recent years, sensor networks have emerged as proficient platforms for different applications, including battlefield surveillance, environment monitoring, health care, etc. In every sensor network, there are a number of tiny sensor nodes comprising certain components for sensing, processing, and communicating the data collected from the given environment [11]. In the context of sensor networks research, it is generally assumed that sensors are omnidirectional, containing an omni-angle of sensing range. Though, directional sensors might have a limited angle of sensing range because of technical constraints and cost considerations. Three widely-used examples of directional sensors are video sensors, ultrasonic sensors, and infrared sensors [2]. The sensing capabilities of directional sensors can be enhanced using several techniques. One of them is putting a number of directional sensors of a similar type on one sensor node, each of which is responsible for monitoring a particular area. Another technique is providing each sensor node with a mobile component enabling it to move around. The third one is to equip the sensor node with a device enabling it to switch to several directions [4]. For the purpose of this study, the third technique is adopted in order to enable the sensors to switch to different directions. The sensor nodes used in this paper carry only one sensor.

One of the key issues in DSNs is coverage that refers to gathering data from a defined area. The coverage problem generally falls into two categories: area coverage and target coverage. The aim of target coverage is determining a set of sensors to provide coverage for only some certain points within a defined environment. On the other hand, area coverage is aimed to monitor a part of the whole environment without concentration on any particular point [2]. Another vital issue in DSNs is how to manage the energy consumption of the sensors. This issue is of high significance since directional sensors work with batteries of limited energy, which are non-chargeable and irreplaceable. This is due to inaccessibility of the environments wherein they are distributed [4]. In this study, we assume that a sensor dies when it runs out of energy. An efficient way of conserving energy is the use of scheduling technique through which we can put some sensors in active mode and switch the others off, while keeping all targets in the network covered [1, 13].

In the present paper, the following scenario is taken into account. Within a two-dimensional Euclidean field, several targets are distributed in known locations. In addition, a number of directional sensors are scattered randomly in the vicinity of the targets. It is assumed that in directional sensors, the sensing region of each direction is only a sector of the sensing disk that is located at the center of a sensor with a sensing radius. For each sensor, there is a uniform sensing region, and note that the sensing regions of different directions of one sensor are not overlapped. After the random distribution of the sensors, each of them is adjusted with one of its own directions. These sensors create a network of directional sensors in a way to collect data and transfer it to a defined sink that is essentially a central base station provided to process the collected data. When a directional sensor is adjusted with a certain direction, we can say the sensor is working in this direction that is actually recognized as working direction. Once this sensor is working in a given direction and there is a target within the sensing region of the sensor, it is assumed that the target is covered by the direction of the same sensor. As the field of view of a directional sensor is normally smaller than that of an omni-directional

sensor, and also it may cover no target when distributed, it is important to adjust the working direction of the sensors in the network so that they can monitor all the targets in the network. A subset of working directions selected to cover all the targets in the network is known as cover set. It should be noted that in one cover set, we cannot have more than one direction of a sensor.

Although the literature consists of a number of algorithms proposed to solve the target coverage problem in DSNs, it lacks studies focusing on optimization algorithms such as Imperialist Competitive Algorithm (ICA) with a target-oriented point of view. In this paper, we propose a target-oriented IC-based algorithm as a solution to the target coverage problem. The proposed algorithm aims not only to construct an appropriate cover set, but also to extend the network life time as much as possible. In this study, several experiments were carried out in order to examine the effects of various parameters on the performance of the algorithm. The results were compared to those of a target-oriented greedy-based algorithm [18]. The final results showed that the proposed algorithm outperformed the other one in terms of solving the problem in hand.

The rest of the paper is organized as follows: Section 2 briefly reviews the studies related to solving the target coverage problem. Section 3 introduces target coverage problem in DSNs. Section 5 gives a brief description of ICA and proposes an algorithm for solving the defined problem. In Section 6, the performance of the proposed algorithm is evaluated through several experiments. Finally, Section 7 contains the conclusion of the study.

2 Related work

The present study is aimed at solving the target coverage problem and, at the same time, maximizing the network life time with the use of the scheduling technique. The primary aim of the scheduling technique is to save the power source of sensors when they are scattered redundantly in an area of interest. This technique makes some of the sensors inactive (it sets them to power saving mode) and makes others active so that they can keep on the monitoring operation. Additionally, scheduling the sensors can significantly extend the network life time since the inactive sensors consume a negligible amount of energy and oscillating between active and inactive modes makes the sensor batteries last for a longer period of time [2]. In the following, some of the most brilliant solutions proposed based on this technique in both WSNs and DSNs are discussed.

In recent years, using the scheduling techniques, some researchers have attempted to find a solution to the target coverage problem in WSNs. Cardei and Du [19] proved NP-completeness of this problem and modeled it as disjoint cover sets each of which were capable of covering all the targets. In [20], the previously-noted study was extended to non-disjoint cover sets in which each sensor could be joined to more than one cover set. In [21], two greedy algorithms were proposed to maximize the number of cover sets and, at the same time, manage the critical targets. The authors in [22] attempted to prolong the network life time through the use of the optimization capability of memetic algorithms. The literature consists of several studies using learning automata to solve the target coverage problem [5, 6, 13, 23]. In [24], an efficient cover set algorithm based on ICA was proposed. In this algorithm, the authors used ICA to determine the sensor nodes

that must be selected in different cover sets. The algorithms designed for WSNs are not directly applicable to DSNs; this is because of the limitation in sensing angle of directional sensors. In the following, some of the important studies formerly carried out using the scheduling technique in solving the target coverage problem in DSNs are reviewed.

The research conducted by Ai and Abouzeid [3] was one of the first studies into the target coverage problem in DSNs. They modeled the problem of the maximum coverage with the minimum sensors in order to obtain maximum number of targets covered by minimum number of active sensors. The authors proposed several algorithms to solve the problem. In [4], the multiple directional cover set problem was defined and it was also proved as an NP-complete problem. To solve this problem, the authors introduced a number of heuristic algorithms. In [7], two algorithms were proposed to solve the target coverage problem; one of them was based on the greedy approach and the other one was designed based on the genetic algorithm. In [18], the heuristics already proposed to solve the target coverage problem in visual sensor networks (VSNs) were classified into two different groups: sensor-oriented and target-oriented. In the sensor-oriented approach, the exact coverage count in various spherical sectors of each sensor is determined concerning the sensors and their situations in the network. To count the coverage, one has to take into consideration the overlapping regions of the adjacent sensors so that redundant coverage can be avoided. On the basis of the coverage count, the algorithm is capable of iteratively choosing the sensors that are able to monitor maximum number of uncovered targets. On the other hand, in the target-oriented approach, the targets are taken into account instead of the sensors. Additionally, the authors proposed three target-oriented heuristics to solve the coverage problem in VSNs. Their results confirmed the superiority of the target-oriented heuristics over the sensor-oriented ones in terms of solving the problem.

To solve the target coverage problem in DSNs, researchers in [12] proposed two scheduling algorithms among which, one was greedy based capable of finding a solution faster than other algorithms proposed in the literature. Though, due to its local search, this algorithm was not able to find an optimal solution to the problem. To remove this problem, they designed the second one based on learning automata in a way to find optimal cover sets in DSNs. In [8], the authors made use of a 2-phase learning automata-based algorithm to solve the target coverage problem. In the first phase, they introduced a distributed algorithm to select the appropriate sensing directions of the available sensors. In the second one, a centralized learning automata-based algorithm was proposed to select proper sensor directions to construct appropriate cover sets. In [9], several sensor-oriented and target-oriented LA-based algorithms were proposed in order to solve the target coverage problem. They showed that all the proposed algorithms were successful in terms of solving the problem in hand; although, the target-oriented algorithms outperformed the sensor-oriented ones concerning the extension of network life time. The researchers in [10] applied LA to solving the connected target coverage problem. Their proposed algorithm fell into two phases. In the first phase, they used LA to select appropriate sensor directions in a way to provide coverage on the area of interest. In the second one, they made use of distributed LA in order to transfer the sensing data. In [15, 16], the authors addressed the coverage problem in a condition where the coverage requirement of each target was different from that of the others. To solve it, two LA-based algorithms were proposed, which

were able to construct cover sets containing sensor directions capable of satisfying the coverage requirement of all the targets in the network.

In [1], the researchers proposed two greedy-based algorithms to solve the target coverage problem in networks where sensors contained multiple sensing ranges. To do this, they used two power saving techniques, scheduling and adjusting, in order to extend the network life time as much as possible. They confirmed the efficiency of a combination of these two techniques in the maximization of network life time. Moreover, some authors concentrated on solving target coverage problem in DSNs, where the sensing radius of sensors is adjustable [25]. In this respect, they introduced a GA-based algorithm and studied the critical target in order to further increase network longevity. They indicated that the network longevity can be significantly increased by adjusting the sensing radius and using scheduling algorithms. In [26], the authors investigated the "heterogeneous coverage" problem in VSNs. In this case, the targets have different coverage requirements. The problem is mainly focused on maximizing the coverage of all targets in order to achieve the coverage requirement associated with each target by activating the minimum number of sensors. To solve the problem, they proposed two solutions based on Integer Linear Programming (ILP) and Integer Quadratic Programming (IQP). Once the problem is solvable, both ILP and IQP provide accurate solutions; however, as these solutions are not scalable due to the computational complexity, a greedy solution was also proposed to solve the heterogeneous coverage problem.

In [27], the authors introduced the balanced k -coverage problem. The problem aims to avoid the situations in which k -coverage is provided merely for some targets in the network, while the rest are left uncovered or covered individually. To solve this problem, they proposed three solutions based on ILP, IQP, and Integer Non-Linear Programming (INLP). Due to the high computational cost of these algorithms and their NP-completeness, the authors introduced a new solution, called the Greedy k -Coverage Algorithm (CGkCA) Centralized, which performed computational operations more properly. In [14], the authors studied the balanced k -coverage problem and proposed a target-oriented LA-based solution to solve the problem. LA are used mainly to select a subset of sensors with suitable working directions providing coverage to all targets in a balanced way with the least number of sensors. To evaluate the algorithm performance, the results were compared with those of a greedy algorithm. The results confirmed the efficiency of the LA-based algorithm in solving the problem. Although the literature consists of several solutions to target coverage problem, no solution has been proposed holding a target-oriented point of view using ICA. Therefore, in this paper, we are to solve the target coverage problem considering both the capabilities of ICA and efficiency of the target-oriented point of view.

3 Problem Definition

The following scenario is taken into consideration in the present paper. In a 2-D Euclidean field, several targets are distributed with known locations and a number of directional sensors are also scattered randomly in the field to cover the targets. Each directional sensor possesses several non-overlapping directions; at each unit of time, only a single direction of a sensor has the chance to be activated, which is known as working direction. On each directional sensor, a special device is

installed, which is responsible for switching the direction of the sensor based on current conditions. A target is covered by a directional sensor if it is situated within both the sensing range of a sensor and the field of view of that sensor. Table 1 presents the notations used in this paper.

Problem: How to determine a subset of directional sensors in a way to provide all targets in the network with full coverage.

Definition: A target is critical if it is covered using the minimum number of sensor directions.

Table 1 Notations

Notation	Meaning
N	Number of sensors
m	Number of targets
w	Number of directions per sensor, $w \geq 1$
$d_{i,j}$	j -th direction of i -th sensor
s_i	A sensor, for all $i = 1, 2, \dots, N$
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, $T = t_1, t_2, \dots, t_M$

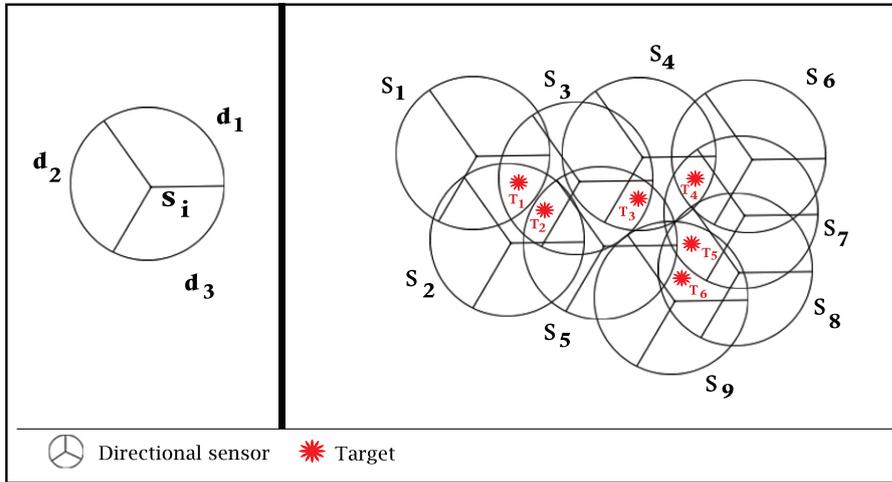


Fig. 1 An example network with nine directional sensors and six targets

To further clarify the problem, Fig. 1 shows a snapshot of a network containing six targets and nine directional sensors. In this example network, each directional sensor consists of three directions. The targets monitored by the sensor directions are as follow: $t_1 = \{d_{1,3}, d_{2,1}, d_{3,2}\}$, $t_2 = \{d_{2,1}, d_{3,2}, d_{5,2}\}$, $t_3 = \{d_{3,3}, d_{4,3}, d_{5,1}\}$, $t_4 = \{d_{4,3}, d_{6,2}, d_{7,2}\}$, $t_5 = \{d_{7,2}, d_{8,2}, d_{9,1}\}$, and $t_6 = \{d_{8,2}, d_{9,1}\}$. The problem is how to form an appropriate cover set that is able to monitor all the targets. An example solution to this problem is $\{d_{2,1}, d_{4,3}, d_{9,1}\}$.

4 Imperialist Competitive Algorithm

Since the 1990s, many researchers have frequently applied the meta-heuristic algorithms to solving the combinatorial optimization problems. Many of these algorithms, e.g., Ant Colony Optimization, Memetic Algorithm, Particle Swarm Optimization, and Simulated Annealing, have been found successful in the solution of many complex optimization problems. Among all, the Imperialist Competitive Algorithm (ICA) has received much attention from the research community in recent years. ICA was pioneered by Atashpaz Gargari inspired from the socio-political evolutionary process of human communities in order to develop a robust optimization algorithm [28]. The imperialism as a level of human social evolution was taken into consideration and through mathematical modelling of such a complex political process, an algorithm was configured for evolutionary optimization. ICA can be applied to many optimization problems in various fields like computer science, etc. Fig. 2 shows the flowchart of the ICA approach.

Similar to the other evolutionary algorithms, the starting point of ICA is an initial population in which each individual plays the role of a country. The countries are separated into two groups: colonies and imperialists, which together build empires. The core of this algorithm is the imperialistic competition among the empires, through which weaker empires gradually lose their power and finally collapse. The collapsed empires join the other empires as new colonies. The imperialistic competition continues until there exists only a single empire possessing all other countries as its colonies. In this situation, all the colonies will be equal in their cost function values, which is also equal to that of the imperialist. When all colonies are distributed to the existing imperialists to form the initial empires, on the basis of the assimilation policy, the colonies move towards their corresponding imperialists [29,30]. The imperialists apply an assimilation policy along with the direction of different optimization axes with the aim of obtaining their colonies' favors. To measure each empire's total power, the summation of the imperialist power and the percentage of the mean power of its colonies is computed. When the empires are initially formed, they start to engage in the imperialistic competition. The empire(s) without any success in this competition will be removed from the competition process. As a result, the survival of an empire will be dependent upon the power of the empire for the assimilation of the colonies of the competitors. Thus, greater empires gradually gain more power through the imperialistic competition process and, on the other hand, weaker empires are removed from the scene. To gain more power, empires need to make improvements inside their colonies. Therefore, colonies sooner or later become like empires from the perspective of power, which causes a kind of convergence. The termination criterion of ICA is the existence of only one empire.

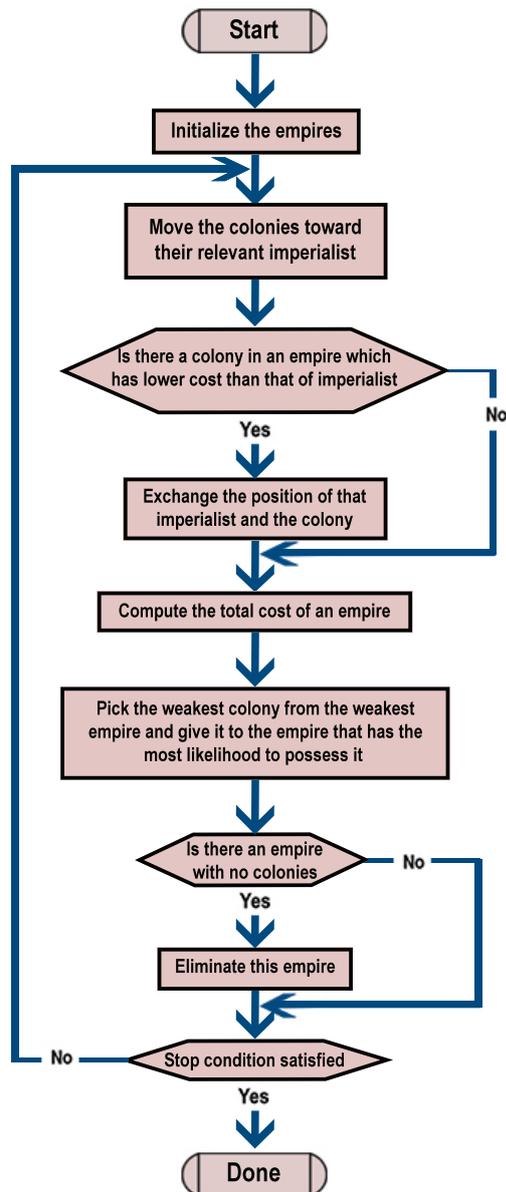


Fig. 2 Illustration of ICA flowchart.

5 Proposed Algorithm

In this section, a novel scheduling algorithm is proposed on the basis of ICA to solve the problem in hand mainly with the aim of constructing cover sets containing minimum number of sensor directions capable of providing coverage on all targets

in the network. Algorithm 1 shows the pseudo code of the proposed algorithm. ICA was implemented to construct a cover set needed in this study. In the following, the proposed algorithm is explained in detail.

Algorithm 1 : The Proposed ICA Procedure

01. **Input**: Directional network $Net = (S, T)$
 02. **Output**: A cover set with appropriate sensor directions
 03. Initialize the ICA parameters, i.e., N_{pop} , N_{imp} , and $MaxIter$;
 04. Produce the N_{pop} number of countries, and then select the N_{imp} number of the best countries as imperialists and determine their colonies based on their power;
 05. If the termination criterion is not satisfied yet, repeat the following steps;
 06. Assimilation;
 07. Imperialistic competition;
 08. Revolution;
 09. Repair invalid countries;
 10. Elimination of powerless empires;
 11. **return** (the best candidate solution)
-

5.1 The Initial Population Generation

To solve a problem using ICA, one of the key steps is translating the available solutions in the search space of the problem into countries. For this purpose, the present study applies an integer-based representation in order to model a country (regardless of being imperialist or colony). According to this model, each country denotes a cover set and each variable of the country represents a working direction of a sensor selected to cover a certain target from among all sensor directions that cover it. Each country is encoded by means of an integer vector of length m ; the parameter m stands for the total number of targets in the DSN. The reason why we configure the country this way is that in literature, it is confirmed that the target-oriented algorithms perform their tasks with a higher efficiency compared to sensor-oriented ones; thus, we take the same strategy in proposing our model. The algorithm can select the value of a variable existing in a country from among the sensor directions covering the targets that correspond to that variable. When the algorithm is to create a country (by allocating a value to its variables), it first attempts to identify the most critical target, then it assigns a value to the variable of that target; in other words, it selects a sensor direction from among the sensors covering that target. Remember that in directional sensors, at each unit of time, only one direction of a sensor can be applied to construction of a cover set. Accordingly, when a direction of a sensor is selected to allocate a value to the variable corresponding to a target, the other directions of that sensor cannot be applied to the same task for the other targets. The process during which a critical target is found and a value is assigned to its corresponding variable continues until the variables of all targets are assigned with appropriate values. This is done because in real-life applications, some targets might be positioned in areas where they can be monitored by only one or a few sensor directions. To have a maximized coverage and prolonged the network lifetime, these targets (recognized as critical targets) need to be covered first. Consequently, the algorithm has to

determine the priority of the targets based on their level of vulnerability. If a country is modeled this way, it can result in producing a feasible country. In the proposed algorithm, a feasible country refers to one that is able to provide all targets with full coverage. Note that in random deployment environments, generally the number of targets is fewer than the sensors. In such condition, this model is able to shrink the length of the country. A set of countries produced in a random way creates the initial population. In the proposed algorithm, the cost of each country is calculated by the fitness function. Generally, fitness value computes the quality level of a country on the basis of the given objectives. The aim of the proposed algorithm is selecting a minimum number of active sensor directions from among available sensor directions so that all the targets in the network can be fully covered. Remember that the fewer active sensor directions compared to the other cover sets, the better fitness function is achieved. As a result, the proposed algorithm aims at determining the candidate solutions that contain minimum number of active sensor directions, which also can prolong the network lifetime.

Here, we present an example to further clarify the proposed model. As shown in Fig. 1, as six targets exist in the network, the length of each country is fixed at 6. In other words, each variable represents the status of its corresponding target. To form a country, (through allocating values to its variables), at first, the most critical target must be identified. As target t_1 in this example is covered by the minimum number of sensor directions, it is marked out as the first critical target and its variable is allocated with a value. In other words, a sensor direction is selected from among all sensor directions covering that target. Let us assume that sensor direction $\{d_{1,3}\}$ has been selected to cover the critical target. Afterward, if one of the other directions of the same sensor (i.e., $\{d_{1,2}\}$ and $\{d_{1,2}\}$) are in the list of the directions covering the other targets, they will be removed from those lists. The process of selecting a critical target and allocating its variable value continues until the country in hand is formed completely (i.e., all variables have their values). Figure 3 shows an example country. The fitness function of the country depicted in Fig. 3 is 5.

Target ID	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆
Values	d ₁₃	d ₂₁	d ₃₃	d ₄₃	d ₉₁	d ₉₁

Fig. 3 An example country for the network.

At the outset, a number ($N_{country}$) of countries are formed, and a number (N_{imp}) of the optimum countries among the initial population (i.e., those of the minimum cost functions) are chosen as imperialist. The rest of the countries (N_{col}) are given the role of colonies each of which belongs to an empire. The power of each imperialist determines how many colonies it can possess. For this purpose, the normalized cost (C_i) of each imperialist (i) is calculated using Equation 1 on

the basis of the all imperialists' costs:

$$C_n = \max(c_i) - c_n \quad (1)$$

where c_n stands for the cost of the n -th imperialist, and C_n denotes its normalized cost. The distribution of the colonies amongst the imperialists is performed considering their Euclidian distance. For each imperialist, the normalized power is defined as follows:

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (2)$$

Then, the number of colonies of an empire is calculated as follows:

$$NC_n = \text{round}(P_n \cdot N_{col}) \quad (3)$$

where *round* signifies a function obtaining the closest integer to a decimal number. For each imperialist, the initial number of colonies is determined in a random way. The imperialistic competition starts after determining the initial state of the imperialists. The evolution process continues until the termination criterion is met. Obviously, the more powerful the imperialist, the more colonies it will possess.

5.2 Assimilation Process

Basically, imperialists in ICA attempt to make their colonies assimilated, i.e., to make them similar to themselves. This objective is realized by the gradual movement of colonies towards imperialists, Assimilation policies are taken into action by the central government depending upon the way a country is represented for solving an optimization problem. Such movement is interpreted as the optimization part of the ICA algorithm, which moves colonies towards the culture of the imperialist. In fact, an operation is executed with the aim of making part of the colonies' structures similar to the imperialist's structure. Original ICA has typically been used for the continuous problems. For discrete problems like one described in this study we use the method introduced in [24]. This operation is taken into action through the following steps:

1. A number of cells are chosen as imperialists in a random way (cells 1, 2, and 4 in the Fig. 4).
2. The cells chosen are copied directly into the New-Colony array at the same indexes.
3. The rest of the New-Colony array are copied from the Colony array at the same indices (cells 3, 5, and 6 in the Fig. 4).

After the assimilation process, there is a chance for invalid countries (New-Colonies) to be produced. A country is invalid, if in allocating values to its variables, more than one direction of a single sensor is used. To change the validity status of a country (i.e., changing an invalid country to a valid one), we propose a repair operator(see section 5.4).

Imperialist	d_{21}	d_{21}	d_{33}	d_{43}	d_{91}	d_{91}
Colony	d_{13}	d_{21}	d_{43}	d_{72}	d_{91}	d_{91}
New-colony	d_{21}	d_{21}	d_{43}	d_{43}	d_{91}	d_{91}

Fig. 4 An example of one assimilation process.

5.3 Revolution Operation

For the purpose of performing the revolution operation, one variable of the country is chosen in a random way and its value is changed. That is, the value of a variable corresponding to a certain target is shifted to a value selected from among the other sensor directions that are monitoring the target. Remember that when a new value is selected for the revolution operation, it is important to avoid invalidity of the country. For this purpose, when a sensor direction is selected to allocate the value to a variable, the value needs to be compared to the values of other variables of a country. If the selected value does not cause the invalidity of the country, it is applied; otherwise, a new value is chosen. This step (i.e., revolution operation) is repeated on the basis of a percentage of the total number of deployed targets, which is expressed by the *%Revolutionparameter*. If the new colony is better than the old one, the old colony will be replaced with the new one; otherwise, the process is repeated. Fig. 5 displays the operation in a graphical way.

Colony	d_{13}	d_{21}	d_{33}	d_{43}	d_{91}	d_{91}
New-colony	d_{21}	d_{21}	d_{33}	d_{43}	d_{91}	d_{91}

Fig. 5 An example of one revolution operation.

5.4 Repair operator:

In this study, a repair operator is proposed in order to assure the validity of new colony resulted from the assimilation and revolution operations. The repair operator takes the value of the variable corresponding to each target. Each value determines two parameters: the sensor number and the direction number of that sensor applied to a country. Next, we check whether the other directions of that sensor are used in allocating value to the variables corresponding to other targets. This process is done to every individual variables; in case invalid values are found for more than one variable, the most critical variable (target) must be identified,

and its value must be changed. This process continues until all variables get a valid status.

New-colony	d_{13}	d_{32}	d_{33}	d_{72}	d_{72}	d_{91}
Repaired-colony	d_{13}	d_{32}	d_{51}	d_{72}	d_{72}	d_{91}

Fig. 6 An example for the repair operator.

5.5 Position Exchanges of Colony and Imperialist

Through the process of moving toward the imperialist, a colony might achieve a position with lower cost compared to its corresponding imperialist. In this happens, the colony takes the role of the imperialist and the old imperialist becomes a new colony. Fig. 7.a depicts the position exchange of the colony and imperialist. In this figure, the best colony (i.e., the one with the cost less than that of the imperialist) is shaded. Then, the post-exchange empire is displayed in Fig. 7.b.

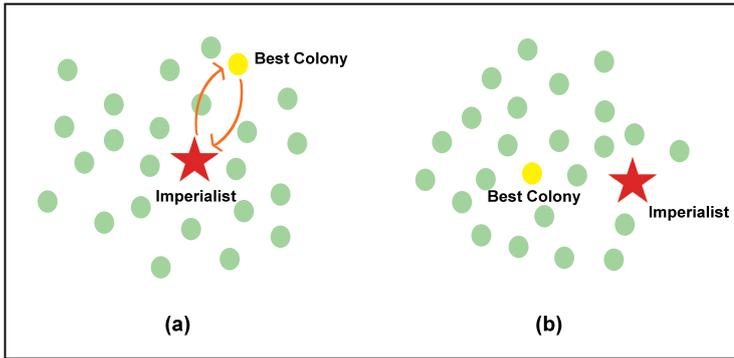


Fig. 7 a. Exchanging the Position of a colony and the imperialist, b. The entire empire after position exchange.

5.6 Total Power Calculation

To measure an empire's total power, one should calculate the summation of the imperialist's power and some percentage of the power of all the colonies belonging to it. As a result, the total cost TC_i of the i -th empire is calculated as follows:

$$T.C_i = cost(imperialist_i) + \zeta mean(cost((colonies of empire))) \quad (4)$$

where ζ stands for a positive real number ranging from 0 to 1. In case a small value is determined for the parameter ζ , the total cost of an empire equals the

cost of its corresponding imperialist. On the other hand, any increase in ζ will lead to increasing the colonies' costs impact upon the total cost of the empire.

5.7 Imperialistic Competition

All empires in ICA engage in a severe competition to take control of more colonies. This competition makes weak empires gradually even weaker, while it makes the powerful empires even more powerful. This competition process is as follows: the weakest colony of the weakest empire is taken from its corresponding imperialist and then it will be awaiting to be possessed a powerful empire. However, the question of which empire will take the control of the freed colonies is answered considering the total power of the other empires; the empires with higher total power has more chance to possess the new colony.

As mentioned earlier, the empires that cannot increase their power will be gradually eliminated from the competition course. At each iteration, the weakest colonies of an empire are chosen, and all other empires compete with each other to take the control of the selected colonies. Note that these colonies are not possessed necessarily by the most powerful empire; however, more powerful empires are more likely to take the possession. For the purpose of modelling such competition, the normalized total cost $N.T.C_i$ of each empire is first computed by means of Equation 5, on the basis of the empire's total cost $T.C_i$ and Equation 6, based on the empire's total cost $T.C_i$ and

$$N.T.C_i = \max_j(T.C_j) - T.C_i \quad (5)$$

Higher normalized total costs are allocated to the empires that have lower total costs. The probability P_{pi} of taking possession of each empire (which is a value proportional to the empire's power) is computed using Equation 1:

$$P_{pi} = \left| \frac{N.T.C_i}{\sum_{j=1}^{N_{imp}} N.T.C_j} \right| \quad (6)$$

and

$$P = [P_{p1}, P_{p2}, P_{p3}, \dots, P_{(pN_{imp})}] \quad (7)$$

Afterwards, vector R with the same size as P is formed, and its elements are distributed uniformly over random numbers:

$$R = [r_1, r_2, r_3, \dots, r_{(N_{imp})}] \quad (8)$$

where

$$r_1, r_2, r_3, \dots, r_{(N_{imp})} \in U(0, 1) \quad (9)$$

Then, vector D is created through subtracting R from P :

$$D = P - R = [D_1, D_2, D_3, \dots, D_{(N_{imp})}] = [P_{p1} - r_1, P_{p2} - r_2, \dots, P_{(pN_{imp})} - r_{(N_{imp})}] \quad (10)$$

5.8 Powerless Empires Elimination

As explained earlier, powerless empires are removed gradually from the course of imperialistic competition, and their colonies will be possessed by more powerful empires. In ICA, the entire elimination of an empire occurs once it loses all of its colonies; after that, the rest of existing empires compete for the eliminated one.

5.9 Termination Criteria

The convergence condition proposed in this paper is met in two states: either the total number of iterations is completed, or all but one of the empires have collapsed. In either case, the imperialistic competition is terminated.

6 Simulation Results

In this section, the efficiency of the ICA target-oriented approach was examined through conducting several experiments, and then the obtained results were compared to those of a target-oriented greedy-based algorithm [4]. To have a reliable comparison, several metrics of performance evaluation were used to evaluate the performance of the proposed algorithm. The metrics were the ratio of active sensor direction (RASD), target coverage per sensor direction (TCPSD), and power consumption (PC). In order to model a DSN, N sensors with the same sensing range r and sensing angle $\frac{2\pi}{3}$ were distributed in a square area of the size $500(m) * 500(m)$, then M targets were also scattered uniformly and randomly within the same environment. By default, the sensing range was set to $100(m)$ and the numbers of sensors and targets were fixed at 120 and 10, respectively. Each scenario was run for 10 times; then for each scenario, the average value was computed. In the proposed algorithm, the value $P_{reveloutin}$ was fixed at 0.1.

Experiment 1. This experiment measured the effect of the number of sensors on three metrics of RASD, TCPSD, and PC. For this purpose, the number of sensors was between 80 and 120 with incremental step of 10. The results presented in Table 2 show that with increasing the number of sensor directions, the ratio of active sensor directions decreased for both algorithms. The reason was that when the number of sensor directions rises, the sensors capable of covering more targets are generally added to the network. The obtained results confirmed the superiority of the proposed ICA-based algorithm over the greedy-based algorithm in terms of the ratio of active sensor directions. This was due to the fact that the proposed algorithm is expected to construct the cover sets containing minimum sensor directions and, at the same time, provide full coverage to all the targets in the network.

Table 2 Effect of the number of sensors on the network

Number of sensors	RASD		TCPSD		PC	
	Greedy	ICA	Greedy	ICA	Greedy	ICA
80	0.1025	0.09	1.2302	1.3988	47.362	42.152
90	0.094444	0.078889	1.1845	1.4325	49.505	42.211
100	0.083	0.072	1.2163	1.3988	49.043	43.312
110	0.072727	0.066364	1.2679	1.381	48.06	44.413
120	0.0675	0.058333	1.2401	1.4464	49.161	43.4

Experiment 2. In this experiment, we measured the ratio of covered targets to the activated sensor directions. In other words, this metric shows the average number of targets covered by each activated sensor direction. We changed the number of targets from 12 to 20 with incremental step of 2. The results (see Table 3) showed that with an increase in the number of targets, the ratio of the covered targets in both algorithms increased, too. This is reasonable since with the addition of new targets to the network, each sensor is able to cover more targets. According to the comparative results, the proposed algorithm outperformed the greedy-based one in terms of accomplishing the defined objectives.

Table 3 Effect of the number of targets on the network

Number of targets	RASD		TCPSD		PC	
	Greedy	ICA	Greedy	ICA	Greedy	ICA
12	0.084167	0.066667	1.1958	1.5095	59.581	48.64
14	0.096667	0.081667	1.2174	1.4368	67.396	58.018
16	0.10667	0.089167	1.2568	1.503	73.648	62.707
18	0.1125	0.098333	1.3454	1.5315	77.295	68.438
20	0.11667	0.10583	1.4396	1.5824	79.9	73.127

Experiment 3. This experiment measured the effect of sensing range on three metrics of RASD, TCPSD, and PC. To do this, we changed the sensing range from $80(m)$ to $120(m)$ with incremental step of $10(m)$. The results summarized in Table 4 demonstrate that with an increase in the sensing range, RASD decreased, while TCPSD increased. This is because with increasing the sensing range, each sensor direction is capable of covering more targets resulting in the decrease of RASD and increase of TCPSD. Table 4 also displays the power consumption of both algorithms, which was obtained using the equation used in [18]. The proposed algorithm was found to consume less energy since it attempts to use fewer active sensor directions in the cover set construction process.

Table 4 Effect of the sensing range on the network

Sensing range	RASD		TCPSD		PC	
	Greedy	ICA	Greedy	ICA	Greedy	ICA
80	0.0725	0.063333	1.1817	1.3452	52.287	46.556
90	0.068333	0.058333	1.2496	1.4524	49.682	43.43
100	0.065	0.055833	1.3325	1.5119	47.598	41.867
110	0.0625	0.054167	1.3873	1.5536	46.035	40.825
120	0.058333	0.048333	1.4813	1.7429	43.43	37.178

7 Conclusion

This study was carried out to solve the target coverage problem in networks comprising sensors with limited sensing angle (i.e., directional sensors). It was aimed at prolonging the network life time as much as possible. To this end, a target-oriented scheduling algorithm on the basis of ICA was proposed. More specifically, the aim of the proposed algorithm was choosing the most suitable directions of the available sensors in a way to construct cover sets capable of covering all the targets in the given network and, at the same time, maximize the network life time. This study had some innovations and contributions: first, due to the efficient performance of the target-oriented algorithms, we attempted to represent the countries in ICA in a target-oriented way; the critical targets were managed at the time of countries production, a simple fitness function was proposed, and a repair operator was also introduced in order to treat the invalidity of the countries (if any). Moreover, to examine the efficiency of the algorithm performance, several experiments were conducted, and the results were compared to those of a target-oriented greedy-based algorithm proposed in the literature. The comparison confirmed the superiority of the proposed algorithm over the other one in terms

of constructing cover sets with fewer active sensor directions, which resulted in a considerable extension of the network life time.

References

1. Mohamadi, H., Salleh, S., & Norsyarizad Razali, M. (2014). Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges. *Journal of Network and Computer Applications*, 46, 26-35.
2. Amac Guvensan, M., & Gokhan Yavuz, A. (2011). On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks*, 9(7), 1238-1255.
3. Ai, J., & Abouzeid, A. A. (2006). Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11(1), 21-41.
4. Cai, Y., Lou, W., Li, M., & Li, X.-Y. (2009). Energy Efficient Target-Oriented Scheduling in Directional Sensor Networks. *IEEE Trans. Comput.*, 58(9), 1259-1274.
5. Mohamadi, H., Ismail, A. S., Salleh, S., & Nodhei, A. (2013). Learning automata-based algorithms for finding cover sets in wireless sensor networks. *The Journal of Supercomputing*, 66(3), 1533-1552.
6. Mohamadi, H., Ismail, A. S., & Salleh, S. (2014). Solving target coverage problem using cover sets in wireless sensor networks based on learning automata. *Wireless Personal Communications*, 75(1), 447-463.
7. Gil, J.-M., & Han, Y.-H. (2011). A Target Coverage Scheduling Scheme Based on Genetic Algorithms in Directional Sensor Networks. *Sensors*, 11(2), 1888.
8. Mohamadi, H., Ismail, A. S. B. H., & Salleh, S. (2013). A learning automata-based algorithm for solving coverage problem in directional sensor networks. *Computing*, 95(1), 1-24.
9. Mohamadi, H., Ismail, A. S., Salleh, S., & Nodehi, A. (2013). Learning automata-based algorithms for solving the target coverage problem in directional sensor networks. *Wireless Personal Communications*, 73(3), 1309-1330.
10. Mohamadi, H., Ismail, A. S., & Salleh, S. (2013). Utilizing distributed learning automata to solve the connected target coverage problem in directional sensor networks. *Sensors and Actuators A: Physical*, 198, 21-30.
11. Zhu, C., Zheng, C., Shu, L., & Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2), 619-632.
12. Mohamadi, H., Salleh, S., Ismail, A. S., & Marouf, S. (2015). Scheduling algorithms for extending directional sensor network lifetime. *Wireless Networks*, 21(2), 611-623.
13. Mohamadi, H., Salleh, S., Razali, M. N., & Marouf, S. (2015). A new learning automata-based approach for maximizing network lifetime in wireless sensor networks with adjustable sensing ranges. *Neurocomputing*, 153, 11-19.
14. Bakht, A. J., Motameni, H., & Mohamadi, H. (2020). A learning automata-based algorithm to solve imbalanced k-coverage in visual sensor networks. *J. Intell. Fuzzy Syst.*, 39(3), 2817-2829.
15. Razali, M. N., Salleh, S., & Mohamadi, H. (2017). Solving priority-based target coverage problem in directional sensor networks with adjustable sensing ranges. *Wireless Personal Communications*, 95(2), 847-872.
16. Mohamadi, H., Salleh, S., & Ismail, A. S. (2014). A learning automata-based solution to the priority-based target coverage problem in directional sensor networks. *Wireless Personal Communications*, 79(3), 2323-2338. doi:10.1007/s11277-014-1987-5
17. Singh, A., & Rossi, A. (2013). A genetic algorithm based exact approach for lifetime maximization of directional sensor networks. *Ad Hoc Networks*, 11(3), 1006-1021.
18. Zannat, H., Akter, T., Tasnim, M., & Rahman, A. (2016). The coverage problem in visual sensor networks: A target oriented approach. *Journal of Network and Computer Applications*, 75, 1-15. doi:https://doi.org/10.1016/j.jnca.2016.08.015
19. Cardei, M., & Du, D.-Z. (2005). Improving Wireless Sensor Network Lifetime through Power Aware Organization. *Wireless Networks*, 11(3), 333-340.
20. Cardei, M., Thai, M. T., Yingshu, L., & Weili, W. (2005). Energy-efficient target coverage in wireless sensor networks. In Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.

-
21. Zorbas, D., Glynos, D., Kotzanikolaou, P., & Douligeris, C. (2010). Solving coverage problems in wireless sensor networks using cover sets. *Ad Hoc Networks*, 8(4), 400-415.
 22. Ting, C.-K., & Liao, C.-C. (2010). A memetic algorithm for extending wireless sensor network lifetime. *Information Sciences*, 180(24), 4818-4833.
 23. Mostafaei, H., & Meybodi, M. R. (2013). Maximizing lifetime of target coverage in wireless sensor networks using learning automata. *Wireless Personal Communications*, 71(2), 1461-1477.
 24. Mostafaei, H., & Shojafar, M. (2015). A new meta-heuristic algorithm for maximizing lifetime of wireless sensor networks. *Wireless Personal Communications*, 82(2), 723-742.
 25. Alibeiki, A., Motameni, H., & Mohamadi, H. (2019). A new genetic-based approach for maximizing network lifetime in directional sensor networks with adjustable sensing ranges. *Pervasive and Mobile Computing*, 52, 1-12.
 26. Zishan, A. A., Karim, I., Shubha, S. S., & Rahman, A. (2018). Maximizing heterogeneous coverage in over and under provisioned visual sensor networks. *Journal of Network and Computer Applications*, 124, 44-62.
 27. Malek, S. M. B., Sadik, M. M., & Rahman, A. (2016). On balanced k-coverage in visual sensor networks. *Journal of Network and Computer Applications*, 72, 72-86.
 28. Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In IEEE congress on evolutionary computation.
 29. Atashpaz Gargari, E., Hashemzadeh, F., Rajabioun, R., & Lucas, C. (2008). Colonial competitive algorithm. *International Journal of Intelligent Computing and Cybernetics*, 1(3), 337-355.
 30. Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In IEEE congress on evolutionary computation, pp. 4661-4667.