# An improved genetic algorithm for bi-level multi-objective Q-coverage in directional sensor networks

Nguyen Thi Hanh[†§], Nguyen Van Son[*§], Huynh Thi Thanh Binh[*], Ban Ha Bang[*],
Trinh Van Chien [*¶], Huynh Cong Phap[‡], Nguyen Huu Nhat Minh[‡]
[*]School of Information and Communication Technology (SoICT), Hanoi University of Science and Technology, Vietnam;
[†]Phuong Dong University, Hanoi, Vietnam;
[‡]The University of Danang, Vietnam - Korea University of Information and Communication Technology.
[*]son.nv212303M@sis.hust.edu.vn, [†]hanhnt@dhpd.edu.vn, [*]binhht@soict.hust.edu.vn, [*]bangbh@soict.hust.edu.vn,
[*]chientv@soict.hust.edu.vn, [‡]hcphap@vku.udn.vn, [‡]nhnminh@vku.udn.vn

***Abstract*—Direction sensor networks are robust systems employed for detecting phenomena in environments or monitoring objects therein. They have a wide range of applications across many different industries and fields. In terms of the availability of resources, direction sensor networks deal with two problems: over-provision and under-provision of sensors. Over-provision occurs when there are too many sensors in the monitoring area, resulting in wasted resources and unnecessary energy consumption as some sensors are not well utilized. In contrast, under-provision occurs when there are too few sensors in the monitoring area, leading to the coverage of targets not satisfied. To ensure balanced coverage in under-provisioned environments, sensors must be placed so as to provide nearly equal fault tolerance to all objects, thereby enhancing the operational efficiency of the network. On the other hand, in over-provisioned environments, the number of active sensors needs to be minimized so that energy consumption is efficient. This study focuses on solving the Q-coverage problem in adjustable-orientation direction sensor networks, aiming to optimize a bi-level objective: maximizing network coverage balancing while minimizing sensor count in both under-provisioned and over-provisioned environments. The proposed Improved Genetic Algorithm utilizes novel operators, including Greedily-tuned Simulated Binary Crossover and Adaptive Polynomial Mutation. Evaluation parameters, including the Q-Balancing Index, Distance Index, Coverage Quality, Power Consumption, and the number of active sensors, demonstrate the efficiency of the proposed algorithm compared to other existing methods.**

***Index Terms*—bi-level multi-objective, directional sensor networks, genetic algorithm, Q-coverage, coverage balancing.**

## I. Introduction

In recent years, wireless sensor networks (WSNs) have become an active research area with many applications, especially with the rise of the Internet of Things (IoT) [20]. WSNs consist of energy-efficient, low-cost, small-sized sensor nodes that can monitor targets and communicate with each other to exchange and transmit data to the base station. WSN applications include military, medicine, environment, plants and animals, industry, and urban [12, 4]. That being said, WSNs still face challenges in coverage, connectivity, and energy efficiency.

In practice, some setups require certain types of sensors such as video, infrared and ultrasonic [3] to employ the directional sensing model. Directional sensors have limited sensing angle at any instant, unline omnidirectional sensors' perfect 360°. WSNs consisting of directional sensors are known as directional sensor networks (DSNs). In general, they can be divided into three types [13]: Aligned-orientation (ALODSN), Fixed-orientation (FIODSN), and Adjustable-orientation (ADODSN). The most flexible model and also the one used in this paper is ADODSN because it allows sensors' direction to be adjusted.

Target coverage problems ensure the quality of services in a network, i.e., monitor predetermined targets within the considered area. Among the target coverage problems, there are three degrees to which coverage constraint is required: simple coverage (1-coverage), k-coverage, and Q-coverage [10]. Each target must be covered by at least one sensor in simple coverage. k-coverage compels that there have to be at least k sensors monitoring each target. However, some applications need different targets to have privileged coverage priorities. To be more specific, the minimum number of nodes required to monitor each target is variable, which is called Q-coverage and is what we investigate in this article.

In previous studies, there are some limitations. First, most only consider k-coverage in DSNs [2, 11], not Q-coverage. Second, similarly, only Q-coverage in omnidirectional WSNs, not DSNs [23, 15]. Third, the considered network model is not very flexible, being either ALODSN or FIODSN. Fourth, the solutions for multi-coverage problems in DSNs are still raw, naive, and elementary [16]. Finally, existing methods were one-sided, only focused on deployment cost, overlooked network lifetime. For these reasons, our paper considers scheduling to optimize energy consumption and maintaining Q-coverage. Since Q-coverage optimization problems are NP-hard [21, 14], we propose a metaheuristic approach which offers significant improvements compared to other benchmarks. Our main contributions are summarized as follows:

- Formulating a bi-level multi-objective optimization problem that manages the directions of sensors in a random

[§]These authors contributed equally to this work.
[¶]Corresponding author.

deployment environment.

- Proposing an improved genetic algorithm (IGA) for both the under-provisioned and over-provisioned environments.
- Analyzing experimental results and compare the proposed algorithm with existing methods for DSNs utilizing various metrics.

The rest of the paper is organized as follows. Section II formulates the problem. Section III covers the related works. Section IV describes the proposed algorithms. Section V provides the experiment results. Finally, section VI concludes the research and gives insights into future works.

## II. Related Works

In [7], Chaudhary et al. proposed a Q-coverage scheme aiming to prolong the network lifetime and avoid complete network shutdown due to the failure of a single node. Additionally, implementation issues in WSNs in general and DSNs, in particular, have faced many challenges, especially the problem of coverage and/or network lifetime [6, 9, 19]. As such, there have been a number of attempts to prolong network lifetime under Q-coverage constraints in WSNs. Arivudainambi et al. [5] presented a memetic algorithm to improve network lifetime for single coverage, k-coverage, and Q-coverage. They divided the set of all the sensors into subsets in such a manner that each subset satisfies the coverage requirement, after that each subset of the sensor was activated subsequently. Similarly, in [15], Manju et al. also partitioned the sensors into subsets and applied an energy-based heuristic approach. The idea was to give priority to the sensors with maximum residual battery life and occupies maximum target coverage, avoiding redundant coverage of critical targets. In [17], Ozdag developed a new approach based on an Electromagnetism-Like (EM) algorithm which imitated the motion of the charged particles in an electromagnetic field to optimize Q-coverage using randomly distributed nodes in the determined ranges. Yarinezhad et al. [23] implemented the Particle Swarm Optimization (PSO) method in two enhanced forms: Cooperative PSO (CPSO) and fuzzy-logic-integrated cooperative PSO (CPSO-fuzzy). The idea of cooperative PSO was to employ $n$ swarms where each swarm is made up of $S$ one-dimensional particles instead of a swarm of $S$ particles for an $n$-dimensional problem.

As mentioned above, DSNs have attracted a great deal of research interest. In [22], Xu et al. proposed a Hierarchical Target-oriented Multi-Agent Coordination (HiT-MAC) framework to solve the maximum target coverage problem in DSNs. HiT-MAC broke down the coverage problem into two subtasks: targets assignment by a coordinator and tracking assigned targets by executors. The authors also introduced practical methods such as self-attention to effectively learn HiT-MAC using reinforcement learning. Akin to WSNs, one of the most common objectives regarding DSNs is prolonging the network lifetime. Peng et al. [18] considered the communication energy of sensor nodes in addition to sensing energy, which gave a reflection on how node energy was consumed in reality, unlike previous studies. They suggested a two-part approach made up of making different sensing direction sets satisfy the Q-coverage requirement in each round, followed by determining an optimal cluster headset and afterward creating multi-hop communication. In [11] and [2], the authors addressed the problem of k-coverage in DSNs. Javan Bakht et al. [11] presented two learning automata-based algorithms, both of which were carried out in a number of rounds, each of which outputted a cover set of sensor directions that satisfied k-coverage and minimizes energy consumption. Each sensor had a variable sensing range, whose magnitude was traded against the amount of battery drained. Alibeiki et al. [2] considered two types of environments: over-provisioned and under-provisioned and proposed a genetic-based algorithm to determine the cover sets of sensor directions, similar to [11].

In the end, there is still a lack of effort in solving the Q-coverage problems in ADODSNs to a large extent and the works concerning such problems faces many limitations as discussed above. Therefore, in this paper, we are interested in solving the Q-coverage problem in ADODSNs using an enhanced GA approach.

## III. Problem Formulation

### A. Sensor Model

This study considers the ADODSN sensor model. Sensors have a Field of View (FoV) range for target observation, representing a circular sector. In this sector, the sensor $s_i$ has center coordinates $(x_{s_i}, y_{s_i})$, sensing radius $R$, Angle of View (AoV) $\theta$, direction vector $\vec{f_i}$ being bisector of $\theta$. Denote $\vec{v}_{i,k}$ as a vector pointing from $s_i$ to target $t_k$. The direction vector $\vec{f_i}$ can rotate dynamically, providing flexible coverage compared to the FIODSN model. Figure 1 provides an example of an ADODSN sensor.

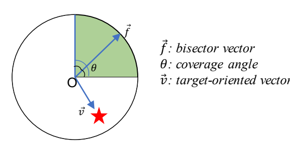



Fig. 1: An example of a directional sensor with regard to target coverage

Let function $\mathcal{C}$ indicate if sensor $s_i$ with direction vector $\vec{f_i}$ covers target $t_k$, then:

$$\mathcal{C}(s_i, f_i, t_k) = \begin{cases} 1 \text{ } if \text{ } \|\vec{v}_{i,k}\| \leq R \wedge \measuredangle(\vec{f_i}, \vec{v}_{i,k}) \leq \frac{\theta}{2}, \\ 0 \text{ } otherwise. \end{cases} \tag{1}$$

### B. Problem Formulation and Performance Metric

The Q-coverage problem is described as a set of $n$ sensors $S$ placed in the domain $A(W \times H)$ to cover $m$ predefined targets $T$. Each target $t_i \in T$ needs at least $q_i$ sensor covering it. Thus, we have $Q = \{q_1, q_2, \ldots, q_m\}$ as the coverage requirement set for all the targets in the Q-coverage problem. The proposed model is based on complete sensor homogeneity, meaning that all sensor nodes are equipped with the same specifications, particularly sensing range $R$ and the same AoV $\theta$. Let $F = \{\vec{f_1}, \vec{f_2}, \ldots, \vec{f_n}\}$ be the set of direction vectors of $S$.

To evaluate the efficiency of a solution, we use five metrics whose details are shown below:

- Q-Balancing Index ($QBI$) [16] which measures the degree of balance regarding network coverage based on the coverage requirement of the targets. A great value of $QBI$ indicates an equal share of fault tolerance between the targets. $QBI = 1$ means that Q-coverage is satisfied.
- Distance Index ($DI$) [1] which evaluates the quality of sensor coverage in the network. The greater $DI$ is, the higher satisfaction of Q-coverage for $T$ is.
- Coverage Quality ($CQ$) [1] which indicates the sensors' proximity to the targets. As sensors' signal decreases with distance, the closer the sensors are to the targets, the higher the quality of coverage is.
- Power Consumption ($PC$) and the number of active sensors ($\|S_A\|$) [1]. The two metrics are highly correlated because the fewer sensors are active, the less power is consumed.

The metrics can be computed as follows:

$$QBI = \frac{(\sum_{i=1}^{m} q_i^+)^3}{(\sum_{i=1}^{m} q_i)^3} \times \frac{\sum_{i=1}^{m} q_i^2}{\sum_{i=1}^{m} (q_i^+)^2} \tag{2}$$

$$DI = \frac{\sum_{i=1}^{m} q_i^2 - \sum_{i=1}^{m} (q_i - q_i^+)^2}{\sum_{i=1}^{m} q_i^2} \tag{3}$$

$$CQ = \sum_{i=1}^{n} \sum_{k=1}^{m} \mathcal{C}(s_i, f_i, t_k) \times \left(1 - \frac{\|\vec{v}_{i,k}\|}{R}\right) \tag{4}$$

$$PC = \|S_S\| \times p_S + \|S_A\| \times p_A + \|S_I\| \times p_I \tag{5}$$

where $q_i^+$ ($\leq q_i$) is the number of sensors that actually cover target $t_i$. The sets $S_A$, $S_I$ and $S_S$ consist of the sensors in active, idle, and sleep modes; $p_A$, $p_I$ and $p_S$ are the power consumption of each sensor by mode, also in that order, respectively. Specifically, $p_A = 5.268W$, $p_I = 1.473W$ and $p_S = 0.058W$ [1].

In an under-provision environment, Q-coverage is not satisfied, so the objective and primary metric is maximizing $QBI$ so that fault tolerance is spread equally between the targets ("good" fault tolerance). On the other hand, in an over-provision environment, $QBI = 1$, which means Q-coverage is (nearly) guaranteed, our aim is to minimize $\|S_A\|$.

Our problem is hereby referred to as the Sensor Direction Determining (SDD) problem, in which we need to determine the set of active sensors $S_A \subseteq S$ and their respective directions $\vec{f_i} \in F_A \subseteq F$ to optimize the following bi-level objective:

$$\begin{cases} \max_{s \in S_A; f \in F_A} \; QBI & \textit{in case of under-provision,} \\ \min_{s \in S_A; f \in F_A} \; \|S_A\| & \textit{in case of over-provision.} \end{cases} \tag{6}$$

**Subject to:**

$$S_A \subseteq S$$
$$\forall \vec{f_i} \in F, 0 \leq \measuredangle(\vec{f_i}, \vec{Oy}) \leq 2\pi$$

## IV. Proposed Algorithm

The Q-coverage problem in DSN refers to determining the active sensors and their directions to maximize Q-Balancing Index in under-provision environments (shortage of sensors) and minimize the number of active sensors in over-provision environments (excess of sensors). In this study, we propose an Improved Genetic Algorithm (**IGA**) to solve the Sensor Direction Determining (**SDD**) problem called **SDD-IGA**. SDD-IGA based on our new hypothesis that individuals with better fitness are more likely to undergo crossover, pass down their good genes for the children and have their probability and degree of mutation limited. Thus, SDD-IGA focuses mainly on improving the crossover and mutation operators and initializing the population for the best efficiency in solving the Sensor Direction Determining problem.

### A. Two-layer chromosome representation

To represent a feasible solution to SDD problem, we encode each individual in two layers. The first layer is a state array that indicates which sensors are active. It is denoted by $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ where $\alpha_i \in \{0, 1\}$ $\forall i \leq n$. The second layer is direction array which contain a set of real numbers representing the direction of each sensor $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ in which $\omega_i = \measuredangle(\vec{f_i}, \vec{Oy}) \in [0, 2\pi)$ $\forall i \leq n$. An illustration of the encoding-decoding of chromosome is shown in Figure 2.

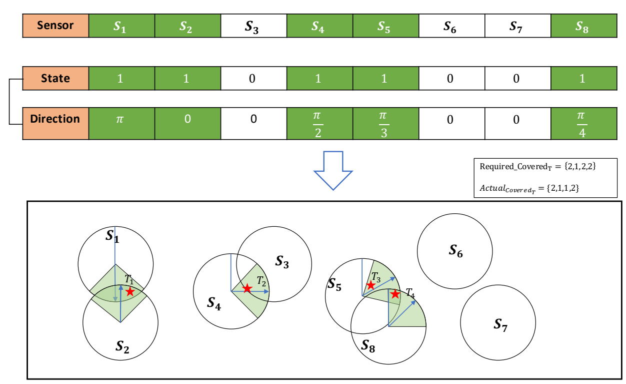



Fig. 2: An example of encoding-decoding a chromosome to a status of DSNs

### B. Population initialization

To initialize the population of $N$ individuals in SDD-IGA, there are two possible methods: Random Initialization and Heuristic Initialization. The Random Initialization assigns a random direction $\vec{f_i}$ to each sensor $s_i$, and randomly selects whether each sensor is active ($\alpha_i = 1$) or inactive ($\alpha_i = 0$).

The Heuristic Initialization method proceeds as follows: for each sensor $s_i \in S$, if $\forall t_k \in T$, $\|\vec{v}_{i,k}\| \geq R$ , the sensor is turned off ($\alpha_i = 0$) with default direction: $\omega_i = 0$. Otherwise, find the nearest target to $s_i$, denoted by $t_{nt}$, and randomly adjusts the direction $\vec{f_i}$ of the sensor to ensure coverage of $t_{nt}$ ($\alpha_i = 1 \wedge \measuredangle(\vec{f_i}, \vec{v}_{i,t_{nt}}) \leq \frac{\theta}{2}$). Figure 3 demonstrates how the heuristic method determines the angle $\omega_i$ of sensor $s_i$ for chromosome formation.

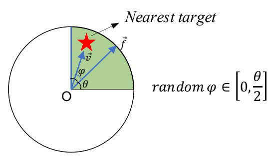



Fig. 3: An example of Heuristic Initialization for a sensor

Each individual in the population has an equal chance of being randomly or heuristically initialized. This approach aims to achieve a diverse population of individuals, which can aid in accelerating convergence and preventing local optima.

### C. Fitness evaluation

In our model, the status of the environment as over-provisioned or under-provisioned is not predetermined, which poses a challenge in defining the objective of the problem. However, it is established that $QBI \in [0, 1]$ and $QBI = 1$ in and only if the environment is over-provisioned. Consequently, we advocate a way to evaluate the fitness of a chromosome based on the objectives stated in (6), as well as determine if the environment is under-provisioned or over-provisioned, as follows:

$$Ft(c) = \Big[QBI(c), \|S_A(c)\|\Big] \tag{7}$$

where:

- $Ft(c)$ is the fitness of chromosome $c$
- $QBI(c)$ is the Q-Balancing Index value of $c$
- $\|S_A(c)\|$ is the number of active sensors of $c$ and can be calculated as $\sum_{i=1}^{m} \alpha_i$

Let $c_1$ and $c_2$ be two chromosomes, then $Ft(c_1) < Ft(c_2) \Leftrightarrow QBI(c_1) < QBI(c_2) \lor \Big(QBI(c_1) = QBI(c_2) \land \|S_A(c_1)\| > \|S_A(c_2)\|\Big)$.

### D. Crossover operation

A chromosome is composed of two parts: the *state array* $\alpha$ and the *direction array* $F$, which indicates the state and direction of each sensor. Hence, we use separate crossover operations for each part.

*1) Multi-point Crossover on state array:* We employ a multi-point crossover technique for two individuals on the state array of the chromosome. The intersection points and their indices are generated randomly and uniformly.

*2) Greedily-tuned SBX Crossover on direction array:* We present a novel real-number crossover technique for the direction array of the chromosome, which builds upon the SBX crossover method previously proposed by Deb et al. [8]. The idea is to iterate through each pair of directions: $\omega_i(p_1) \in \Omega(p_1)$ and $\omega_i(p_2) \in \Omega(p_2)$, allowing for automatic bias towards the better parent. Compared to the traditional SBX crossover, which treats the two parents equally, our approach have the children leaning towards the better parent to inherit its good genes. The basic formula of the SBX crossover in which two parents $p_1, p_2$ produce two offsprings $c_1, c_2$ is as follows:

$$c_1 = \frac{p_1 + p_2}{2} + \beta \times \frac{p_1 - p_2}{2}\ ;\ c_2 = \frac{p_1 + p_2}{2} - \beta \times \frac{p_1 - p_2}{2} \tag{8}$$

The SBX base $\beta^{base}$ is generated using the random variable $u \in [0, 1)$:

$$\beta^{base} = \begin{cases} (2 \times u)^{\frac{1}{n+1}}\ if\ u < 0.5, \\ (\frac{1}{2\times(1-u)})^{\frac{1}{n+1}}\ if\ u \geq 0.5 \end{cases} \tag{9}$$

To address the issue of the traditional SBX crossover treating the two parents as equal, this study introduces a tuning parameter denoted by $\Delta(\xi_i)$, which modifies the value $\beta_i^{base}$ of each sensor $s_i$ in offspring $c$, so that $c$ is inclined to the optimal direction angle $(\omega_i)$ of either $p_1$ or $p_2$, whichever is better, that results in a higher coverage count for the target set $T$.

First, we introduce the parameter $\xi_i$ for each sensor $s_i$ of offspring $c$ to estimate the coverage quality of $\omega_i$. The calculation of $\xi_i$ for each sensor $s_i$ with coordinates $(x_{s_i}, y_{s_i})$ involves:

- Determine the set of targets $T_i^+$ currently covered by $s_i$.
- Determine the set of targets $T_i^-$ currently not covered by $s_i$ while may possibly be covered by $s_i$ ($\|\vec{v}_{i,k}\| \leq R \land \measuredangle(\vec{f}_i, \vec{v}_{i,k}) > \frac{\theta}{2}$).

then we calculate the $\xi_i$ as below:

$$\xi_i = \|T_i^+\| + \sum_{t_k \in T_i^-} \frac{1}{1 + \measuredangle(\vec{f}_i, \vec{v}_{i,k})} \tag{10}$$

The illustration of estimating $\xi_i$ is shown in 4.

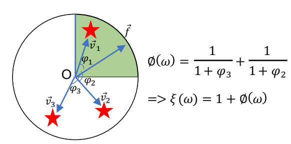



Fig. 4: An illustration of estimating $\xi_i$.

For each sensor $s_i$ in the chromosome, there exist two direction angles $\omega_i(p_1)$ and $\omega_i(p_2)$ of the parents $p_1, p_2$, We construct the coverage quality difference $\Delta(\xi_i)$:

$$\Delta(\xi_i) = \frac{\xi_i(p_1) - \xi_i(p_2)}{\max(\xi_i(p_1), \xi_i(p_2))} \tag{11}$$

We incorporate $\Delta(\xi)$ into $\beta^{base}$: $\beta = \beta^{base} - \Delta(\xi)$ to get a new $\beta$ value, which translates the genotypic region to an angle with improved coverage, as opposed to balancing equally between the two parents. Specifically, if $p_1$ has $\omega_i(p_1)$ that offers superior coverage compared to $\omega_i(p_2)$ of $p_2$ ($\Delta(\xi_i) > 0$), the random range of values for $\omega_i(c)$ will lean towards $\omega_i(p_1)$, and vice versa. A detailed simulation of the $\beta$ correction can be found in Figure 5.

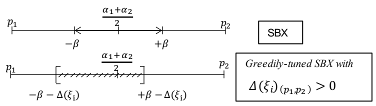



Fig. 5: A comparison of the traditional SBX crossover and the Greedily-tuned SBX Crossover.

### *E. Mutation operation*

*1) Bit Flip Mutation on state array:* The state array uses the conventional Bit Flip Mutation method. A greedy probability is assigned to each individual for mutation, and the mutation point is selected based on a uniform distribution. This approach ensures that sudden drastic changes are avoided, which may obscure the optimal solution to our problem. Additionally, it achieves high efficiency and performance due to its simple computation.

*2) Adaptive Polynomial Mutation on direction array:* Similarly to the crossover process on direction array, we leverage the "coverage quality" index of any angle $\omega$, represented by $\xi_i$, to automatically adjust the mutation domain of the Polynomial mutation operation.

The Polynomial mutation operation uses the random variable $u \in [0, 1)$:

$$\omega_i' = \begin{cases} \omega_i + \sigma_L(\omega_i - x_i^{(L)}) \ \textit{if}\ u \leq 0.5, \\ \omega_i + \sigma_R(x_i^{(U)} - \omega_i) \ \textit{if}\ u \geq 0.5 \end{cases} \tag{12}$$

where $\omega_i$ is the original value of sensor $s_i$ in chromosome $c$, $\omega_i'$ is the mutated value, $x_i^{(L)}, x_i^{(U)}$ is the lower bound and the upper bound of the parent values range, with $\eta_m \in [20, 100]$, $\sigma_L$, $\sigma_R$ is estimated as:

$$\begin{cases} \sigma_L = (2u)^{\frac{1}{1+\eta_m}} - 1, \\ \sigma_R = 1 - (2(1-u))^{\frac{1}{1+\eta_m}} \end{cases}$$

We introduce a parameter for adjusting the level of mutation for $\sigma_L$ and $\sigma_R$, called $\gamma_i$ for each angle $\omega$ at sensor position $s_i$. Specifically, the formula is as follows:

$$\gamma_i = 1 - \frac{\xi_i}{max(\{q_k \mid \|\vec{v}_{i,k}\| \leq R\})}$$
$$\sigma_L \coloneqq \sigma_L \times \gamma_i$$
$$\sigma_R \coloneqq \sigma_R \times \gamma_i$$

The above is the mutation process for each sensor $s_i$ in chromosomes $c$. To mutate overall $c$, we do this process for all sensor $s \in S$.

It is clear that $\gamma_i$ serves to constrain the mutation of direction when the direction angle $\omega_i$ is advantageous for sensor $s_i$ in terms of target coverage (i.e., when the value of $\gamma_i$ is small, it helps to decrease the value of $\sigma$). Conversely, when the direction angle $\omega_i$ results in poor coverage by sensor $s_i$, $\gamma_i$ will stimulate an increase in the value of $\sigma$, making the direction angle more variable.

At the chromosome selection phases, we use the combination of roulette wheel selection and elite selection.

## V. Experiments and Results

### *A. Problem instances*

We evaluate SDD-IGA with a dataset from [16] that includes 24 cases across three scenarios. Targets are placed randomly in the monitoring area using a uniform distribution. Scenarios 1 and 2 cover a $500 \times 500(m^2)$ monitoring area, while scenario 3 uses a larger $1000 \times 1000(m^2)$ area due to the larger sensing range. The dataset parameters are set to evaluate the algorithm under both under-provisioned and over-provisioned domains, ensuring a fair assessment of the impact of each parameter on the algorithm. Details for each scenario are provided below:

**Scenario 1**: 9 datasets generated with a fixed sensor radius of 100m and 130 targets, while the number of sensors varied from 20 to 180 in increments of 20.

**Scenario 2**: 8 datasets generated with 100 sensors, 100m radius, and varying number of targets from 10 to 150 in increments of 20.

**Scenario 3**: 7 datasets generated with 100 sensors, 130 targets, and sensor radius increasing from 60m to 120m in increments of 10m.

### *B. Algorithm parameters*

Parameter values for SDD-IGA were selected as follows: $\theta = \frac{\pi}{2}$;$q \in [1, 6]$ for coverage constraints; crossover rate of 0.8; mutation rate of 0.1; population size of 100; maximum generation of 1000; and the number of convergence iterations of 100.

### *C. Experiment settings*

All algorithms are implemented in Python and executed on a machine with Intel Core i5 2.4 GHz CPU, RAM 8GB 1600 MHz DDR3.

### *D. Experiment Results and Discussions*

In this section, we present the results of three experiments that evaluate the performance of **SDD-IGA** compared to Standard GA (SGA) and DSGA [16] adapted to the ADODSN model for a fair comparison. The experiments evaluate the effect of the number of targets (**scenario 1**), number of sensors (**scenario 2**), and sensing radius (**scenario 3**). Each algorithm is run 15 times on every problem instance, and five metrics ($QBI, DI, CQ, PC$, and $\|S_A\|$) are reported from these runs. In these experiments, we also compare the performance of the algorithms in both under-provisioned and over-provisioned domains to assess their robustness and generalization ability.

**Experiment 1:** We evaluate the effect of the number of sensors on the performance of the algorithms. The results in Table 6a show that in the under-provisioned environment, all five metrics increased with the number of sensors in all three algorithms, while in the over-provisioned environment, the values of these metrics converged and no longer increased. DSGA could not meet the Q-coverage requirements, resulting in a continued increase of $\|S_A\|$. On average, SDD-IGA required 101% of $\|S_A\|$ compared to DSGA but yielded better $QBI$ (262%), $DI$ (8%), and $CQ$ (14%). Moreover, SDD-IGA outperformed SGA on $QBI$, $DI$, and $CQ$ by $4 - 36\%$, while only using

$3\%$ more active sensors. Additionally, the values of $\|QBI\|$ and $DI$ for SDD-IGA were significantly lower than those of other algorithms.

**Experiment 2:** In this experiment, we mainly used the under-provisioned dataset to evaluate the impact of target numbers on $QBI$ and the Q-coverage ability $DI$. The results are shown in Figure 7. It is observed that both $QBI$ and $DI$ decrease almost linearly with the increasing number of targets. This can be easily proven since, with a fixed number of sensors and radius, the coverage ability of the sensors is limited, while the increasing number of targets raises the overall coverage demand of the network. The results also indicate that in the over-provisioned domain, the most important indices are $\|S_A\|$ and $PC$, in which the proposed SDD-IGA algorithm outperforms the other two algorithms: it improves by an average of $5\%$ compared to DSGA and $2\%$ compared to SGA. In the under-provisioned domain, SDD-IGA also performs better than both DSGA and SGA on important indices such as $QBI$ and $DI$. On average, SDD-IGA is better than $231\%$ and $6\%$ for $QBI$ and $DI$, respectively. Similarly, SDD-IGA is better than SGA by $8\%$ and $2\%$ for $QBI$ and $DI$, respectively. Moreover, SDD-IGA also produces better results than DSGA in terms of $\|S_A\|$ and $PC$, so it can be seen that SDD-IGA completely outperforms DSGA. On the other hand, SGA has better results than SDD-IGA by an average of $50\%$ in both $\|S_A\|$ and $|PC|$. Besides that, three algorithms are competitive in terms of $CQ$. However, in the under-provisioned domain, the objective of the problem is more focused on $QBI$, which shows that SDD-IGA can better utilize the number of sensors and efficiently activate sensors than SGA.

**Experiment 3:** In an under-provisioned environment (sensing range $R \leq 90m$), we primarily focused on comparing $QBI$ and $DI$ of the three algorithms. Similar to the previous experiments, SDD-IGA outperforms both SGA and DSGA on these metrics. Specifically, as shown in Figure 8a and 8b, SDD-IGA beats SGA by $25\%$ on $QBI$ and $18\%$ on $DI$. Additionally, SDD-IGA is significantly better than DSGA, with a $297\%$ improvement on $QBI$ values while maintaining the same value for $DI$. However, as seen in Figure 8c, DSGA outdoes both SGA and SDD-IGA on $CQ$. This suggests that while DSGA activates a large number of sensors, it does not meet the Q-coverage requirement.

In an over-provisioned environment, when $QBI \simeq 1$ and $DI \simeq 1$, we mainly focused on comparing $\|S_A\|$ and $PC$ of each algorithm. As shown in Figure 8d, those of SDD-IGA are better than both SGA and DSGA in over-provisioned scenarios.

Overall, these results demonstrate that our proposed algorithm, SDD-IGA, outperforms SGA and DSGA in under-provisioned and over-provisioned environments while utilizing only a single genetic algorithm instead of multiple algorithms like DSGA. However, in under-provisioned scenarios, SDD-IGA may sacrifice coverage quality $CQ$ in order to achieve a better balance of coverage $QBI$ and to favor other coverage objectives ($DI$). Despite this trade-off, SDD-IGA still offers superior performance on multiple comparative metrics regarding all experiment scenarios in both environments.

## VI. Conclusion

In this study, we address the Q-coverage problem in DSNs and propose an efficient solution called the Improved Genetic Algorithm with Sensor Direction Determining for both over-provision and under-provision environments. Our algorithm introduces a novel prioritization scheme for selecting individuals for crossover and mutation, along with two new operators: Greedily-tuned SBX Crossover and Adaptive Polynomial Mutation. These operators, combined with the prioritization scheme, generate high-quality solutions effectively by retaining good genes and enhancing diversity. We evaluate our method using various metrics such as Q-Balancing Index, Distance Index, Coverage Quality, Power Consumption, and the number of active sensors. The results show that our approach outperforms previous works and efficiently solves problems. In future work, we plan to expand our research to include simultaneous sensor placement and adjustment of the direction angle, as well as address real-time coverage where targets can move.

## Acknowledgment

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2022-TT-001 for Trinh Van Chien and Nguyen Van Son.

This research is funded by Ministry of Education and Training under project number B2023.DNA.13 for Huynh Cong Phap and Nguyen Huu Nhat Minh.

## References

[1] Abdullah Al Zishan et al. “Maximizing heterogeneous coverage in over and under-provisioned visual sensor networks”. In: *Journal of Network and Computer Applications* (2018).

[2] Abolghasem Alibeiki et al. “A new genetic-based approach for solving k-coverage problem in directional sensor networks”. In: *Journal of Parallel and Distributed Computing* (2021).

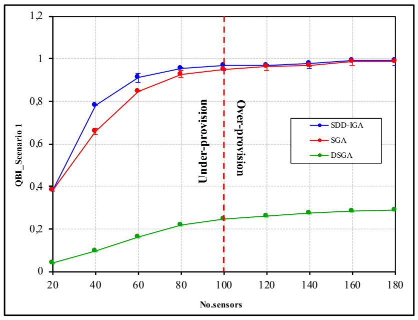
[3] M. Amac Guvensan et al. “On coverage issues in directional sensor networks: A survey”. In: *Ad Hoc Networks* (2011).

[4] J. Amutha et al. “WSN Strategies Based on Sensors, Deployment, Sensing Models, Coverage and Energy Efficiency: Review, Approaches and Open Issues”. In: *Wireless Personal Communications* (2020).

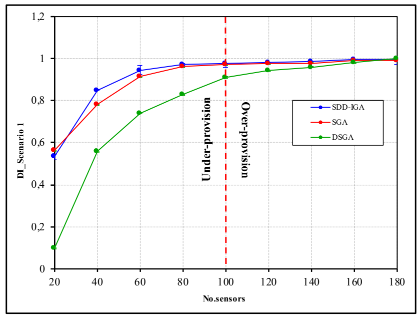
[5] D Arivudainambi et al. “Energy efficient sensor scheduling for Q-coverage problem”. In: *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2017.

[6] Azzedine Boukerche et al. “Connectivity and coverage based protocols for wireless sensor networks”. In: *Ad Hoc Networks* (2018).

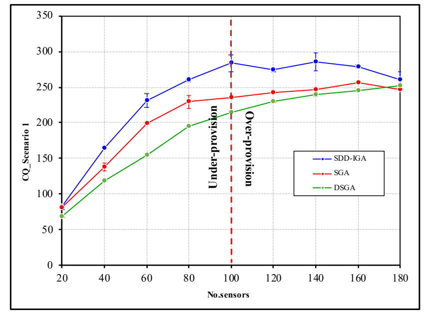
[7] Manju Chaudhary et al. “Q-coverage problem in wireless sensor networks”. In: *International Conference on Distributed Computing and Networking*. 2009.



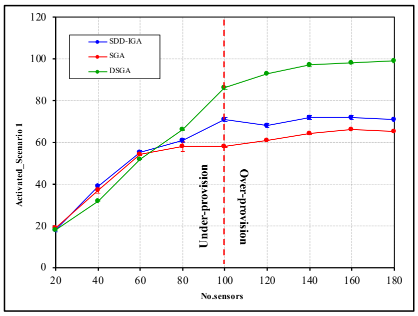

(a) The Q-Balancing Index values of **SDD-IGA**, SGA, and DSGA



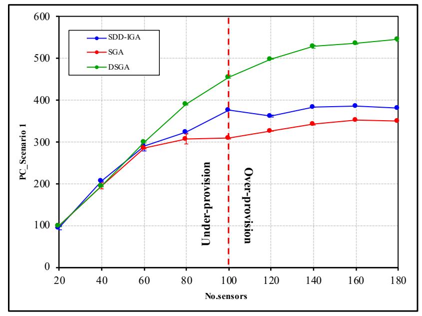

(b) The Distance Index values of **SDD-IGA**, SGA, and DSGA




(c) The Coverage Quality values of **SDD-IGA**, SGA, and DSGA

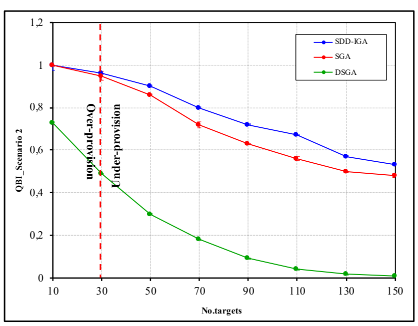



(d) The number of active sensor of **SDD-IGA**, SGA, and DSGA

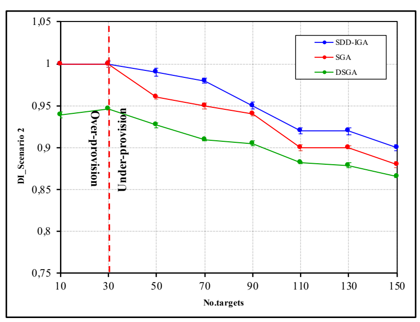



(e) The Power Consumption of **SDD-IGA**, SGA, and DSGA

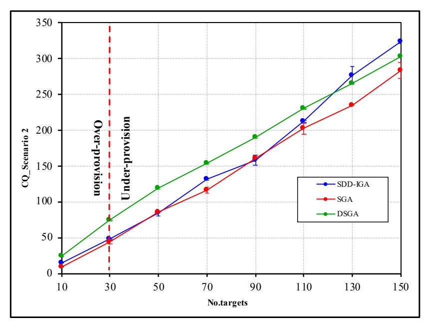
Fig. 6: The results over five metrics in Scenario 1 of **SDD-IGA**, SGA, and DSGA



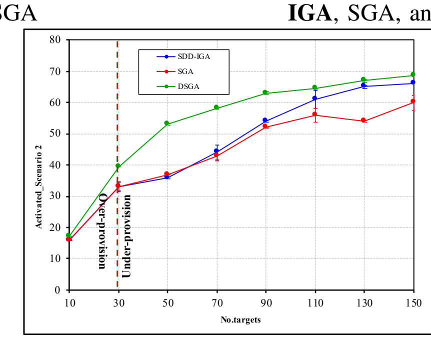

(a) The Q-Balancing Index values of **SDD-IGA**, SGA, and DSGA



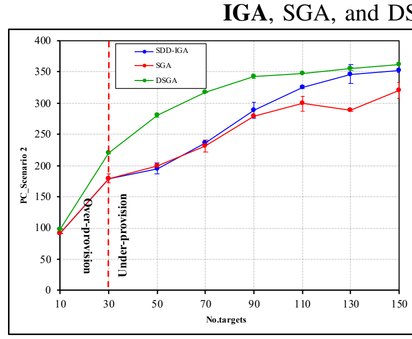

(b) The Distance Index values of **SDD-IGA**, SGA, and DSGA




(c) The Coverage Quality values of **SDD-IGA**, SGA, and DSGA

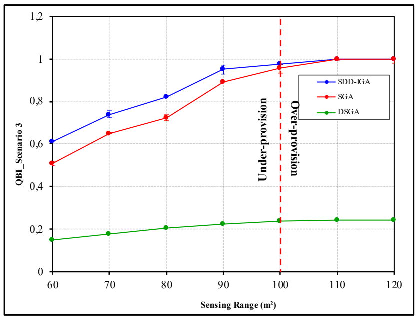



(d) The number of active sensor of **SDD-IGA**, SGA, and DSGA

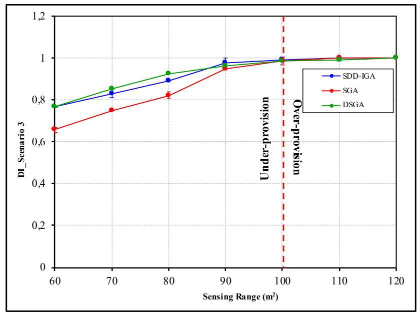



(e) The Power Consumption of **SDD-IGA**, SGA, and DSGA

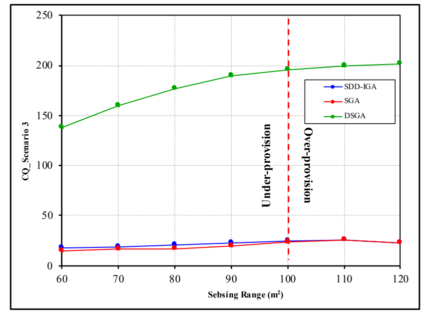
Fig. 7: The results over five metrics in Scenario 2 of **SDD-IGA**, SGA, and DSGA



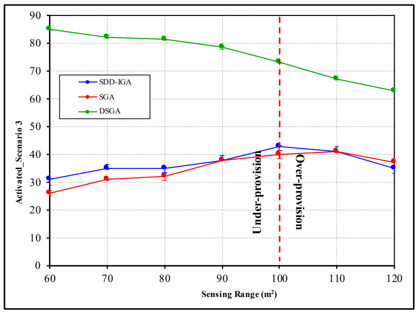

(a) The Q-Balancing Index values of **SDD-IGA**, SGA, and DSGA



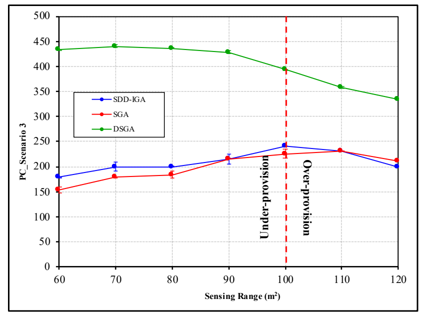

(b) The Distance Index values of **SDD-IGA**, SGA, and DSGA




(c) The Coverage Quality values of **SDD-IGA**, SGA, and DSGA




(d) The number of active sensor of **SDD-IGA**, SGA, and DSGA




(e) The Power Consumption of **SDD-IGA**, SGA, and DSGA

Fig. 8: The results over five metrics in Scenario 3 of **SDD-IGA**, SGA, and DSGA


[8] Kalyanmoy Deb et al. “Simulated Binary Crossover for Continuous Search Space”. In: *Complex Systems* (1995).

[9] Mohammed Farsi et al. “Deployment Techniques in Wireless Sensor Networks, Coverage and Connectivity: A Survey”. In: *IEEE Access* (2019).

[10] Nguyen Thi Hanh et al. “Node placement optimization under Q-Coverage and Q-Connectivity constraints in wireless sensor networks”. In: *Journal of Network and Computer Applications* (2023).

[11] Ahmad Javan Bakht et al. “A learning automata-based algorithm for solving the target k-coverage problem in directional sensor networks with adjustable sensing ranges”. In: *Physical Communication* (2020).

[12] Dionisis Kandris et al. “Applications of Wireless Sensor Networks: An Up-to-Date Survey”. In: *Applied System Innovation* (2020).

[13] Mohammad Khanjary et al. “Barrier coverage in adjustable-orientation directional sensor networks: A learning automata approach”. In: *Computers & Electrical Engineering* (2018).

[14] Hui Liu et al. “Energy-efficient algorithm for the target Q-coverage problem in wireless sensor networks”. In: *Wireless Algorithms, Systems, and Applications: 5th International Conference, WASA 2010, Beijing, China, August 15-17, 2010. Proceedings 5*. 2010.

[15] Manju et al. “Proficient QoS-Based Target Coverage Problem in Wireless Sensor Networks”. In: *IEEE Access* (2020).

[16] Nemat allah Mottaki et al. “A genetic algorithm-based approach for solving the target Q-coverage problem in over and under provisioned directional sensor networks”. In: *Physical Communication* (2022).

[17] Recep Ozdag. “Optimization of Target Q-Coverage Problem for QoS Requirement in Wireless Sensor Networks”. In: *Journal of Computers* (2018).

[18] Song Peng et al. “A Lifetime-Enhancing Method for Directional Sensor Networks with a New Hybrid Energy-Consumption Pattern in Q-coverage Scenarios”. In: *Energies* (2020).

[19] Rahul Priyadarshi et al. “Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues”. In: *The Journal of Supercomputing* (2020).

[20] Sumit Pundir et al. “Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges”. In: *IEEE Access* (2020).

[21] Bang Wang. *Coverage control in sensor networks*. 2010.

[22] Jing Xu et al. “Learning Multi-Agent Coordination for Enhancing Target Coverage in Directional Sensor Networks”. In: *Advances in Neural Information Processing Systems*. 2020.

[23] Ramin Yarinezhad et al. “A sensor deployment approach for target coverage problem in wireless sensor networks”. In: *Journal of Ambient Intelligence and Humanized Computing* (2020).