A Hybrid Algorithm for Q-coverage Problem in Under Provisioned Directional Sensor Networks

Babak Mahmoudi[1], Homayun Motameni[2*], Hosein Mohamadi[3]

Abstract–In the context of emerging technologies like the Internet of Things (IoT) and home automation, researchers are increasingly exploring the use of Wireless Sensor Networks (WSNs). One of the primary functions of these sensors is coverage, which refers to how effectively they can monitor targets within a given environment. In many applications, it is essential that each target is covered by at least one sensor; this is known as "Simple Coverage." In other cases, multiple sensors may be required to cover a single target, a situation referred to as "Multiple Coverage." When the number of sensors covering a target can vary, it is termed "Q-coverage." When there are not enough sensors in an environment, achieving balanced coverage becomes critical. To address this challenge, current research presents a hybrid algorithm that combines genetic algorithms and Tabu Search as a promising solution for monitoring targets in such under-provisioned environments. To evaluate the effectiveness of this proposed algorithm, several experiments were conducted, and the results were compared with those obtained from a genetic algorithm introduced in recent studies.

Keywords: Coverage, Genetics Algorithm, Tabu Search

1. Introduction

Wireless sensor networks (WSNs) consist of a series of sensor nodes specifically designed to collect, process, and transmit information from their surrounding environment. These networks are used in various fields, including environmental monitoring, security, agriculture, industry, and the military. In directional sensor networks, each sensor node has a limited field of view and can only gather information from a specific direction. This limitation presents significant challenges in the design and management of these networks. A critical concept in designing such networks is coverage, which plays an essential role in their overall performance and efficiency [1].Coverage in directional sensor networks can be categorized into three main types: Target Coverage, Area Coverage, and Barrier Coverage. Target Coverage focuses on covering specific targets within the environment. The primary goal is to achieve complete coverage of these targets using the fewest number of sensors. This type of coverage is commonly applied in security and the monitoring of sensitive locations, such as gates, military bases, and critical infrastructure. Area Coverage aims to ensure that an entire region is adequately covered by distributing sensors so that no point within the area remains uncovered. This type of coverage is utilized in environmental monitoring, smart agriculture, and urban surveillance. Barrier Coverage seeks to establish a sensory barrier that prevents objects or individuals from crossing a designated boundary. This coverage is particularly relevant in applications like border security, safeguarding critical infrastructure, managing vital areas, and creating sensory barriers around essential locations such as power plants, chemical storage facilities, or weapon depots [2].

Despite their wide range of applications, Wireless Sensor Networks (WSNs) have significant limitations. The sensor nodes are small and powered by tiny batteries, which means the lifespan of a sensor network is heavily dependent on how long these nodes last. A major challenge is to extend the network's overall lifespan. WSNs are often deployed in hard-to-reach areas, such as hazardous chemical sites or hostile military zones. In some scenarios, like forest monitoring, the operational environment can be vast, making it difficult to search for, recharge, or replace batteries. There are several methods for deploying sensors in an environment, with two common approaches being predetermined placement and random deployment. In predetermined placement, the positions of the sensors are decided in advance. In random deployment, a relatively large number of sensors are scattered throughout the environment without predetermined positions. This randomness can lead to some sensors being placed in unfavorable locations, such as in water or behind obstacles, which renders them ineffective. Additional challenges include potential sensor damage, consumption by animals, and other factors. Because some targets may remain uncovered due to these issues, it is essential to deploy extra sensors to ensure coverage, depending on the specific type of network. Coverage can be divided into two categories: simple coverage and multi-coverage, based on the number of sensors required to cover each target. In simple coverage, each target requires coverage from at least one sensor. A failure or lack of coverage for some targets does not significantly impact the overall network performance. In contrast, multi-coverage requires that each target be

1 Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran. Email:bbmdi9696@gmail.com

2* Corresponding Author: Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran. Email:h_motameni@yahoo.com

3 Department of Computer Engineering, Azadshahr Branch, Islamic Azad University, Azadshahr, Iran. Email:h.mohamadi1983@gmail.com

Received: 2025.04.14; Accepted: 2025.06.01

covered by a greater number of sensors. When all targets must have coverage from a fixed number of sensors (like k), this is referred to as k-coverage. Since not all targets hold equal importance, and covering them all with the same number of sensors could increase network overhead, another coverage type is defined where the coverage requirements vary for different targets; this is known as Q-coverage [4].

2. Related Works

Directional sensor networks play a significant role in improving monitoring and data management with their unique capabilities. Applying these networks helps resource optimization, security enhancement, and service quality improvement in various domains. Extensive research has been conducted on simple and multi-coverage. In simple coverage, researchers have proposed several methods to maximize target coverage using the minimum number of active sensors. One of the earliest studies [6] offered a model for "Maximum Coverage with Minimum Sensors". This model was formulated using an Integer Linear Programming (ILP), and two greedy algorithms were developed to solve the problem: A centralized and a distributed algorithm. These algorithms were very optimized in terms of computational efficiency. Some research has been conducted to extend the network's lifespan in simple coverage. In research [7,8], some methods are proposed, including sensor scheduling and sensors' range adjustment. In research [7], two greedy algorithms were designed to make coverage sets that provide coverage in various steps. This study specifically emphasizes critical targets, which are those targets covered by consuming less energy. In [8], the problem of "Maximum Network Lifetime with Adjustable Ranges" (MNLAR) was investigated, and sensors were selected in terms of direction and range to create a coverage set. Research has also been conducted, focusing on multi-objective optimization, and we can imply research [2]. This study [2] uses an algorithm based on NSGA-II. Focusing on multi-objective optimization, this algorithm tries to simultaneously improve parameters such as the number of sensors and coverage quality.

In k-coverage, each target must be covered by a minimum of k sensors. Research has proposed methods to optimize k-coverage and extend network lifetime. Research [9] proposed a heuristic method focusing on reducing energy consumption. By preventing activation of unnecessary sensors, battery life and network lifespan will be increased. Study [10] investigated the k-coverage problem and two algorithms based on automated learning with the relevant rules. These rules prevented the selection of multiple directions for each sensor in a coverage set. Where the number of sensors is insufficient for a complete coverage of targets, it is called an under-provisioned environment, and some methods have been proposed for a balanced network coverage. In research [11], in addition to extending integer linear programming (ILP) for the simple coverage, a centralized greedy algorithm was designed to create balanced coverage in the network. Study [12] proposed an algorithm based on automated learning for selecting the minimum number of sensors for each coverage set and maintaining balanced coverage amongst targets. In sensor networks, heuristic algorithms are used to achieve an optimal method. In study [13], two genetics-based algorithms were designed to be applicable in over-provisioned and under-provisioned environments. These algorithms aimed to achieve balanced coverage in line with the Balance Index (BI). Q-coverage refers to a state where the coverage requirements of targets vary. Research in this area is highly complex because different targets may require different numbers of sensors. Study [14] suggests that the Q-coverage problem is of NP-complete type. Linear programming techniques were used to solve this problem. In research [15], a greedy algorithm was proposed to create non-overlapping coverage sets in which sensors with batteries of longer lifespan are of higher priority. In study [16], a genetic-based algorithm was designed to extend network lifetime and optimize resource consumption. In research [17], using IQP formulae and balanced index (BI), we attempted to provide a balanced coverage in the networks for targets. Additionally, greedy algorithms were designed to solve large-scale problems.

In study [4], the authors proposed two genetics-based algorithms to solve the Q-coverage problem in directional sensor networks (DSNs). This research has one important goal: Proposing approaches for covering targets under various conditions and optimizing network lifetime. The most significant works in [4] include examining target coverage in the over-provisioned and under-provisioned environments. Two target-based algorithms were developed. The first algorithm aims to cover all targets with the minimum number of active sensors, and the second algorithm is designed for under-provisioned environments to provide balanced coverage for different targets based on their coverage requirements. The authors developed a model for the chromosome that manages the varying coverage requirements of targets. This model allows the genetic algorithm to adapt to changes in coverage requirements.

The proposed algorithms are compared to algorithms of previous research using five indices. These five indices include Coverage Balanced Index (QBI), Coverage Quality (CQ), Distance Index (DI), Power Consumption (PC), and Number of Active Sensors (AS). Assessment is carried out using various criteria under various conditions (like change in the number of sensors, number of targets, and scope of sensor coverage).

3. Network Model and Problem Expression

Immersion of sensors in water, natural obstacles between sensors and targets, or early battery depletion are possible and may result in some targets not being properly covered. In such cases, each target must be covered with multiple sensors. Not all targets are equally important. For example, in a museum or gallery, not all objects may be important, and some may require coverage by more sensors. Monitoring all targets with a fixed number of sensors is not a wise approach as it increases network overhead and additional costs, which render it not cost-effective. Therefore, different targets should be covered by a varying number of sensors. To better understand the network model used in this research, Table 1 introduces the parameters used for the proposed algorithms.

Table 1: Parameters Used for the Proposed Algorithm

Definition	Symbol
Number of sensors	n
Number of Targets	m
Number of Sectors in Each Sensor	w
Sensor No. *i* in Network 1≤*i*≤n	si
Target No. *k* in Network 1≤*k*≤m	tk
Sensors Set {S_1, S_2, …, S_n}	S
Targets Set {t_1, t_2, …, t_m}	T
Sector *j* of Sensor *i*	$d_{i,j}$
Coverage Required by Target *t*	k_t
Coverage Obtained for Target *t*	ρ_k
A Set of All Sectors of Sensor D= {di,j|i= 1, …, n, j = 1, …, w}	D

3.1 Network Model

A directional sensor network consists of *m* targets and *n* directional sensors used for sensor coverage. In this environment, both the targets and sensors are stationary and do not move. All targets are positioned at known locations on a two-dimensional plane, while the directional sensors are randomly distributed near the targets.

3.2 Problem Statement

Here is an official definition of the problem.
Assumptions for the problem statement are as follows:

- A series of targets that need to be covered by the network; T= {t1, t2 … tm}
- Sensors with several sectors, and each could be active in a sector; S= {s1, s2 … sm}
- K= {k1, k2 … km}, that is a set of numbers, where ki is the required cover for ti in the network;

Problem: How can we find a coverage set that achieves non-uniform balanced coverage of targets with varying coverage requirements in an under-provisioned environment, in order to maximize balanced coverage within the network?

Proposed Algorithm

The proposed algorithm for solving the problem combines genetic algorithms with Tabu search, creating a hybrid approach. Traditional genetic algorithms are often not suitable for precise searches in complex hybrid spaces. However, hybrid algorithms, when combined with other techniques, enhance the efficiency of the search process. In this hybrid approach, the genetic algorithm first identifies the optimized region, and then a local optimizer, such as Tabu search, is employed to find the optimal solution. An HA-based algorithm has been developed to leverage the benefits of this combined strategy. By integrating genetic algorithms and Tabu search, this algorithm offers improved search capabilities [3].

3.3 An Overview of Genetic Algorithms

In evolutionary algorithm research, the genetic algorithm is the most widely used method. The primary distinction between evolutionary algorithms and other types is that evolutionary algorithms are population-based. Typically, these algorithms start by creating and evolving an initial population. They utilize a search method to find near-optimal solutions within a reasonable timeframe to optimize problems. The genetic algorithm begins with a primary population of potential solutions, each represented as a chromosome. All possible solutions must be encoded using a specific coding system. Next, a set of reproduction operators needs to be defined, as these operators directly affect the chromosomes. Following this, the chromosomes undergo mutation and crossover operations. It is crucial to design both the coding structure and the operators carefully, as this design significantly impacts the performance of the genetic algorithm [18]. The selection process involves competition among individuals in the population and is based on a competency function. Each chromosome has a value related to the quality of the solution it represents. The objective of the genetic algorithm is to maximize the value of this competency function. If the goal is to minimize a target function, it can be adjusted to reflect a straightforward minimization process. Any cost function can be easily transformed into a competency function. Once the reproduction steps and competency function are defined, a genetic algorithm can be developed based on this fundamental structure [19].

3.4 Proposed Genetic-Based Algorithm Chromosome Structure

The structure of chromosomes in a genetic algorithm is essential, and the first step is to figure out how to model and solve this structure. Each gene in the chromosome represents the sector number of a sensor, and each chromosome symbolizes a cover set. A two-dimensional matrix is employed to represent a chromosome; the number of rows indicates the maximum coverage required for the targets, while the number of columns signifies the number of targets within the network. The coverage requirement for each target is displayed by the number of non-empty rows beneath that target in the matrix. Some genes in the matrix may be empty since the coverage requirements for each target can differ. The values in a column correspond to the sensor numbers that cover the target in a specific sector. Figure 1 illustrates a chromosome with four targets to enhance understanding of the proposed model.

Target Number	t1	t2	t3	t4
Genes	$S_{2,3}$	$S_{1,1}$	$S_{2,1}$	$S_{8,6}$
	$S_{7,3}$	$S_{2,3}$	$S_{5,3}$	$S_{1,3}$
	$S_{9,2}$			$S_{3,3}$
	$S_{11,3}$			

Fig. 1. An example chromosome in the network

A set of chromosomes is produced to make the primary population randomly.

3.4.1 Chromosome Evaluation Function

To identify the best chromosomes based on the problem's conditions, we utilize an evaluation function. By appropriately setting the function's parameters, we can obtain an optimal or near-optimal solution within the search space. The evaluation function does not necessarily aim to fulfill all coverage requirements; instead, coverage is assessed based on the prioritization of targets or through balanced coverage across targets. This function employs a parameter to evaluate the chromosomes and determine their relative superiority. Equation 1-4 illustrates the evaluation function used within the network, wherekt is the total number of sensors required to cover the target, t and m are the number of targets, and Øt is the total number of sensors currently covering target t.

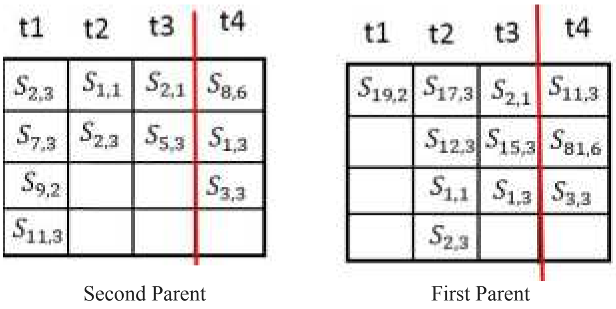
$$QBI = \frac{(\sum_{t=1}^{m} \varphi_t)^3}{\sum_{t=1}^{m} (\varphi_t)^2} \times \frac{(\sum_{t=1}^{m} (k_t)^2)}{(\sum_{t=1}^{m} k_t)^3} \tag{1}$$

3.4.2 Selection Operator

To enhance selection and reproduction, this operator identifies suitable chromosomes from the population, allowing them to be reproduced more frequently than others. Various selection methods exist in genetic algorithms (GAs), including roulette wheel selection, rank selection, and tournament selection [20]. In this algorithm, we use roulette wheel selection, which means chromosomes that perform better have a higher likelihood of being chosen.

3.4.3 Crossover Operator

Different methods for crossover performance have been proposed, including single-point, two-point, and uniform crossover. In these methods, the offspring inherits some genes from one parent and the remaining genes from the other parent. The crossover operator is utilized to create new offspring by combining genetic information from both parents. This approach generates new solutions within the population [21]. First, two parents must be selected.



t1	t2	t3	t4
$S_{2,3}$	$S_{1,1}$	$S_{2,1}$	$S_{8,6}$
$S_{7,3}$	$S_{2,3}$	$S_{5,3}$	$S_{1,3}$
$S_{9,2}$			$S_{3,3}$
$S_{11,3}$			

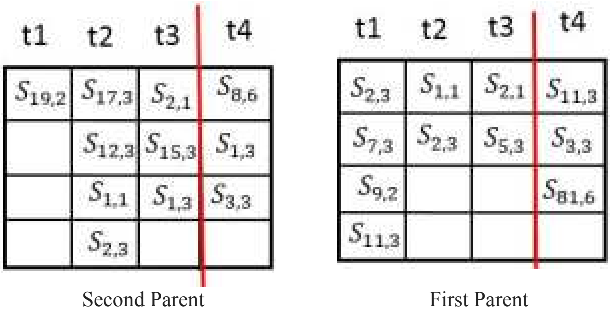
Second Parent

t1	t2	t3	t4
$S_{19,2}$	$S_{17,3}$	$S_{2,1}$	$S_{11,3}$
	$S_{12,3}$	$S_{15,3}$	$S_{81,6}$
	$S_{1,1}$	$S_{1,3}$	$S_{3,3}$
	$S_{2,3}$		

First Parent

Fig. 2. Single-Point Operator Performance

Next, a point is randomly selected, and two parent chromosomes transfer their genetic information to produce two offspring. After this phase, some offspring might have sensors in multiple sectors, which invalidates their chromosomes and requires correction.



t1	t2	t3	t4
$S_{19,2}$	$S_{17,3}$	$S_{2,1}$	$S_{8,6}$
	$S_{12,3}$	$S_{15,3}$	$S_{1,3}$
	$S_{1,1}$	$S_{1,3}$	$S_{3,3}$
	$S_{2,3}$		

Second Parent

t1	t2	t3	t4
$S_{2,3}$	$S_{1,1}$	$S_{2,1}$	$S_{11,3}$
$S_{7,3}$	$S_{2,3}$	$S_{5,3}$	$S_{3,3}$
$S_{9,2}$			$S_{81,6}$
$S_{11,3}$			

First Parent

Fig.3. Single-point crossover operator and second offering is correct

3.4.4 Mutation Operator

The mutation operation helps a genetic algorithm (GA) avoid getting stuck in local optima. A local optimum occurs when the algorithm searches for solutions within a limited area, without exploring the broader search space. The mutation operator introduces significant changes to the chromosome by randomly selecting a gene and altering its value [5].

3.4.5 Stopping Condition

One of the stopping conditions for the genetic algorithm is reaching a predetermined number of iterations. The stopping condition for the algorithm is the number of iterations specified in advance.

3.5 An Overview of Tabu Search

Tabu Search (TS) is an optimization algorithm first introduced by Glover in 1986 [22]. It employs a list of movements, known as Tabu points, to prevent revisiting these movements in subsequent searches. This approach enables the algorithm to move beyond local optimization and work towards achieving global optimization. The two main components of the Tabu Search algorithm are the

Tabu list and the aspiration criterion. The Tabu list tracks recent moves, thus avoiding their selection for as long as possible. The duration a move remains in the Tabu list is defined by a parameter called Tabu Tenure. If a move in the Tabu list could potentially lead to a better solution, it may still be selected based on the aspiration criterion, despite being in the list. Once a new move is chosen and added to the Tabu list, some previously listed moves may be removed [22]. In the proposed algorithm, a new function is introduced to enhance the quality of chromosomes, which can be applied to all chromosomes in the current population. Since evaluating all chromosomes is computationally expensive, extensive use across all potential solutions is impractical. This method effectively adjusts the convergence of Genetic Algorithms (GA), as Tabu Search typically enhances chromosomes. The steps for executing Tabu Search are as follows:

1. Begin by creating an initial solution based on the defined problem conditions. Evaluate this solution to determine if it is the best option available.

2. Prepare a list of allowed operations that are based on adjacent production methods.

3. Execute both allowed and non-Tabu operations to determine the solution according to the target function.

4. Select the best solution from those obtained in the previous step.

5. Update the Tabu list, which in the fast Tabu search algorithm involves adding the selected operation to the Tabu list and removing one or more operations from it. In this phase, also update the best solution found. 6. If the stopping criterion has not been met, return to step 4.

4. Experiments

To create a network scenario, various targets were randomly distributed within a 500m x 500m area. Several sensors, each with a sensing radius (r) and a sensing angle ($\pi/32$), were employed to monitor these targets. The coverage requirements for each target were established beforehand. To ensure more reliable results, each scenario was repeated ten times, and the average outcome was recorded as the algorithm's performance. In the experiments, the population size was set to 50, with crossover and mutation rates of 0.2 and 0.05, respectively. To accurately assess the algorithm's performance, a comparison was made with a recently proposed greedy algorithm (referenced as [4]). Performance evaluation utilized indices introduced in [4], which include the Distance Index (DI), Q-Balancing Index (QBI), and Active Sensor (AS).

The Distance Index (DI) metric is defined in [4]. In equation 2, k_t represents the required coverage for target t, while Øt indicates the coverage achieved for that target. A higher DI value signifies better network coverage, with a maximum possible value of 1.

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

The QBI metric serves as an assessment function for chromosomes. A higher QBI value indicates better-balanced coverage. According to equation 3, the maximum value of this metric is 1, which is achieved when the required coverage is provided for all targets.

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

4.1 Coverage Quality (CQ)

The CQ metric proposed in reference [27] is utilized in this research. An increase in the distance between targets and sensors leads to a decrease in the quality of coverage. Additionally, the distance between targets and sensors plays a significant role in determining the coverage quality within the environment. In a network, both the number of sensors and their distance from the targets influence the CQ value. This parameter, represented as cq, is calculated as follows:

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

4.2 First Experiment

Impact of Reducing the Number of Sensors on QBI

When there are enough sensors in a network, all targets can meet their coverage requirements, which results in a QBI value of 1. In this experiment, we decreased the number of sensors from 150 to 50 to observe the impact on QBI. As the number of sensors decreases, it becomes more challenging to meet the coverage requirements for all targets, necessitating a balanced level of coverage based on the available sensors. As illustrated in Figure 4, the reduction in the number of sensors leads to a gradual decline in the QBI value. In both algorithms being compared, the proposed algorithm consistently achieves a higher QBI value than the GA algorithm, indicating that it effectively maintains a more balanced level of coverage.

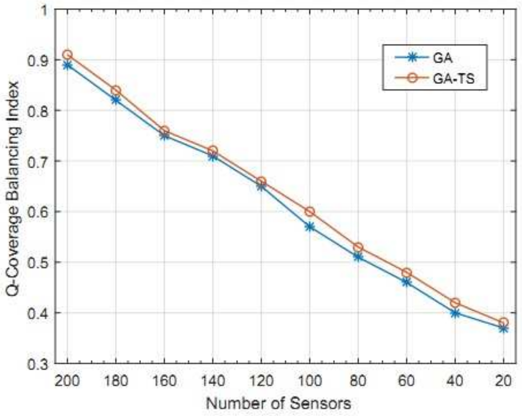


Fig. 4. Effect of increasing the number of sensors on the Q-coverage balancing index

4.3 Second Experiment

In this experiment, we examine how increasing the number of targets affects the DI (coverage performance index). As the number of targets rises, the coverage requirements also increase. Consequently, the distance between the required coverage vector and the achieved coverage vector grows. As illustrated in Figure 5, with more targets, the curve deviates further from the normal value (DI=1). A comparison of the proposed GA-TS algorithm with the standard GA shows that the GA-TS curve remains closer to 1, indicating better performance in this aspect.

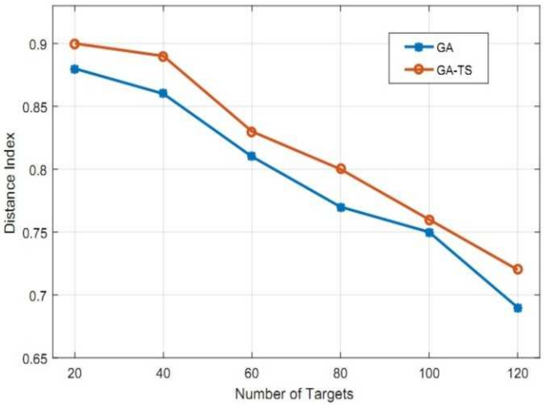


Fig. 5. Effect of increasing the number of targets on the Distance Index

4.4 Third Experiment
Impact of Increasing the Sensing Range on CQ

The third experiment aimed to evaluate the effect of changing the sensing range on the coverage quality (CQ). In this scenario, both the sensors and the targets are situated in a fixed environment. Increasing the sensing range of the sensors does not alter the distance between the targets and the sensors. However, a larger sensing range allows a single sensor to cover more targets. As demonstrated in Equation (5) and illustrated in Figure 6, the CQ value increases as the sensing range is increased.

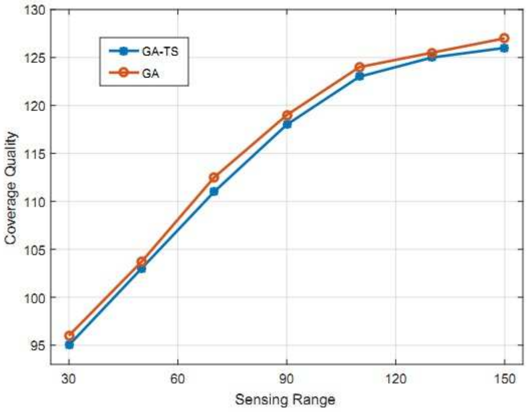


Fig. 6. Effect of increasing the sensing range on the coverage quality

5. Conclusion

Coverage in sensor networks is a key research area for many scholars. When the number of sensors in a network decreases, maintaining sufficient coverage becomes a significant challenge, especially considering the varying coverage requirements of different sensors. In such situations, one effective solution is to develop a method that balances the coverage needs of various targets, ensuring that these needs can be met with the available sensors. This research presents a method for achieving balanced coverage in the network, taking into account that each target may have different coverage requirements. The approach combines a genetic algorithm with a tabu search to optimize coverage.

References

[1] Ajam, Leila, Ali Nodehi, and Hosein Mohamadi. "A Genetic-based algorithm to solve priority-based target coverage problem in directional sensor networks." Journal of Applied Dynamic Systems and Control 4.1 (2021): 89-96.

[2] N. Mottaki, H. Motameni, H. Mohamadi, Multi-objective optimization for coverage aware sensor node scheduling in directional sensor networks, J. Appl. Dyn. Syst. Control (2021) 43–52.

[3] L. Ajam, A. Nodehi and H. Mohamadi, "A new approach to solving target coverage problem in wireless sensor networks using an effective hybrid genetic algorithm and tabu search", J. Intell. Fuzzy Syst., vol. 42, no. 6, pp. 6245-6255, 2022.

[4] Mottaki NA, Motameni H, Mohamadi H (2022) A genetic algorithm-based approach for solving the target Q-coverage problem in over and under provisioned directional sensor networks. Phys Commun 54:101719. https:// doi. org/ 10. 1016/j. phycom. 2022. 101719.

[5] Mottaki, Nemat allah, Homayun Motameni, and Hosein Mohamadi. "An effective hybrid genetic algorithm and tabu search for maximizing network lifetime using coverage sets scheduling in wireless sensor networks." The Journal of Supercomputing 79.3 (2023): 3277-3297.

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

[7] H. Mohamadi, et al., Heuristic methods to maximize network lifetime indirectional sensor networks with adjustable sensing ranges, 46 (2014)26–35.

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

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

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

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

[12] A. Javan Bakht, et al., A learning automata-based algorithm to solve imbalanced K-coverage in visual sensor networks, J. Intell. Fuzzy Systems39 (3) (2020) 2817–2829.

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

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

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

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

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

[18] S. Sivanandam, S. Deepa, Genetic algorithms, in: Introduction to Genetic Algorithms, Springer, 2008, pp. 15–37.

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

[20]D.E. Goldberg, Genetic Algorithms, Pearson Education India, 2006.

[21] Prajapati V, p et al Tabu Search Algorithm (TSA): A Comprehensive Survey. In:2020 3rd International Conference on emerging technologies in computer engineering: Machine Learning and Internet of Things (ICETCE), https:// doi. org/10. 1109/ICETC E48199. 2020. 091743.