Journal of Network and Computer Applications 124 (2018) 44–62

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Maximizing heterogeneous coverage in over and under provisioned visual sensor networks

Abdullah Al Zishan, Imtiaz Karim, Sudipta Saha Shubha, Ashikur Rahman *

Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Bangladesh

ARTICLE INFO

Keywords:
Directional sensors
Heterogeneous coverage
Integer linear programming
Integer quadratic programming
Visual sensor network

ABSTRACT

We address "heterogeneous coverage" in visual sensor networks where coverage requirements of some randomly deployed targets vary from target to target. The main objective is to maximize the coverage of all the targets to achieve their respective coverage requirement by activating minimal sensors. The problem can be viewed as an interesting variation of the classical Max-Min problem (i.e., Maximum Coverage with Minimum Sensors (MCMS)). Therefore, we study the existing Integer Linear Programming (ILP) formulation for single and k-coverage MCMS problem in the state-of-the-art and modify them to solve the heterogeneous coverage problem. We also propose a novel Integer Quadratic Programming (IQP) formulation that minimizes the Euclidean distance between the achieved and the required coverage vectors. Both ILP and IQP give exact solution when the problem is solvable but as they are non-scalable due to their computational complexity, we devise a Sensor Oriented Greedy Algorithm (SOGA) that approximates the formulations. For under-provisioned networks where there exist insufficient number of sensors to meet the coverage requirements, we propose prioritized IQP and reduced-variance IQP formulations to capture prioritized and group wise balanced coverage respectively. Moreover, we develop greedy heuristics to tackle under provisioned networks. Extensive evaluations based on simulation illustrate the efficiency and efficacy of the proposed formulations and heuristics under various network settings. Additionally, we compare our methodologies and algorithm with two state-of-the-art algorithms available for target coverage and show that our methodologies and algorithm substantially outperform both the algorithms.

1. Introduction

A visual sensor network (VSN) consists of a large number of visual sensors having local image processing, communication, and storage capabilities that monitor a set of targets within an area of interest. The sensors–also known as *smart cameras*–are capable of self-controlling their orientation and range based on environmental conditions. Visual Sensor Networks have received appreciable attention of researchers due to their applicability in a wide number of significant real-life scenarios.

Visual sensors can be either omni-directional or directional. A sensor can provide coverage to a target if the target is within the sensing region of the sensor. Omni-directional visual sensors can provide coverage to all targets placed within its sensing region at the same time, whereas directional visual sensors can provide coverage only in a fixed direction at a time.

1.1. Deployment and application of visual sensor networks

A visual sensor network can be deployed in two ways:–(i) deterministic placement, and (ii) random scattering. In deterministic placement, the visual sensors can be suitably positioned to meet the coverage requirements. However, this is only possible in a small or medium-scale network where only a specified set of sensor locations is available and/or the topography is completely known. But in reality, deployment could be in large-scale containing thousands of sensors possibly in an inaccessible terrain (such as in battlefield) where random scattering is the most convenient and (perhaps) the only option.

The real world scenarios of the large-scale randomly deployed VSNs include surveillance system, target tracking, environment monitoring, traffic controlling, and battlefield monitoring, to name a few.

* Corresponding author.
E-mail addresses: 1205040.aaz@ugrad.cse.buet.ac.bd (A.A. Zishan), 1205098.ik@ugrad.cse.buet.ac.bd (I. Karim), 1205014.sss@ugrad.cse.buet.ac.bd (S.S. Shubha), ashikur@cse.buet.ac.bd (A. Rahman).

https://doi.org/10.1016/j.jnca.2018.09.009
Received 21 November 2017; Received in revised form 29 August 2018; Accepted 14 September 2018
Available online 21 September 2018
1084-8045/

1.2. Necessity of fault tolerance in visual sensor network

The basic form of Maximum Coverage with Minimum Sensor (MCMS) problem in a VSN deals with covering maximum targets using minimum sensors. Activating minimum possible sensors is necessary for building cost-effective and energy-efficient networks. However, in real environment, a target may lose its coverage due to various reasons such as power drainage of the sensors, malfunctioning sensors, sudden appearance of obstacle(s) along the covering pan of a sensor etc. So, besides coverage we also need to deliberately introduce *fault tolerance* by covering each target with more than one sensor.

1.3. Practicality of heterogeneous coverage in visual sensor network

While providing fault tolerance one might envision a homogeneous system where every target is covered equally. However, in reality we may not need the same degree of fault tolerance for all of the targets because all of them might not be of equal importance. For example, in an educational institution, there are various places such as classrooms, common rooms, laboratory rooms, office rooms, teachers' rooms, corridors etc. All places are not of equal importance. Therefore, while building a surveillance system for such an educational institution we may need different degree of coverage for different places. Perhaps one need to be more cautious with monitoring a laboratory room (which is mainly a private space installed with costly scientific equipments) than monitoring a corridor (which is more like a public place). So, we may want to assign more sensors for covering the laboratory room than for covering the corridor. Another important motivating scenario is deployment of visual sensors in battlefield monitoring. In a battlefield, critical targets like bastions or headquarters require greater number of visual sensors than comparatively less important targets like small barracks. On the contrary, assigning the same number of additional sensors for all the targets in such scenarios may introduce too many redundant sensors substantially increasing the network installation and maintenance cost. Thus, we end up with having heterogeneous coverage requirement problem of targets in such kind of scenarios.

1.4. Types of visual sensor networks

We are concerned about two kinds of visual sensor networks:–(i) Over-provisioned, and (ii) Under-provisioned. We call a system over-provisioned if we have enough sensors to fulfill the coverage requirements of the targets, whereas, a system is under-provisioned otherwise.

1.5. Practicality of under-provisioned network

We face the problem of scarcity of sensors in real environment. We need to focus on under-provisioned networks where there are insufficient number of sensors to ensure the heterogeneous coverage requirements or fault tolerance. Even a previously over-provisioned network may become under-provisioned in course of time due to discovery of some additional new targets but the number of sensors may remain the same. Moreover sensors are costly. Thus under-provisioned networks exist in real life. In that case we may assign coverage priorities; targets with higher coverage requirement should get higher coverage. Or we may maintain a balanced coverage; targets with the same coverage requirements would get more or less similar coverage.

All of these scenarios motivate us to investigate the heterogeneous coverage problem focusing on both over-provisioned and under-provisioned networks.

1.6. Previous works and our contributions

In this section, we present the related works that are aligned with our research. Also we point out the contributions of the paper that are novel with respect to the existing works.

1.6.1. Previous works

There are two main categories of research for single coverage problem (i.e., each target requires only one sensor to get covered) in *omni-directional* sensor setting. One thread of works deals with designing online algorithm according to some off-duty eligibility rule and other thread of works deals with designing offline algorithm. Under the first thread of work (designing online algorithm), Tian and Nicolas (Tian and Georganas, 2002) introduce the idea of "sponsored area" in designing an off-duty eligibility rule to ensure complete coverage. Analysing intersection points by sensors, Wang et al. (2003) design an off-duty eligibility rule. Zhang and Hou (2005) developed Optimal Geographical Density Control (OGDC) algorithm for minimizing sensing-overlap. Under the second thread of work, researchers design (offline) algorithms for organizing sensor nodes in power-aware fashion. Megrian and Potkonjak (Meguerdichian and Potkonjak, 2003) present ILP formulations and approaches to reduce energy consumption by sensor nodes while guaranteeing single coverage of all targets. Slijepcevic and Potkonjak (2001) propose set k-cover problem where they maximize k which is the number of disjoint set covers; here a set cover refers to a set of sensor nodes which can completely cover required area. The chosen sets will be active successively along time. Adding bandwidth constraint with disjoint set cover, Cheng et al. (2005) formulate minimum breach problem where sizes of set covers are bounded; they show that network lifetime can be extended by additional bandwidth constraint at the cost of coverage breach. Following disjoint set cover approach, Cardei et al. (2005) improved network lifetime by using the same node in multiple set covers. Zhao and Gurusamy (2008) consider the Connected Target Coverage (CTC) problem with the goal of maximizing network lifetime. The objective of their work is: scheduling the sensors in multiple sets each of which can both maintain the connectivity among the sensors and target coverage. Lu et al. (2015) study the Maximum Lifetime Coverage Scheduling (MLCS) problem for WSNs, considering both data collection and target coverage. In a survey Yetgin et al. (2017) present a comprehensive discussion on network lifetime optimization in WSN.

There exists a good number of research works on k-coverage (i.e., each target needs to be covered by at least k sensors) (Yen et al.; Ammari and Das, 2010; Hefeeda and Bagheri, 2007; Bejerano, 2008) using *omni-directional* sensors. Yen et al. divide the deployment area into circular sensing regions of fixed radius centered at each available sensors and perform k-coverage of those circular regions. Ammari and Das (2010) address the k-coverage problem of wireless sensor networks in three dimensional space. Hefeeda and Bagheri (2007) solve the k-coverage problem on dense networks. Bejerano (2008) works on k-coverage problem in situation where the location of targets and sensors is not known before. Notably, none of these works are directly applicable for directional visual sensors.

Existing works in *directional* sensor networks, can be broadly classified into several categories. In one category, the coverage requirement is homogeneous; each target requires to be covered by the same number of sensors. Under this category, Ai and Abouzeid (2006) formulate the single coverage requirement (i.e., every target needs to be covered by at least one sensor) as Maximum Coverage with Minimum Sensors (MCMS) problem and devise the exact integer linear programming (ILP) solution. They also provide greedy heuristics to approximate the optimal formulation. Lu et al. (2014) study Maximum Directional Target Coverage Problem (MDTCP). They mathematically formulate the problem as a Mixed Integer Linear Programming (MILP) and propose an approximation algorithm. Cai et al. (2009) approach single coverage problem in target-oriented way. They organize sensors in cover sets and activate only one cover set at a time to increase network lifetime. Zannat et al. (Zannat et al., 2016) study the single coverage problem from target oriented approach. They provide greedy algorithm that prioritize the targets that are less coverable. They also provide approximation bounds on existing and proposed heuristics. Our work differs from Zannat et al. (2016) in many aspects: they are concerned with single coverage whereas we deal with heterogeneous coverage, i.e., coverage requirements of targets differ from target to target. They do not formulate mathematical formulation, whereas we have precise ILP and IQP formulations for solving heterogeneous coverage problem. Their greedy algorithm is target-oriented, whereas ours is sensor-oriented. Fusco and Gupta (2009) study the k-coverage with minimum sensors problem where each target should be covered by at least k sensors. They present greedy algorithm to tackle the k-coverage problem. Malek et al. (2016) discuss the coverage imbalance in k-coverage problem. Especially in under-provisioned networks where there are insufficient number of sensors to provide k-coverage, their work provides balanced coverage. Our work differs from Malek et al. (2016) in various aspects: their work deals with k-coverage problem, i.e., each target needs to be covered by at least k sensors, whereas our work deals with heterogeneous coverage problem, i.e., coverage requirement of targets differ from target to target. Their work focuses on coverage imbalance problem in under-provisioned network, whereas, our work deals with maximizing heterogeneous coverage in both over-provisioned and under-provisioned network. Another stream of works (Wang et al., 2009; Yang et al., 2010; Salleh et al., 2014; Razali et al., 2017; Sharmin et al., 2016, 2017) deals with priority based target coverage in directional sensor networks. In these works, each target has pre-assigned priority in the range from 0 to 1, the higher this value is, the higher the target is assigned priority. Wang et al. (2009) provide genetic algorithm for addressing priority based target coverage in directional sensor networks. Yang et al. (2010) use the idea of cover set to address priority based target coverage problem: a cover set can provide required priority for all targets and one cover set will be activated at a time. They focus on maximizing network lifetime in their work. Salleh et al. (2014) provide a new learning automata based algorithm for priority based target coverage problem in directional sensor networks. Razali et al. (2017) provide scheduling algorithms for addressing priority based target coverage problem in directional sensor network with adjustable sensing ranges. Sharmin et al. (2016) devise scheduling algorithm to maximize coverage quality of targets using minimum number of sensors. Sharmin et al. (2017) deal with tradeoff between maximizing coverage quality of the targets and maximizing network lifetime. These studies do not impose the necessity for a specific number of sensors for each target. These studies use "Coverage Quality Function" and exact number of sensors covering a target may vary for different coverage quality functions. In our work the *fault tolerance* aspect is more strongly addressed as we require specific number of cameras to cover each target. In another category of works, coverage requirements are heterogeneous, i.e., each target requires different coverages. In literature this is well known as target Q-coverage (TQC) problem where, $Q = (q_1, q_2, q_3, \dots, q_m)$ is a vector and q_i is the required coverage of ith target. Gu et al. (2009) discuss TQC problem with maximization of network lifetime for omni-directional sensors. Their formulation chooses a set of sensors that satisfy the requirements and optimize the network lifetime. Chaudhary and Pujari (2009) discuss the TQC problem with maximization of network lifetime. However, in both (Gu et al., 2009; Chaudhary and Pujari, 2009), sensors are omni-directional. Li et al. (2012) modify the TQC problem for directional sensors with the goal of maximizing network lifetime. They find a collection of SDQ-covset (a set of sectors that satisfy the requirements) that maximize the network lifetime with a given bound on service delay. There is another thread of works related to heterogeneous coverage, called differentiated coverage, where required coverage degree for an area vary from area to area. Yan et al. (2003) address a differentiated surveillance problem. They provide a scheduling protocol that determine the time slots for sensors to give the required coverage for the grid points. Du and Lin (2005) provide another scheduling algorithm for differentiated coverage with heterogeneous sensors. These works basically deal with designing scheduling algorithms.

1.6.2. Our contributions

Although our work is aligned with Q-coverage and differentiated coverage to some extent, there are significant differences. The major differences with Q-coverage problem are as follows. Firstly, we mathematically formulate the problem as a variant of Maximum Coverage with Minimum Sensors (MCMS) problem whereas in Q-coverage the problem is formulated focusing on network lifetime. Secondly, formulation in Q-coverage assumes all coverage patterns are already generated and it chooses the optimal one whereas, our formulations generate such patterns. Thirdly, in Q-coverage, due to large number of coverage patterns, methods like column generation are followed (Gu et al., 2009) whereas, we directly solve our formulations. With respect to differentiated coverage, the differences are as follows. Firstly, in differentiated coverage, there is no mathematical formulation whereas we provide precise formulation of heterogeneous coverage as a variant of Maximum Coverage with Minimum Sensors (MCMS) problem. Secondly, works on differentiated coverage basically deal with protocol design whereas, our work finds a sensor orientation using ILP, IQP and greedy heuristics. *Moreover*, to the best of our knowledge, no works either on Q-coverage or differentiated coverage provide special emphasis on under-provisioned network which is a quite reasonable scenario in real life. In under provisioned network, as there is insufficient number of sensors to attain coverage requirements of the targets, we need to provide *best-effort* to maximize the coverage of each target. For under-provisioned networks, we propose priority IQP and reduced-variance IQP formulations to capture prioritized and group wise balanced coverage respectively.

The major *contributions* of the paper are summarized below:

(i) We study the heterogeneous coverage as a variant of Maximum Coverage with Minimum Sensors (MCMS) problem where coverage of each target should be maximized with minimum number of sensors for both over-provisioned and under-provisioned networks. We propose ILP and IQP formulations for heterogeneous coverage. Especially for under-provisioned networks, we propose two constraints: prioritized and group-wise balanced coverage to achieve the coverage maximization goal.

(ii) We derive an upper bound on the penalty coefficient of ILP formulation which can be easily generalized for other formulations.

(iii) We provide prioritized Integer Quadratic Programming (pIQP) and reduced-variance IQP formulations for under-provisioned networks.

(vi) Also, we devise a greedy algorithm SOGA (Sensor Oriented Greedy Algorithm) that gives near optimal solutions for the above formulations.

(v) Additionally, we compare our methodologies and algorithm with two state-of-the-art algorithms available for target coverage and show that our methodologies and algorithm substantially outperform both the algorithms.

1.6.3. Organization of this paper

The paper is organized as follows. This section summarizes the related works along the same research direction. Here we discuss about the existing works that deal with either omni-directional or directional sensors. In Section 2, we present the network model with relevant parameters. Also we develop our problem statement with explanations. Section 3 represents an Integer Linear Programming (ILP) approach (and its explanation) to solve our problem. In Section 4 we develop an Integer Quadratic Programming (IQP) approach to solve our problem. Section 5 discusses important modifications in our basic problem statement for tackling under-provisioned network. Here we devise prioritized IQP and reduced-variance IQP formulation to incorporate prioritized coverage and group-wise balanced coverage respectively. In Section 6, we represent a Sensor Oriented Greedy Algorithm (SOGA) which is a heuristic to approximate the formulations. Here we use different *benefit* functions to follow different formulations. Section 7 depicts different simulation results for various network settings. Here

Table 1
A concise list of symbols.

Symbol	Meaning
m	number of targets
n	number of sensors
q	number of non-overlapping pans of a sensor
g_t	the tth target
s_i	the ith sensor
k_t	coverage requirement of target g_t
ψ_t	achieved coverage of target g_t; maximum value is the requirement, k_t
α_t	actual coverage of target g_t; may exceed the requirement, k_t
$\mathcal{T}_{ij}$	a subset of targets, that are coverable by sensor s_i in pan p_j

we compare the approaches and heuristics under some well-defined metrics. Finally in Section 8, we draw our conclusions with a summary of our important findings.

2. Preliminaries

In this section, we define our visual sensor network model with different parameters. Then we describe a testing method through which the location of the targets can be determined. Finally, we define our problem with a brief explanation. In Table 1 we present a concise list of symbols used in the paper. Note that, for the first time reading, we may skip this table without any trouble. The symbols are defined later, in the relevant sections. This table is just for easy reference (Table 1).

2.1. Network model

A visual sensor network (VSN) consists of smart sensors (or, cameras) which have some local image processing, communication and storage capabilities. Unlike omni-directional sensors, these cameras are directional in their field of view (FoV). FoV is the span of observable area at any given direction by the camera. In VSN, sensors are usually *Pan-Tilt-Zoom* (PTZ) cameras, where the FoV is adjustable in (i) horizontal direction (or pan), (ii) vertical direction (or tilt), and (iii) depth-of-field (or zoom).

In our model, we assume a camera can move only in the horizontal direction. Also its FoV is defined by its pan. We present the sensing area or pan of a camera as a sector in a circle lying on a 2-dimensional plane as shown in Fig. 1.

To formalize the different aspects of a camera node, we introduce the following parameters (please refer to Fig. 1):

- (x_i, y_i): the location of the camera s_i in Cartesian coordinate system.
- θ: Angle of View (AoV) or maximum sensing/coverage angle of a camera in any direction.
- R_s: maximum *coverage range* of a camera beyond which it cannot detect any target.
- $\overrightarrow{d_{ij}}$: a unit vector which passes through the middle of a pan, representing the orientation of camera s_i towards pan p_j.
- $\overrightarrow{v_{it}}$: a vector in the direction from camera s_i to target g_t.

In our model we assume that cameras are homogeneous; each camera has similar parameter values. Also we assume that each camera has finite orientations or pans and the pans are disjoint. For example, in Fig. 1, a camera with $\frac{\pi}{4}$ of FoV can choose one of the eight pans which are mutually non-overlapping. The combination of all the pans generate full circular view of a camera's entire sensing region.

Our proposed approaches are *Centralized*. A sensor transmits data (such as, which targets are in which pan) to its neighboring nodes, they transmit to their neighboring nodes and in this way, the data ultimately goes to the sink (or, central) node. After getting data from all the sensors, algorithm for orientating the sensors is run in the sink node and the results are sent back to all the sensors in the similar fashion.

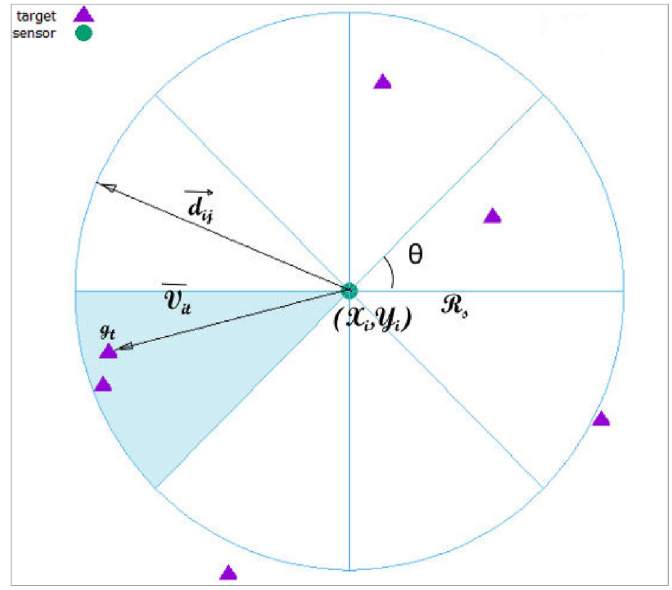

Fig. 1. A directional sensor and its coverage parameters.

2.2. Target in which pan (TIWP) test

It is a testing scheme, through which we can verify whether a target g_t is coverable by a given sensor s_i or not. Also we can find the corresponding pan in which the target belongs to. This test is similar to TIS test described in Ai and Abouzeid (2006). The TIWP test for a sensor s_i and target g_t can be performed as follows:

- First we calculate the angle ϕ_{it} between camera orientation $\overrightarrow{d_{ij}}$ of pan p_j and the target vector $\overrightarrow{v_{it}}$,

$$\phi_{it} = \cos^{-1}\left(\frac{\overrightarrow{v_{it}}.\overrightarrow{d_{ij}}}{|\overrightarrow{v_{it}}||\overrightarrow{d_{ij}}|}\right) \tag{1}$$

- Then we check whether the target vector $\vec{v}_{it}$ falls within the FoV of the camera s_i by checking the constraint, $\phi_{it} \le \frac{\theta}{2}$; another way to perform this checking is to use the inner product,

$$\overrightarrow{v_{it}}.\overrightarrow{d_{ij}} \ge |\overrightarrow{v_{it}}| \cos\left(\frac{\theta}{2}\right) \tag{2}$$

- Finally, we verify whether the target g_t is within the sensing range of the camera by checking, $|\overrightarrow{v_{it}}| \le R_s$.

If a target passes these steps, the result of the TIWP test is true, i.e, the camera s_i covers the target g_t in pan p_j.

By performing TIWP tests on every pan p_j of camera s_i and every target g_t, we can build a subset of targets $\mathcal{T}_{ij}$ for each sensor s_i which contains the targets that are coverable by sensor s_i in pan p_j.

2.3. Problem formulation

We formally define the Heterogeneous Coverage of Targets (HCT) problem as follows:

Given:

- a set of targets, $\mathcal{G} = \{g_1, g_2, g_3, . . ., g_m\}$ to be covered
- a tuple of positive integers, $\mathcal{K} = (k_1, k_2, k_3, . . ., k_m)$, where k_t is the required coverage of target g_t for all $g_t \in \mathcal{G}$; two targets g_i and g_j are in the same coverage group if they have same coverage requirements, i.e., $k_i = k_j$
- a set of homogeneous cameras (or directional sensors), $\mathcal{S} = \{s_1, s_2, s_3, . . ., s_n\}$, each of which can be oriented in one of q possible disjoint pans

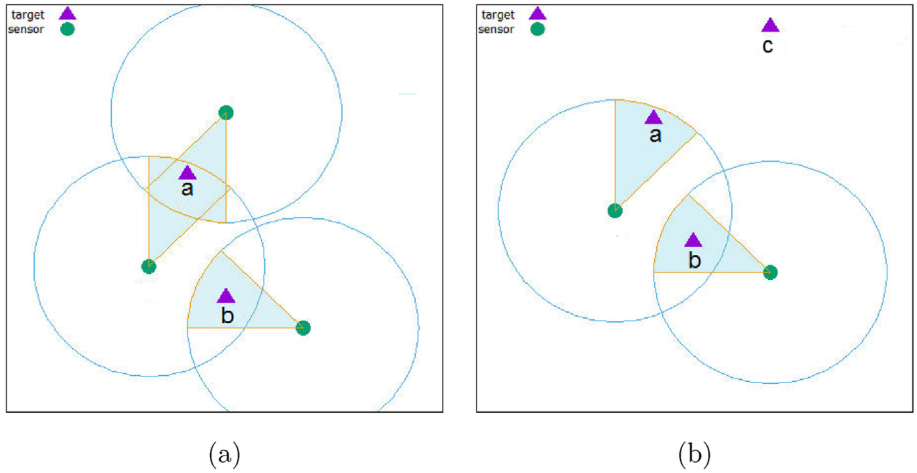

Fig. 2. A simple illustration of solvable and unsolvable scenario. (a) A solvable scenario; the targets (a, b) have the requirements of $(2, 1)$ respectively. (b) An unsolvable scenario; the targets (a, b, c) have the requirements of $(2, 1, 1)$ respectively.

- a set of disjoint (or non-overlapping) pans, $\mathcal{P} = \{p_1, p_2, p_3, \ldots, p_q\}$
- a set of all possible (*sensor*, *orientation*) pairs is defined as, $\mathcal{F} = \{(s_i, p_j) | \textit{sensor } s_i \textit{ is activated on pan } p_j\}$.

Problem: Find a subset $\boldsymbol{C}$ of $\mathcal{F}$ such that, there is at most one pair for a sensor in $\boldsymbol{C}$ and achieved coverage of each target g_t gets maximized under the condition that target g_t is covered by at least k_t sensors and the total number of active sensors gets minimized.

2.4. Discussion on problem statement

The single coverage (or MCMS) problem which is a simplified version of HCT is NP-hard (Ai and Abouzeid, 2006). The optimal solution of k-coverage problem is NP-hard (Fusco and Gupta, 2009) too. Naturally, the HCT problem is (at least) as hard as k-coverage problem.

We say a problem instance is *solvable* if at least one set of (*sensor*, *orientation*) pairs exists to achieve all the coverage requirements. The previously introduced two kinds of sensor systems, namely the *over-provisioned*, and the *under-provisioned* systems can be redefined in terms of solvability. We call a system over-provisioned if the problem instance is usually solvable i.e., we assume to have enough sensors to fulfill the coverage requirements of the targets, whereas, in under-provisioned systems there do not exist enough sensors to meet the coverage requirements of the targets, i.e., the problem instance is *unsolvable*.

For example, consider the Fig. 2 (a). Here we have three sensors and two targets, (a, b) with coverage requirements $(2, 1)$ respectively. The problem is solvable and one of the possible solutions is presented by the shaded pans of the sensors.

On the other hand, the scenario of Fig. 2 (b), represents an unsolvable problem. Here, targets (a, b, c) have the requirements of $(2, 1, 1)$ respectively. Clearly, the target c can not be covered by any sensor but its requirement is 1.

If coverage requirements can not be met, we aim at achieving one of the following two possibilities:– (i) achieve *prioritized* coverage, or (ii) achieve *groupwise* balanced coverage.

In prioritized coverage, one can set priority to targets; the coverage of higher priority targets should be maximized although for other targets, requirements may not be fulfilled. Usually, higher coverage requirement of a target implicitly indicates its higher importance. On the other hand, in group-wise balanced coverage, one can activate a set of sensors that minimizes variances of achieved coverage within each coverage group. The intuition behind this approach is, the targets with the same coverage requirement should be covered by the same number of active sensors. Thus, within a coverage group achieved coverage should have *zero* variance. In the following sections we present all different approaches to solve the HCT problem.

3. Linear programming approach to solve HCT problem

The motivation behind using linear programming method to solve HCT problem is to maximize the total achieved coverages of all the targets and minimize the number of active sensors. This optimization task can be captured by an Integer Linear Programming (ILP) formulation. An ILP formulation to solve Maximum Coverage with Minimum Sensors (MCMS) problem is explained in Ai and Abouzeid (2006). They formulate it to solve the single coverage requirement problem. Malek et al. (2016) extend the ILP formulation for k-coverage requirement problem.

In this section we explain an ILP formulation to solve the HCT problem. Then we discuss some important aspects of this formulation.

3.1. ILP formulation

The parameters that we use for ILP formulation are:

- n: number of sensors
- m: number of targets
- q: number of non-overlapping pans of a sensor
- α_t: an integer variable that counts *actual coverage* of target g_t by the sensors; the value of this variable may exceed the coverage requirement k_t
- ψ_t: an integer variable that denotes *achieved coverage* of target g_t by the sensors; the maximum value of this variable is bounded by k_t (i.e. there is no extra benefit for more coverage than the requirement)
- $\chi_{(i,j)}$: a binary variable with value one when the sensor s_i is in orientation p_j and zero otherwise
- *Iverson bracket*: let P be a proposition, either true or false; then *Iverson bracket* is defined as:

$$[P] = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Therefore, α_t can be expressed as:

$$\alpha_t = \sum_{i=1}^{n} \sum_{j=1}^{q} [g_t \in \mathcal{T}_{ij}] \chi_{(i,j)} \tag{4}$$

Now, the ILP formulation for HCT problem becomes,

$$\textbf{maximize} : \sum_{t=1}^{m} \psi_t - \rho \sum_{i=1}^{n} \sum_{j=1}^{q} \chi_{(i,j)} \tag{5}$$

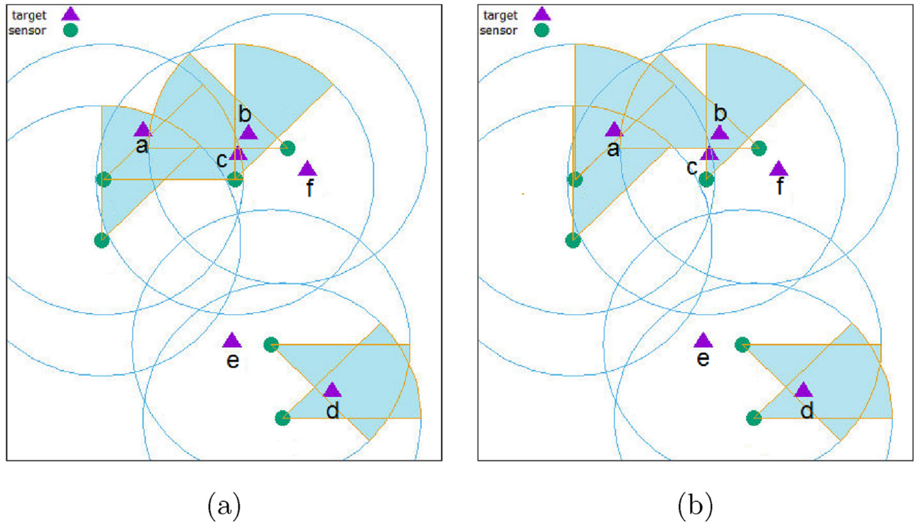

Fig. 3. Solutions produced by ILP and IQP for a scenario with 6 sensors and 6 targets. (a) Selection of pans by ILP (b) Selection of pans by IQP. A detailed analysis is presented in Table 2.

subject to constraints:

$$\frac{\alpha_t}{n} \leq \psi_t \leq \alpha_t, \quad \forall t = 1, 2, \ldots, m \tag{6}$$

$$\psi_t = 0 \text{ or } 1 \text{ or } 2 \text{ or } \ldots \text{ or } k_t, \quad \forall t = 1, 2, \ldots, m \tag{7}$$

$$\sum_{j=1}^{q} \chi_{(i,j)} \leq 1, \quad \forall i = 1, 2, \ldots, n \tag{8}$$

$$\chi_{(i,j)} = 0 \text{ or } 1, \quad \forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, q \tag{9}$$

3.2. Explanation

The objective function in Eqn. (5) does two things: maximize the total coverage and minimize the number of active sensors. Here the number of active sensors is multiplied by a penalty coefficient $\rho\,(<1)$.

Eqn. (6) indicates if target g_t gets covered or remains uncovered. If a target g_t gets no coverage by any sensor, then α_t is zero and also ψ_t is zero which conforms to the right inequality. If target g_t gets covered by one sensor (at least), then $\alpha_t > 0$. As α_t cannot exceed n, the fraction, $\frac{\alpha_t}{n}$ (a real number) becomes less than one and $\psi_t \geq 1$ which follows the left inequality.

Eqn. (7) presents that the *achieved coverage*, ψ_t of target g_t is bounded by k_t. That is if target g_t gets covered by more than k_t times, we do not provide extra benefit for extra coverage exceeding k_t. Note that, there is a subtle difference between ψ_t and α_t. α_t denotes the *actual coverage*; it may exceed the coverage requirement of target g_t. For example, let us consider two targets with coverage requirements $(2, 1)$ respectively. Now, ILP may choose a solution with $(3, 2)$ *actual coverages*. But there is no benefit to give more coverage than the requirement. So, in this case, $(\psi_1, \psi_2) = (2, 1)$ but, $(\alpha_1, \alpha_2) = (3, 2)$ respectively.

Eqn. (8) represents that each directional sensor will be in a single orientation at a time, if it is active.

3.3. Properties of ILP

One important aspect of ILP is that, it gives exact solution (or, achieves the coverage requirement of each target) when the problem is solvable. This characteristic of ILP can be proved easily from the constraint in Eqn. (7). Here ψ_t has maximum value k_t for each target g_t. If the equality is possible then ILP will choose it to maximize the objective function of Eqn. (5).

When the problem is not solvable, ILP gives an upper bound on the total achieved coverage as ILP maximizes the total coverage according to Eqn. (5). In summary, when the problem is solvable, ILP will maximize the coverage of each target and find the exact solution. But if the problem is not solvable, there is no guarantee that ILP will maximize the coverage of each target. Instead it will provide maximum total coverage.

For example, consider the scenario of Fig. 3(a). Here the six targets a, b, c, d, e, f have the coverage requirements of $(3, 3, 2, 2, 1, 1)$ respectively, ILP gives a solution of achieved coverage $(1, 2, 2, 2, 0, 0)$ respectively with total coverage of 7 and all the sensors are active. Note that, no orientation of the sensors provide required coverage for all the targets and as a result, the problem instance is not solvable. A detailed analysis is given in Table 2.

3.4. Impact of penalty coefficient

The penalty coefficient ρ is a positive real number. It is used in the ILP formulation to impose a penalty on the number of activated sensors. If there are two solutions, one of them has higher total coverage than the other, this penalty coefficient ensures to choose the former one, irrespective of the number of activated sensors. To ensure this, ρ must be smaller than some number. Lemma 3.1 provides the upper bound of the penalty coefficient ρ.

Lemma 3.1. *Let n be the number of sensors. Then for Integer Linear Programming (ILP) formulation, the penalty coefficient, ρ should be smaller than $\frac{1}{n}$ to pick a solution with higher coverage count, irrespective of activated sensor count.*

Table 2
A detailed analysis of Fig. 3

Target ID	Requirement	Maximum possible coverage	Coverage by ILP	Coverage by IQP
a	3	3	1	2
b	3	2	2	2
c	2	3	2	1
d	2	2	2	2
e	1	2	0	0
f	1	2	0	0

Proof. Suppose, we have two solutions; in the first solution total coverage and total activated sensors are a_1 and b_1 respectively and for the second solution these are a_2 and b_2 respectively. So the ILP objective functions for these solutions are $a_1 - \rho b_1$ and $a_2 - \rho b_2$ respectively. Now if $a_2 > a_1$ (and also $b_2 \geq b_1$ because higher coverage requires higher activated sensors), then to choose the second solution uniquely we have,

$$a_2 - \rho b_2 > a_1 - \rho b_1 \tag{10}$$

From this equation we have $\rho < \frac{a_2-a_1}{b_2-b_1}$. Now the minimum value of $a_2 - a_1$ is 1 and the maximum value of $b_2 - b_1$ is n (number of sensors). Thus, plugging these values we get an upper bound for ILP which is, $\rho < \frac{1}{n}$.

Lemma 3.1 can be generalized for other formulations (i.e. IQP) as well. Note that, further smaller value of ρ does not change the solution of ILP. That is, if ρ_1 and ρ_2 both are smaller than $\frac{1}{n}$ then ILP will give same solution for these two penalty coefficients.

Note that, if we set $\rho > 1$, then more priority will be given to reduce the number of activated sensors than maximizing the total coverage. Throughout this paper, we penalize the number of active sensors and choose $0 < \rho < \frac{1}{n}$ to ensure unique solutions.

4. Quadratic programming approach

The motivation behind using quadratic programming method to solve HCT problem is to minimize the distance between required and achieved coverage vectors by activating minimum number of sensors. Here the required and achieved coverages both are represented as vectors in multidimensional space. Due to quadratic nature of the formulation, the higher is the difference between required and achieved coverage for a target, the higher that difference will be penalized.

More precisely, we have two m-dimensional vectors (depicted in Fig. 4): $\vec{k} \equiv (k_1, k_2, k_3, \ldots, k_m)$ and $\vec{\psi} \equiv (\psi_1, \psi_2, \psi_3, \ldots, \psi_m)$ where k_t is the coverage requirement and ψ_t is the achieved coverage of target g_t for $t = 1, 2, 3, \ldots, m$.

We want to minimize the vector distance $d = \|\vec{k} - \vec{\psi}\|_2$ and also the total number of active sensors. To include this distance we have a new objective function to minimize:

$$\textbf{minimize}: \sum_{t=1}^{m} (k_t - \psi_t)^2 + \rho \sum_{i=1}^{n} \sum_{j=1}^{q} \chi_{(i,j)} \tag{11}$$

All the other constraints are same as ILP formulation. The objective function of this optimization problem is quadratic in nature. So we call it Integer Quadratic Programming (IQP).

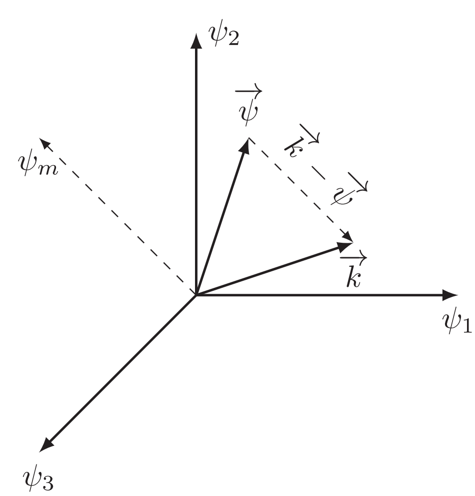

Fig. 4. Two m-dimensional vectors. The target vector: $\vec{k}$ and the achieved vector: $\vec{\psi}$.

Like ILP, IQP achieves the required coverages if the problem is solvable. In this case the vector distance is zero and the objective function of Eqn. (11) is minimized. When the problem is not solvable IQP gives minimum vector distance between achieved and required coverage vectors.

For example, consider the scenario of Fig. 3(b). Here the six targets a, b, c, d, e, f have the coverage requirements of $(3, 3, 2, 2, 1, 1)$ respectively, IQP gives a solution with $2, 2, 1, 2, 0, 0$ coverages respectively and all the sensors are activated. Note that, no orientation of the sensors provide required coverage for all the targets and as a result, the problem instance is not solvable. Here the total squared difference is $(3-2)^2 + (3-2)^2 + (2-1)^2 + (2-2)^2 + (1-0)^2 + (1-0)^2 = 5$. A detailed analysis is given in Table 2.

5. Special constraints on problem statement for under provisioned network

The proposed ILP and IQP cannot achieve the requirements when the network is in under-provisioned state. In this case we impose some special criteria on our basic HCT problem to get different solutions. In this section we explore such two criteria:–one is prioritized solution and the other is group-wise balanced coverage.

5.1. Prioritized coverage

When the problem is not solvable we may assign priorities–targets with higher coverage requirements will be assigned with higher priorities. Basically the motivation behind this method is that due to the multiplication of k_t (required coverage of target g_t) with $(k_t - \psi_t)^2$ (squared difference of required and achieved coverage for target g_t), the optimizer will be compelled to choose higher value of ψ_t (achieved coverage of target g_t) for targets with higher value of k_t so that the value of the whole expression is minimized, as a result, higher priority targets will receive higher coverage. Also we want to achieve such goal activating minimum number of sensors. We modify the objective function of the basic IQP to capture the priority criteria as follows:

$$\textbf{minimize}: \sum_{t=1}^{m} k_t (k_t - \psi_t)^2 + \rho \sum_{i=1}^{n} \sum_{j=1}^{q} \chi_{(i,j)} \tag{12}$$

All other constraints remain the same as ILP. The new objective function tries to minimize the distance between achieved and required coverage vector based on the coverage requirements of the targets. With the introduction of k_t, the term $(k_t - \psi_t)^2$ gets prioritized, according to the requirement k_t. For example, consider a case of three targets with 3, 2 and 1 coverage requirements respectively. Suppose we have two possible coverage tuples: $(2, 2, 1)$ and $(3, 1, 1)$. Also the number of active sensors is same for these two possibilities. In this scenario, the prioritized-IQP (pIQP) will choose the tuple $(3, 1, 1)$ because, the prioritized distance, $3(3-3)^2 + 2(2-1)^2 + 1(1-1)^2 = 2$ is smaller than the alternative, $3(3-2)^2 + 2(2-2)^2 + 1(1-1)^2 = 3$.

5.2. Groupwise balanced coverage

Another approach for under-provisioned case, is to minimize coverage variance within each individual coverage group. It is desirable to have the same coverage for the targets which have the same requirements. The basic ILP or IQP does not guarantee this type of coverage balance. Basically, the motivation behind this method is: besides minimizing the distance between required and achieved coverage vectors by activating minimum number of sensors, we want to minimize the total coverage variance of all the coverage groups too, in other words, we want to minimize the coverage variance within each coverage group. To incorporate the balance we modify the objective function of the basic

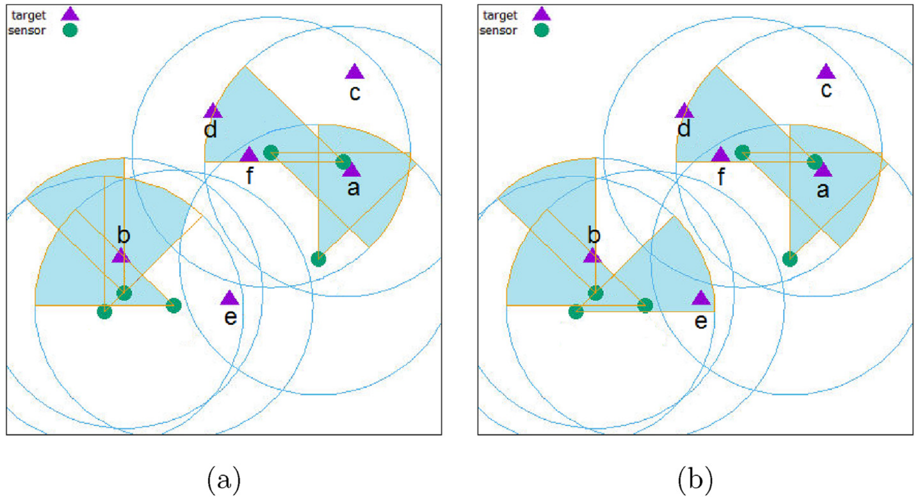

Fig. 5. Solutions produced by pIQP and reduced-variance IQP. There are 6 targets and 6 sensors (a) Selection of pans by pIQP (b) Selection of pans by reduced-variance IQP. A detailed analysis is presented in Table 3.

IQP as:

$$\textbf{minimize}: \sum_{t=1}^{m}(k_t-\psi_t)^2+\sum_{t=1}^{m}\frac{(\psi_t-\mu_{k_t})^2}{m_{k_t}}+\rho\sum_{i=1}^{n}\sum_{j=1}^{q}\chi_{(i,j)} \tag{13}$$

where m_{k_t} is the number of targets that require k_t-coverage and μ_{k_t} represents the mean (or average) coverage of k_t-coverage group. More precisely, μ_{k_t} is defined as:

$$\mu_{k_t}=\frac{\sum_{i=1}^{m}\psi_i[k_i=k_t]}{m_{k_t}} \tag{14}$$

Also, $[k_i = k_t]$ is defined under *Iverson bracket* in Eqn. (3).

The objective function has three terms. The first term, $\sum_{t=1}^{m}(k_t-\psi_t)^2$ reduces the distance between achieved and required coverage vector. The second term, $\frac{(\psi_t-\mu_{k_t})^2}{m_{k_t}}$ is the portion of variance for the target g_t in its coverage group; here, the variance for the coverage group k_t is $\sum_{i=1}^{m}[k_i=k_t]\frac{(\psi_i-\mu_{k_t})^2}{m_{k_t}}$. As a result, the term $\sum_{t=1}^{m}\frac{(\psi_t-\mu_{k_t})^2}{m_{k_t}}$ represents the sum of variance-portion of each target in its coverage group. So this term is the summation of variance in each coverage group, in other words, this term represents total variance of all the coverage groups. The last term, $\rho\sum_{i=1}^{n}\sum_{j=1}^{q}\chi_{(i,j)}$ minimizes the number of active sensors.

All the other constraints remain same as for the ILP.

This IQP reduces the total variance of different coverage groups as well as minimizes the vector distance. As a result this approach provides a balanced coverage and we call it *reduced-variance* IQP.

For example, consider the Fig. 5. In this scenario, there are six targets, a, b, c, d, e, f with coverage requirements, $(3,3,2,2,1,1)$ respectively. In Fig. 5(a) pIQP provides a solution with $(2,3,0,1,0,1)$ coverages respectively. Note that, pIQP tries to give more coverage in 3-coverage group, though, some lower coverage group (i.e. 1 and 2) targets remain uncovered.

In Fig. 5(b) reduced-variance IQP provides a solution with $2,2,0,1,1,1$ coverages respectively. Clearly, in this case, reduced-variance IQP gives a balanced coverage within the coverage groups. A detailed analysis is given in Table 3.

Table 3
A detailed analysis of Fig. 5

Target ID	Requirement	Maximum possible coverage	Coverage (pIQP)	Coverage (reduced-variance IQP)
a	3	3	2	2
b	3	3	3	2
c	2	2	0	0
d	2	2	1	1
e	1	4	0	1
f	1	3	1	1

6. Algorithms

The above-mentioned approaches are time-consuming and are non-scalable for large networks. We therefore present a sensor oriented greedy algorithm, a greedy polynomial time heuristic to solve HCT problem.

This is an iterative algorithm. The basic idea of this algorithm is to activate (at each iteration) a sensor in a pan that maximizes the *benefit* value. The *benefit* of a pan for a sensor depends on the nature of the algorithm. We envision two types of benefit functions– one is linear and the other is quadratic. The algorithm is described in Algorithm 1.

Algorithm 1 at first constructs $\mathcal{T}_{ij}$ $\forall i,j$, based on targets, sensors and all the possible orientations using TIWP test. It then maintains $\mathcal{F}$, a set of all possible (*sensor*, *orientation*) pairs and $\mathcal{N}$, a set of targets that have not achieved their requirements. Initially all the sensors are inactive. In each iteration, a (*sensor*, *orientation*) pair with maximum benefit is selected and is included in a set $\boldsymbol{C}$. If any target within this sensor orientation achieves its requirement, then the target is discarded from $\mathcal{N}$. Also the set $\mathcal{F}$ is updated by removing all the pairs corresponding to the selected sensor. Finally, the algorithm terminates if there is no inactive sensor or there is no target that has not received its required coverage fully. The set $\boldsymbol{C}$ is the final output of this algorithm.

The *benefit* function (Algorithm 2) used in the SOGA algorithm (Algorithm 1) has two versions; *linear*, corresponding to ILP and *quadratic*, corresponding to IQP. In the *linear* version the benefit value is the total number of targets within the orientation of the sensor that have not yet received required coverage fully. This is a modified version of algorithm proposed by Fusco and Gupta (2009). Their greedy algorithm (GA) selects a sensor and orientation that can cover most of the targets that are not yet k-covered, whereas the *linear* version of our algorithm selects a sensor orientation pair that covers most of the targets that are not yet k_t-covered. This *linear* benefit function considers all the unachieved targets as same although an unachieved target with higher coverage gap (i.e. higher $(k_t - a_t)$) should be considered first as a candidate for coverage. The second version (i.e. *quadratic*) of the func-

Algorithm 1 SOGA (Sensor Oriented Greedy Algorithm) to solve HCT problem.
1: $\mathcal{F}$ {a set of all possible (sensor, orientation) pairs}
2: $\mathcal{C} \leftarrow \emptyset$ { a subset of $\mathcal{F}$, final output of this algorithm}
3: $\mathcal{N} \leftarrow \mathcal{G}$ {a set of unachieved targets}
4: $a_t \leftarrow 0$ {achieved coverage for each target g_t upto previous iteration}
5: **While** $\mathcal{F} \neq \emptyset$ **and** $\mathcal{N} \neq \emptyset$ **do**
6: $\quad (s_{i^*}, p_{j^*}) \leftarrow \arg \max_{(s_i, p_j) \in \mathcal{F}} \mathit{benefit}(s_i, p_j)$
7: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{(s_{i^*}, p_{j^*})\}$
8: $\quad$ **for all** $g_t \in \mathcal{T}_{i^*j^*}$ **and** $g_t \in \mathcal{N}$ **do**
9: $\quad\quad a_t \leftarrow a_t + 1$
10: $\quad\quad$ **if** $a_t = k_t$ **then**
11: $\quad\quad\quad \mathcal{N} \leftarrow \mathcal{N} - \{g_t\}$
12: $\quad$ **remove** all pairs for sensor s_{i^*} from $\mathcal{F}$
13: **return** $\mathcal{C}$

tion considers this issue and contributes higher benefit values for higher $(k_t - a_t)$. Also it helps to minimize squared distances between achieved and required coverage vectors which in turn approximates the proposed IQP.

Algorithm 2 *benefit*(s_i, p_j).
1: *value* $\leftarrow 0$
2: **for all** $g_t \in \mathcal{T}_{ij}$ and $a_t < k_t$ **do**
3: $\quad$ **if** *linear* **then**
4: $\quad\quad$ *value* $\leftarrow$ *value* $+ 1$ (i.e. modified from Fusco and Gupta (Fusco and Gupta, 2009))
5: $\quad$ **if** *quadratic* **then**
6: $\quad\quad$ *value* $\leftarrow$ *value* $+ [(k_t - a_t)^2 - (k_t - a_t - 1)^2]$
7: **return** *value*

For example, consider the scenario of Fig. 6. Here we have three sensors and three targets, a, b, c with coverage requirements $(3, 1, 1)$ respectively. The *linear* version of *benefit* function (Algorithm 2) picks the pans according to the number of unachieved targets within a pan. For this version, the final sensor orientations are depicted in Fig. 6(a).

On the other hand, the *quadratic* version chooses the pans according to the coverage gap $(k_t - a_t)$. The final pan selection for this type of *benefit* function is depicted in Fig. 6(b).

6.1. Time complexity

The complexity of the Algorithm 1 in worst case is as follows. The outer while loop iterates $O(n)$ times; for each iteration the line 6 takes $O(nq)$ checking; for each checking the benefit function costs $O(m)$. Also the inner loops cost $O(m)$. So each iteration costs $O(nqm + m)$. Thus the complexity is $O(n^2qm)$.

6.2. Prioritized coverage

Like IQP, prioritized-IQP requires higher computational time than greedy heuristic. The greedy algorithm for this case is similar to Algorithm 1 with necessary changes in the benefit function (Algorithm 3). Unlike the *quadratic* version of Algorithm 2, this *benefit* function imposes priority on the coverage gap by multiplying with k_t. As a result targets with higher coverage requirements contribute higher increase in the benefit value. If the target achieves its requirement it adds zero to the benefit value. The *benefit* function in this case is as follows:

Algorithm 3 *benefit*(s_i, p_j) (for prioritized-IQP).
1: *value* $\leftarrow 0$
2: **for all** $g_t \in \mathcal{T}_{ij}$ and $a_t < k_t$ **do**
3: $\quad$ *value* $\leftarrow$ *value* $+ k_t[(k_t - a_t)^2 - (k_t - a_t - 1)^2]$
4: **return** *value*

For example, consider the scenario of Fig. 7. Here we have two sensors and three targets, a, b, c with coverage requirements $(1, 1, 2)$ respectively. The *benefit* function for prioritized-IQP (Algorithm 3) selects the pans according to the coverage requirements of the targets within a pan. For this version, the final sensor orientations are depicted in Fig. 7(a). It provides $1, 0, 2$ coverages respectively. It is clear from the figure that, greedy algorithm for prioritized-IQP gives more coverage for higher coverage requirement targets.

On the other hand, in Fig. 7(b), the orientations of sensors for the same scenario are selected according to the reduced-variance IQP version of *benefit* function (Algorithm 4). It provides $1, 1, 1$ coverages respectively. Clearly, this version of *benefit* function produces groupwise balanced coverage.

6.3. Groupwise balanced coverage

The greedy algorithm for groupwise balanced coverage is similar to Algorithm 1 with necessary changes in the benefit function. The modified function returns the total change in both vector distance (between required and achieved coverage vector) and variances of the coverage groups due to new orientation of sensors. The new *benefit* function is as follows:

Algorithm 4 *benefit*(s_i, p_j) (for reduced-variance IQP).
1: *value* $\leftarrow 0$
2: **for all** $g_t \in \mathcal{T}_{ij}$ and $a_t < k_t$ **do**
3: $\quad \mu \leftarrow \mu_{k_t}$ {average of k_t-coverage group}
4: $\quad g \leftarrow m_{k_t}$ {number of targets in k_t-coverage group}
5: $\quad$ *old* $\leftarrow (k_t - a_t)^2 + \frac{(a_t - \mu)^2}{g}$
6: $\quad$ *new* $\leftarrow (k_t - a_t - 1)^2 + \frac{(a_t - \mu + 1 - \frac{1}{g})^2}{g}$
7: $\quad$ *value* $\leftarrow$ *value* $+$ (*old* $-$ *new*)
8: **return** *value*

Here, when the coverage of a target g_t is increased from a_t to $a_t + 1$, the mean or average of the corresponding group is changed from μ to $\mu + \frac{1}{g}$.

7. Simulations

For evaluating the performances of the proposed formulations and heuristics, we perform extensive simulation experiments. Especially we provide performance comparisons with two state-of-the-art algorithms: Modified-Greedy (Fusco and Gupta) (the modified version of Fusco and Gupta's (Fusco and Gupta, 2009) greedy algorithm) and Greedy-Razali (the greedy algorithm of Razali et al. (2017)) with our proposed IQP, pIQP, reduced-variance IQP and greedy algorithm. We show that our methodologies and algorithm substantially outperform both the above mentioned algorithms in terms of maximizing coverage of targets according to their requirements. Also we vary the network topology by changing it's size and node distribution. This section

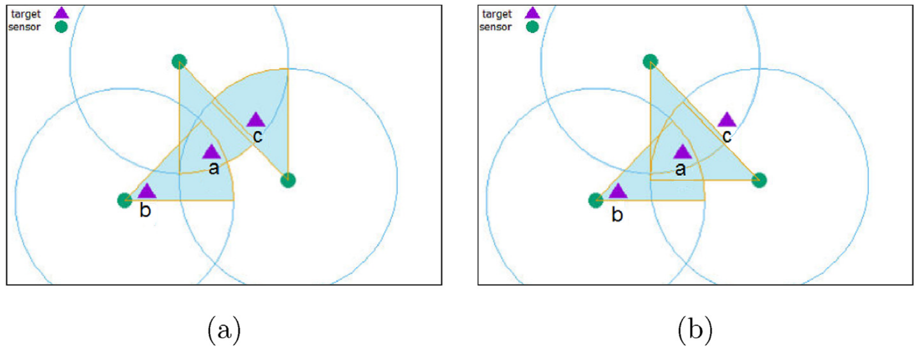

Fig. 6. The outputs of the SOGA (Algorithm 1). (a) Selection of pans by *linear* version and (b) by *quadratic* version of *benefit* function (Algorithm 2).

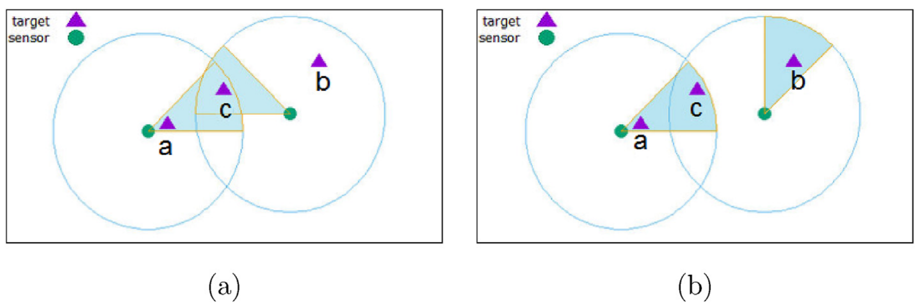

Fig. 7. The outputs of the SOGA (Algorithm 1). (a) Selection of pans by prioritized-IQP version (Algorithm 3) and (b) for reduced-variance IQP version (Algorithm 4) of *benefit* function.

provides detailed experimental setup, performance metrics and performance comparisons.

7.1. Experimental setup

In these simulations we use a 2D grid as our deployment area with size 200×200 sq. units for "Small" scale and 1000×1000 sq. units for "Large" scale network. According to our model camera can move only in horizontal direction. Therefore, we can envision deployment area placed in a two dimensional space. Both the targets and sensors are considered as points and are randomly deployed. For the deployment of sensors, we consider two types of distributions:

1. *Uniform:* sensors are distributed uniformly all over the deployment area.
2. *Zipf:* 80% sensors are distributed within 20% of the area and remaining 20% sensors are distributed within 80% of the deployment area. Fig. 8 depicts zipf distribution of 100 sensors within 100×100 deployment area. Rahman et al. (2009) use this distribution for performance analysis.

The range of each sensor $R_s = 20$ units and FoV, $\theta = \frac{\pi}{4}$ for both sized networks. Thus each sensor has eight non-overlapping pans. Coverage requirements are 1, 2 and 3 with uniform number of targets; that is there are $\frac{m}{3}$ targets in each coverage requirement group where m is the number of total targets. We set $\rho = 0.0001$ for ILP and IQPs.

For small scale network, we perform two types of simulations. In the first type, we vary the number of targets from 3 to 120, while keeping the number of sensors fixed at 30 and in the second type, we vary the number of sensors from 3 to 120, while keeping the number of targets fixed at 30.

For large scale network, we also perform two types of simulations. In the first case, we vary the number of targets from 6 to 180, while keeping the number of sensors fixed at 45 and in the second type, we vary the number of sensors from 6 to 210, while keeping the number of targets fixed at 45.

For each type of simulations, we generate the scenarios in such a way that, a larger scenario contains the smaller one. All the simulations are performed in JAVA. We have used CPLEX (IBM ILOG CPLEX) optimizer library to solve ILP and IQPs. The simulations are performed in

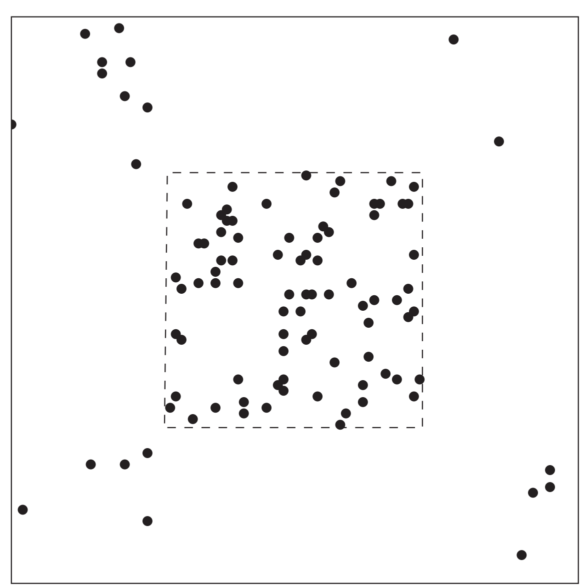

Fig. 8. Zipf distribution of sensors. Here 100 sensors (filled, black circles) are distributed on 100×100 area. However, 80% of the sensors are distributed on 20% of the area (dashed square, around 45×45) and remaining 20% sensors are distributed on rest of the area.

Table 4
Table of experimental parameters.

Parameter	Value
Deployment area	200×200 sq. units (small) & 1000×1000 sq. units (large)
Range of a sensor, R_s	20 units
Field of view of a sensor, θ	$\frac{\pi}{4}$
Fixed number of sensors	30, while varying number of targets from 3 to 120 (small)
	45, while varying number of targets from 6 to 180 (large)
Fixed number of targets	30, while varying number of sensors from 3 to 120 (small)
	45, while varying number of sensors from 6 to 210 (large)

a computer having 8 GB RAM, 1 TB Hard Disk Drive and Intel Core i5 2.30 GHz CPU.

The values of different simulation parameters are summarized in Table 4.

7.2. Performance metrics

We use five metrics defined below to compare performance of all approaches:

Distance Index ($\mathcal{DI}$). The primary performance metric is *distance index* ($\mathcal{DI}$) which is defined as follows:

$$\mathcal{DI} = \frac{\sum_{t=1}^{m} k_t^2 - \sum_{t=1}^{m} (k_t - \psi_t)^2}{\sum_{t=1}^{m} k_t^2} \tag{15}$$

where $\sum_{t=1}^{m} k_t^2$ is the maximum possible squared-distance. A higher value of this metric indicates better coverage achieved with respect to requirements.

Activated Sensors. To analyze sensor usage the number of *activated sensors* in each approach is counted. An algorithm may not activate all the sensors. If an algorithm provides same coverage as another algorithm, but activates less number of sensors, we say that first algorithm is better than the second, as the more is the sensor usage, the more is network installation and maintenance cost. A lower value of activated sensors indicates better sensor usage.

Variance. To evaluate the group-wise balance approach, *variance* is measured which is the sum of group-wise variances. A lower value on this metric indicates better coverage balancing. Formally, we measure the following term:

$$\sum_{t=1}^{m} \frac{(\psi_t - \mu_{k_t})^2}{m_{k_t}} \tag{16}$$

where m_{k_t} is number of targets that require k_t-coverage and μ_{k_t} represents the mean (or average) coverage of k_t-coverage group. More precisely, μ_{k_t} is defined as:

$$\mu_{k_t} = \frac{\sum_{i=1}^{m} \psi_i [k_i = k_t]}{m_{k_t}} \tag{17}$$

Here, $[k_i = k_t]$ is defined under *Iverson bracket* in Eqn. (3). The term $\frac{(\psi_t - \mu_{k_t})^2}{m_{k_t}}$ is the portion of variance for the target g_t in its coverage group. As a result, the term $\sum_{t=1}^{m} \frac{(\psi_t - \mu_{k_t})^2}{m_{k_t}}$ represents the sum of variance-portion of all the targets which in turn is the total variance of all the coverage groups.

Power(W). We measure consumed *power (W)* to compare power expenditure in each approach. Consumed power is an important factor for a network. Energy-efficient network is necessary for many applications. A lower value of this metric indicates better energy efficiency.

Coverage Quality. Finally we compute *coverage quality* for all the proposed approaches. In this work, we primarily focus on "coverage quantity" or the number of sensors covering each target. However, the distance between targets and sensors affects the coverage or monitoring quality. Usually coverage quality degrades as the distance between targets and sensors increases. In Yang et al. (2010), Razali et al. (2017) and Mohamadi et al. (2014) the coverage quality function is defined as: $u(x) = 1 - x^2$ where x is the ratio of the distance between the target and sensor to the sensing range.

Here we adopt the same notion. More specifically, the achieved coverage quality of a target g_t within a (sensor, pan) pair, (s_i, p_j) is defined as:

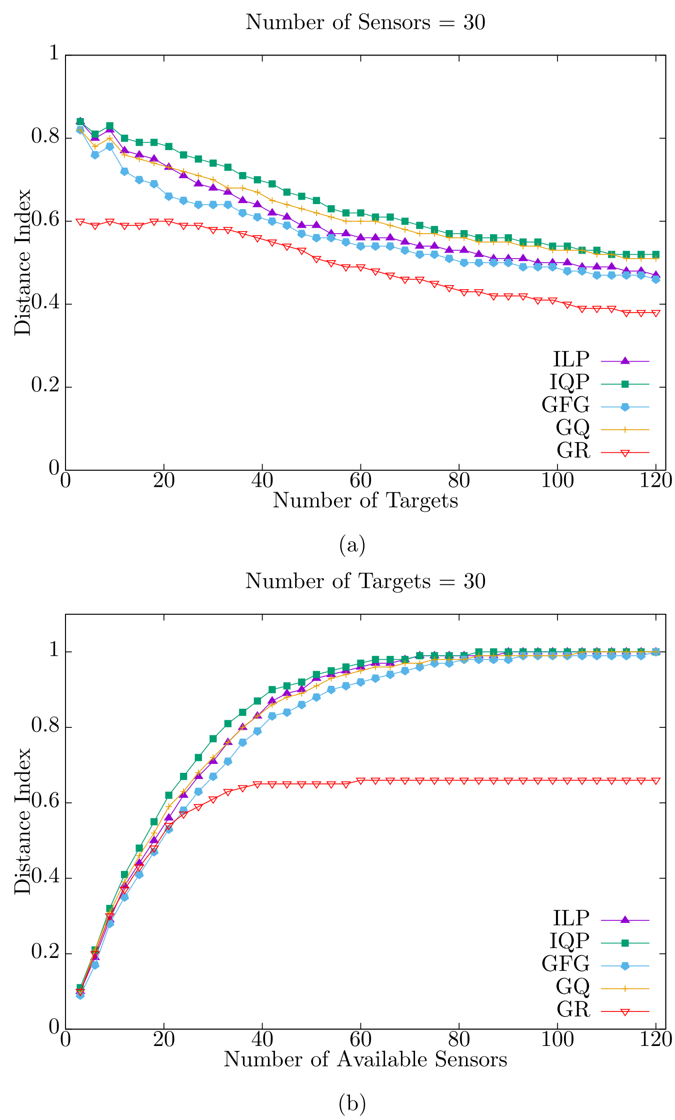

Fig. 9. For small scale network, uniform distribution of sensors: Distance index comparison of ILP, IQP, GQ, GFG and GR. (a) keeping sensor number fixed at 30, increasing the number of targets from 3 to 120 (b) keeping target number fixed at 30, increasing sensor number from 3 to 120.

$$U(i,j,t) = \begin{cases} 1 - \left(\frac{|\vec{v}_{it}|}{R_s}\right)^2, & \text{if } g_t \in \mathcal{T}_{ij} \text{ and } |\vec{v}_{it}| < R_s \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

where $\vec{v}_{it}$ is a vector pointing from sensor s_i to target g_t and R_s is the sensing range of each homogeneous sensor.

Eventually the total coverage quality (or, simply coverage quality or $\boldsymbol{CQ}$) is the total achieved qualities of all the targets for a given solution:

$$\boldsymbol{CQ} = \sum_{i,j,t} U(i,j,t)[s_i \textit{ is activated in } p_j] \tag{19}$$

The higher value of this index denotes better quality of coverage.

Note that, boundary between over-provisioned and under-provisioned network is same for all graphs with varying “Number of Targets” and the boundary is also same for all graphs with varying “Number of Available Sensors”; at such boundary, sensor usage reaches saturation (evident from Fig. (Fig. 10(a)) and (Fig. 10(b))).

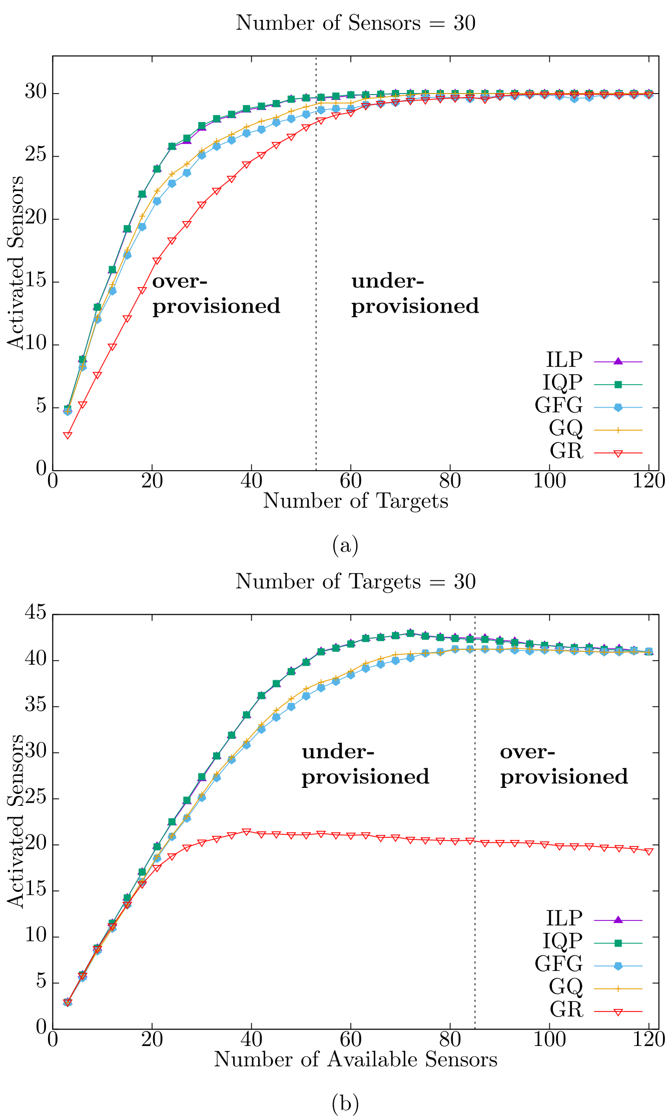

Fig. 10. For small scale network, uniform distribution of sensors: Sensor usage comparison of ILP, IQP, GQ, GFG and GR. (a) keeping sensor number fixed at 30, increasing the number of targets from 3 to 120 (b) keeping target number fixed at 30, increasing sensor number from 3 to 120.

Table 5
Table of abbreviations.

Name	Abbreviation
Integer Linear Programming	ILP
Integer Quadratic Programming	IQP
Prioritized IQP	PIQP
Reduced Variance IQP	RVIQP
Greedy Quadratic	GQ
Prioritized Greedy Quadratic	PGQ
Modified-Greedy (Fusco and Gupta)	GFG
Greedy-Razali	GR
Reduced Variance Greedy Quadratic	RVGQ

7.3. Performance comparisons with other algorithms

The greedy algorithms proposed by Fusco and Gupta (2009) are for k-coverage problem for gaining coverage maximization with minimum sensors, which is a special case of our problem domain, when all the targets have k-coverage requirement. Therefore we choose their greedy algorithm (with modification for heterogeneous or k_t-coverage requirement, we showed the modification in Algorithm 1 inside benefit function under linear case) to provide performance comparison with other approaches.

We incorporate another work for performance comparison. In this work (Razali et al., 2017), Razali et al. propose a greedy based and a learning-automata based algorithm to solve priority-based target coverage with adjustable sensing ranges (PTCASR) problem. Here targets have different coverage quality requirements that reflect their priorities and sensors are directional with adjustable ranges. The problem is to find some cover sets with appropriate sensor configurations each of which provides required coverage quality to all the targets and maximize the network lifetime.

We choose their greedy-based algorithm (calling it Greedy-Razali) to show comparisons. Here we choose the input values of different parameters as follows:

- The number of power levels, $a = 1$; in this algorithm a denotes the alternative power levels which in turn denotes the number of different ranges for a sensor.
- The criteria weighting parameter, $\Gamma = (1, 0, 0)$; in each iteration the algorithm chooses a sensor direction that maximizes a benefit value. This value is computed based on three criteria i.e covering power (CP), covering waste (CW), and residual lifetime (RL). The input vector $\Gamma = (\gamma_1, \gamma_2, \gamma_3)$ is used for weighting the three above-mentioned criteria based on relevance. We choose only the covering power (CP) to compute the value.
- The coverage quality requirement for target g_t as $\frac{k_t}{n_{\max}}$; in this algorithm, the coverage quality requirements are selected between 0 and 1. Here we assign the quality requirements according to the coverage requirements of the targets. Note that, in these simulations maximum number of sensors $n_{\max}$ is 120 for small network and 210 for large network.

This algorithm may generate more than one cover set. Without loss of generality we choose the first cover set and apply the above mentioned metrics to compare performances.

We show that our methodologies and algorithm substantially outperform both the above mentioned algorithms in terms of maximizing coverage of targets according to their requirements.

For convenience we use abbreviations for the formulations and algorithms. The abbreviations are presented in Table 5.

7.4. Small scale network with uniform distribution

In this subsection, we perform simulations for small scale network with uniform sensor distribution. Detailed analyses are given below:

7.4.1. Distance index analysis

In first set of experiments, we gradually increase number of targets from 3 to 120, while keeping the number of sensors fixed at 30. As the number of targets gradually increases the network enters into under-provisioned system from over-provisioned system and the distance between achieved and required coverage vectors also gradually increases. The $\mathcal{DI}$ curve is shown in Fig. 9. From Fig. 9 (a), it is evident that the distance index decreases as the number of targets increases and IQP performs best among all approaches. We can interpret the fact that distance index decreases as the number of targets increases in the following way:

From Eqn. (15), we get:

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

or,

$$\mathcal{DI} = 1 - \frac{\sum_{t=1}^{m} (k_t - \psi_t)^2}{\sum_{t=1}^{m} k_t^2} \tag{20}$$

With the increase of number of targets, both $\sum_{t=1}^{m} (k_t - \psi_t)^2$ and $\sum_{t=1}^{m} k_t^2$ from Eqn. (20) increases. But the amount of increase for $\sum_{t=1}^{m} (k_t - \psi_t)^2$ is less than or equal to the amount of increase for $\sum_{t=1}^{m} k_t^2$ as $\psi_t \geq 0$ for all $t = 1, 2, \ldots, m$. As a result, with the increase of number of targets, the fraction $\frac{\sum_{t=1}^{m} (k_t-\psi_t)^2}{\sum_{t=1}^{m} k_t^2}$ from Eqn. (20) increases making $\mathcal{DI}$ decrease.

In the second set of experiments, we gradually increase the number of sensors from 3 to 120, while keeping the number of targets fixed at 30. Clearly, the network changes from under-provisioned to over-provisioned state. The result is shown in Fig. 9(b). With the increase of number of available sensors, achieved coverage of each target increases, i.e., ψ_t increases for $t = 1, 2, \ldots, m$. As a result, with the increase of number of available sensors, the fraction $\frac{\sum_{t=1}^{m} (k_t-\psi_t)^2}{\sum_{t=1}^{m} k_t^2}$ from Eqn. (20) decreases making $\mathcal{DI}$ increase. But ψ_t becomes close to k_t at a point and as ψ_t is bounded by k_t, $\mathcal{DI}$ becomes saturated to value 1 except Greedy-Razali (evident from Eqn. (20)). We see the validation of this fact from Fig. 9(b). The distance index becomes higher as the number of sensor increases and IQP performs the best. We see from Fig. 9(b) that with the increase of number of available sensors, distance index becomes saturated at a point of time. At such saturation, value of distance index becomes close to 1. From the equation of distance index (Eqn. (15)), we understand that value of distance index becomes 1 when all the targets receive their necessary coverages.

For the Greedy-Razali algorithm, $\mathcal{DI}$ index becomes saturated at below 1. Once all the targets get their quality requirements, increasing the number of sensors does not change the distance index value.

From Fig. 9 it is evident that for both kinds of simulations greedy algorithms are able to approximate the corresponding ILP and IQP formulations. More specifically, the Modified-Greedy (Fusco and Gupta) algorithm approximates ILP and Greedy-Quadratic approximates IQP. Hence, we conclude that IQP is better in achieving $\mathcal{DI}$ than other approaches and Greedy-Quadratic approximates the optimal solution very closely.

7.4.2. Sensor usage analysis

Fig. 10 shows the number of activated sensors for both kinds of simulation experiments. In the first kind when the number of targets increases (Fig. 10(a)) the number of activated sensors also increases. But when the network enters into under-provisioned state, all the sensor usage reaches saturation and further increase in targets does not change the shape. Also it is evident from Fig. 10(a) that all the approaches use almost same number of sensors. The Greedy-Razali method reduces sensor usage, but in distance index analysis this method performs worse than IQP.

When we increase the number of sensors (Fig. 10(b)) the number of activated sensors become fixed after a certain number of available sensors. At that point the number of sensors becomes good enough to meet all coverage or quality requirements and the network turns into over-provisioned system. As a result, further increase in sensor does not affect sensor usage. Also when the network remains in under-provisioned state, Greedy-Razali uses fewest sensors but it performs worse than IQP in achieving $\mathcal{DI}$.

7.4.3. Relation between distance index analysis and sensor usage analysis graphs at sensor saturation

With the increase of number of targets, sensor usage reaches saturation at a point of time and the network enters into under-provisioned state at that point (evident from Fig. 10 (a)). When the sensor usage reaches saturation, the newly added targets do not get required coverage and as a result Distance Index value lowers down, evident from Fig. 9 (a).

With the increase of number of available sensors, sensor usage reaches saturation at a point of time and the network enters into over-provisioned state at that point (evident from Fig. 10 (b)). The saturation in sensor usage refers to the fact that newly added sensors do not help in improving required coverage of the targets as all the targets have already received their required coverage. This is evident from Fig. 9 (b) as value of Distance Index is becoming saturated to 1.

7.4.4. Power analysis

In order to measure power consumption we use the same model provided in Farzana et al. (2016). In this model, a visual sensor can be either in three possible states: *sleep, idle, active*. In *active* state sensors monitor the targets, transmit accrued data and consume highest power among other states. In *idle* state sensors are turned on but does not perform any specific task rather than simply running the OS processes in background. In *sleep* state sensors are inactive and consume lowest power. The total consumed power (P_c) is calculated as:

$P_c = (\textit{number of active sensors}) \times P_a + (\textit{number of idle sensors}) \times P_i + (\textit{number of sensors in sleep state}) \times P_s$

where P_a is consumed power of an active sensor, P_i is consumed power of an idle sensor and P_s is consumed power of a sensor in sleep state. We use the same values used for Panoptes video sensors (Feng et al., 2005) which are $P_a = 5.268$ W, $P_i = 1.473$ W and $P_s = 0.058$ W.

In Fig. 11(a) the number of targets is increased from 3 to 120 with fixed 30 sensors. So the network gradually enters into under-provisioned state, more and more sensors get activated and the total power consumption increases. ILP and IQP both consume almost same power and their greedy versions almost merge with them. When the network is in under-provisioned state, the sensor usage reaches saturation and the total consumed power remains constant, i.e., further increase in targets does not affect the power consumption.

In Fig. 11(b) we change the network from under-provisioned to over-provisioned state by adding sensors, while keeping the number of targets fixed at 30. In under-provisioned state any added new sensor becomes immediately activated. As a result, power consumption increases gradually. When the network enters into over-provisioned state, adding new sensors does not matter much and the power consumption remains almost flat. The ILP and IQP consume almost same amount of power. Greedy algorithms follow corresponding ILP, IQP patterns. Here the Greedy-Razali algorithm consumes lowest power.

7.4.5. Relation between sensor usage analysis and power analysis graphs

With the increase of number of targets, sensor usage rises at first, then becomes saturated at a point and at that point network enters into under-provisioned state (evident from Fig. 10 (a)). We see similar pattern in power analysis graph too. From Fig. 11 (a), we see that, with the

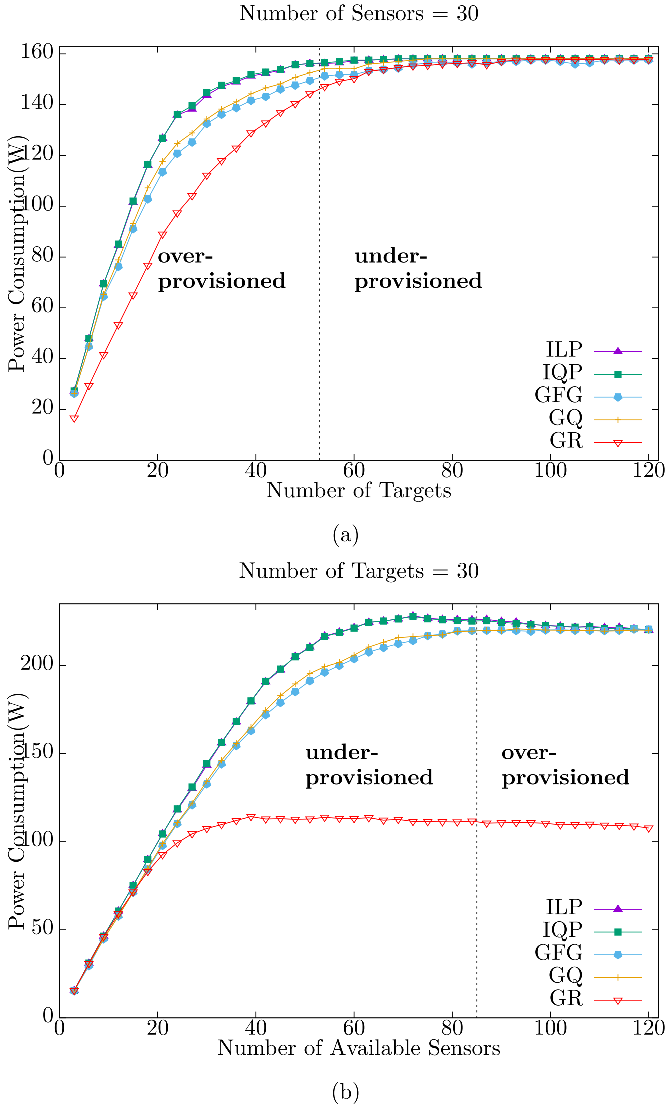

Fig. 11. For small scale network, uniform distribution of sensors: Power consumption comparison of ILP, IQP, GQ, GFG and GR. (a) keeping sensor number fixed at 30, increasing the number of targets from 3 to 120 (b) keeping target number fixed at 30, increasing sensor number from 3 to 120.

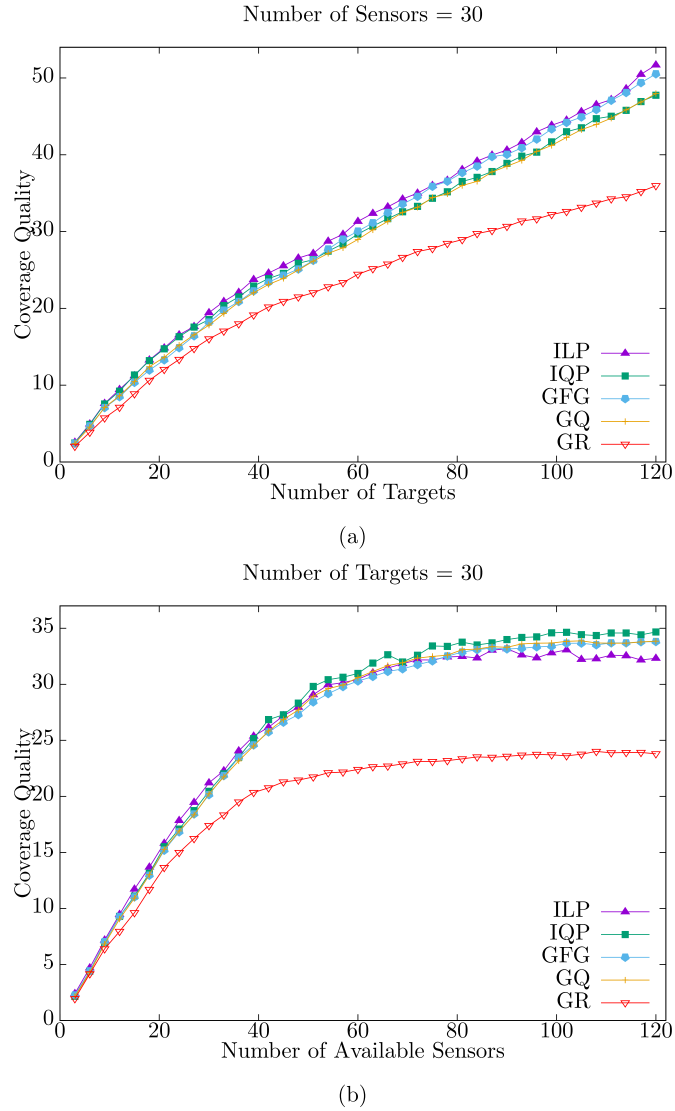

Fig. 12. For small scale network, uniform distribution of sensors: Coverage quality comparison of ILP, IQP, GQ, GFG and GR. (a) keeping sensor number fixed at 30, increasing the number of targets from 3 to 120 (b) keeping target number fixed at 30, increasing sensor number from 3 to 120.

increase of number of targets, power consumption by the activated sensors rises at first, then becomes saturated at a point and at that point, network enters into under-provisioned state. We can conclude that with the increase of number of targets, sensor usage and power consumption by activated sensors follow similar pattern. This is also intuitive as power consumption is proportional to sensor usage.

With the increase of number of available sensors, sensor usage rises at first, then becomes saturated at a point and at that point, network enters into over-provisioned state (evident from Fig. 10 (b)). We see similar pattern in power analysis graph too. From Fig. 11 (b), we see that, with the increase of number of available sensors, power consumption by activated sensors rises at first, then becomes saturated at a point and at that point, network enters into over-provisioned sate. We can conclude that with the increase of number of available sensors, sensor usage and power consumption by activated sensors follow similar pattern. This is also intuitive as power consumption is proportional to sensor usage.

7.4.6. Coverage quality analysis

Fig. 12 shows the coverage quality index ($\mathcal{CQ}$) for both kinds of simulation experiments. In the first kind when the number of targets increases (Fig. 12(a)) the coverage quality increases. Here increasing the number of targets increases the density of objects (i.e. sensors and targets) and thus reduces the distances among sensors and targets. Also additional targets contribute to the $\mathcal{CQ}$ value. Thus coverage quality increases. Here ILP and IQP both achieve maximum coverage quality.

When we increase the number of sensors (Fig. 12(b)) the coverage quality also increases. But in this case, coverage quality becomes saturated after certain point. Though increasing the number of sensors reduces the distances among sensors and targets, the sensor usage becomes saturated; additional sensors do not need to be activated once the requirements are met. As a result $\mathcal{CQ}$ index becomes saturated.

Also, for both these cases, IQP achieves maximum coverage quality. Note that, performance of Greedy-Razali depends on given priorities of targets.

7.4.7. Prioritized coverage analysis

Fig. 13 depicts the group-wise $\mathcal{DI}$ of Modified-Greedy (Fusco and Gupta), IQP, prioritized-IQP, greedy-quadratic, Greedy-Razali and prioritized-greedy for an under-provisioned network.

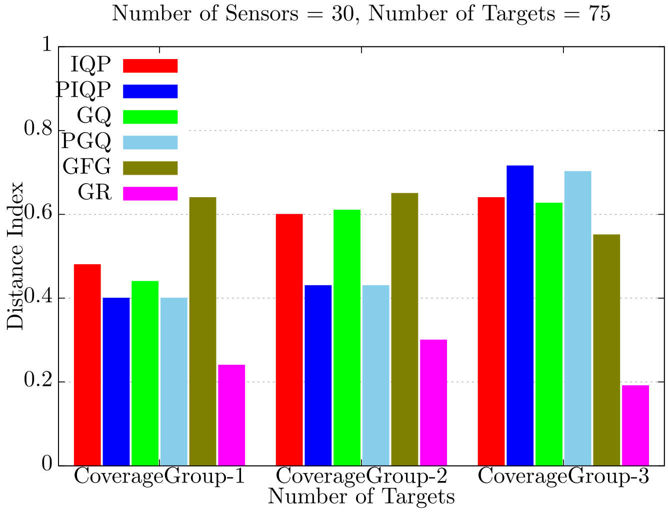

Fig. 13. Distance index for various coverage groups for under-provisioned network with number of targets = 75 and number of sensors = 30.

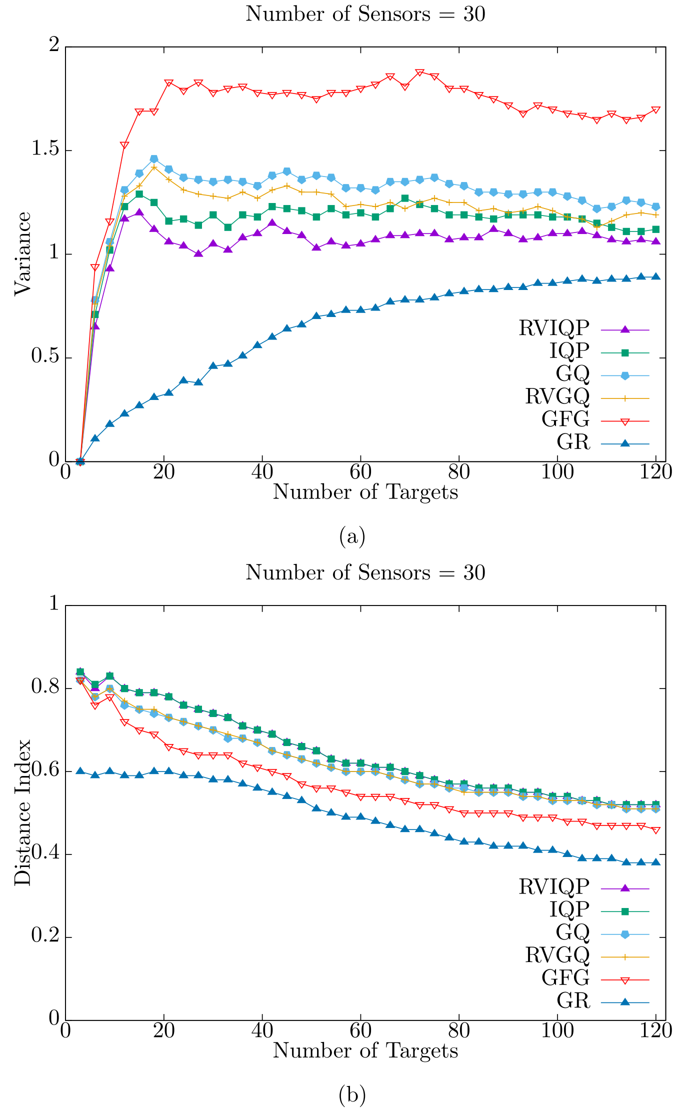

Fig. 14. Variance Analysis for under-provisioned network. (a) Total variance of three coverage groups vs Number of Targets. (b) Distance index vs Number of Targets. Sensor number is fixed at 30 and Targets changing from 3 to 120.

In this case the number of targets is 75 and the number of sensors is 30. From the Fig. 10(a) it is clear that the scenario is in under-provisioned state. The prioritized IQP (pIQP) provides more coverage in 3-coverage requirement group as it assigns highest priority to that group. As a result the $\mathcal{DI}$ in coverage group 3 is the maximum for pIQP. IQP provides more coverage in 1-coverage and 2-coverage requirement group than the prioritized IQP (pIQP), but the prioritized IQP (pIQP) provides greater coverage in 3-coverage requirement group than IQP–this was the main motivation behind designing prioritized IQP (pIQP)–to provide more coverage in higher coverage requirement group. For all the three groups, clearly the priority based greedy-quadratic algorithm approximates pIQP. The Modified-Greedy (Fusco and Gupta) and Greedy-Razali do not impose any priority and thus cannot provide higher $\mathcal{DI}$ in coverage group 3 than pIQP and priority based greedy-quadratic.

7.4.8. Variance analysis

In Fig. 14(a) the group-wise coverage balance is captured using total variance. Total variance is the summation of variance within each coverage group. The number of targets is increased from 3 to 120 while keeping total sensors fixed at 30. The curves of total variance for Modified-Greedy (Fusco and Gupta), IQP, reduced-variance IQP, Greedy-Quadratic, Reduced Variance Greedy-Quadratic and Greedy-Razali are presented. It is evident from the graph that Greedy-Razali can reduce the total variance among other approaches. But it performs worse than IQP in $\mathcal{DI}$ analysis (Fig. 14(b)). Also the Greedy-Variance approximates the reduced-variance IQP the Greedy-Quadratic performs better than Modified-Greedy (Fusco and Gupta).

The curves of Fig. 14(a) reveal an interesting nature of total variance. The total variance becomes saturated with some fluctuations when the network enters into under-provisioned state. This is because at that state additional targets get very few coverage as the number of sensors is fixed. As a result additional targets contribute nearly zero to the total variance which makes it almost constant in under-provisioned state.

The small fluctuations are due to the fact that a few of the newly added targets receive some coverage by the existing deployed sensors. Fig. 14(b) depicts the $\mathcal{DI}$ values for above mentioned approaches. Here the reduced-variance IQP has highest $\mathcal{DI}$ value. The basic IQP almost follows the reduced-variance IQP but from 14(a) we see, reduced-variance IQP has lower variance than basic IQP. Greedy approaches approximate the corresponding formulations.

Here considering both the distance index and groupwise variance, reduced-variance IQP performs best and the corresponding greedy, RVGQ approximate the formulation.

7.5. Large scale network

In this subsection, we present the performance results for large scale network with uniform sensor distribution. Detailed analyses for Distance Index and Activated Sensors are given below:

7.5.1. Distance index analysis

In first set of experiments, we gradually increase number of targets from 6 to 180, while keeping the number of sensors fixed at 45. As the number of targets gradually increases the network enters into under-provisioned system from over-provisioned system and the distance between achieved and required coverage vectors also gradually increases. The $\mathcal{DI}$ curve is shown in Fig. 15. From Fig. 15 (a), it is evident that the distance index decreases as the number of targets increases and IQP performs best among all approaches.

In the second set of experiments, we gradually increase the number of sensors from 6 to 210, while keeping the number of targets fixed

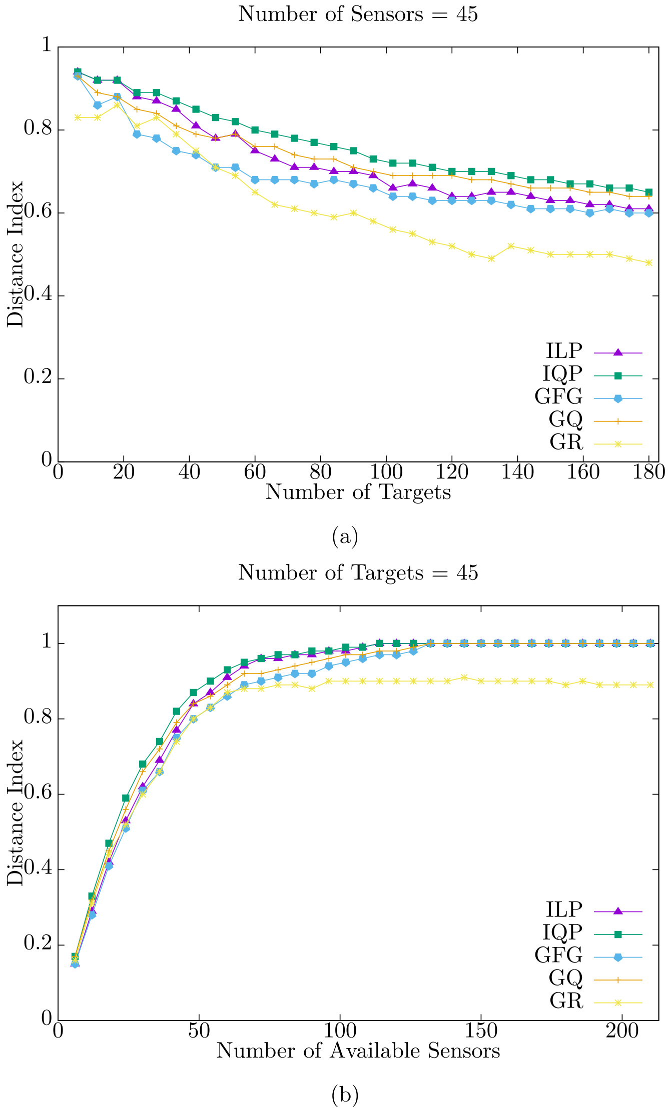

Fig. 15. For large scale network, uniform distribution of sensors: Distance index comparison of ILP, IQP, GQ, GFG and GR (a) keeping sensor number fixed at 45, increasing the number of targets from 6 to 180 (b) keeping target number fixed at 45, increasing sensor number from 6 to 210.

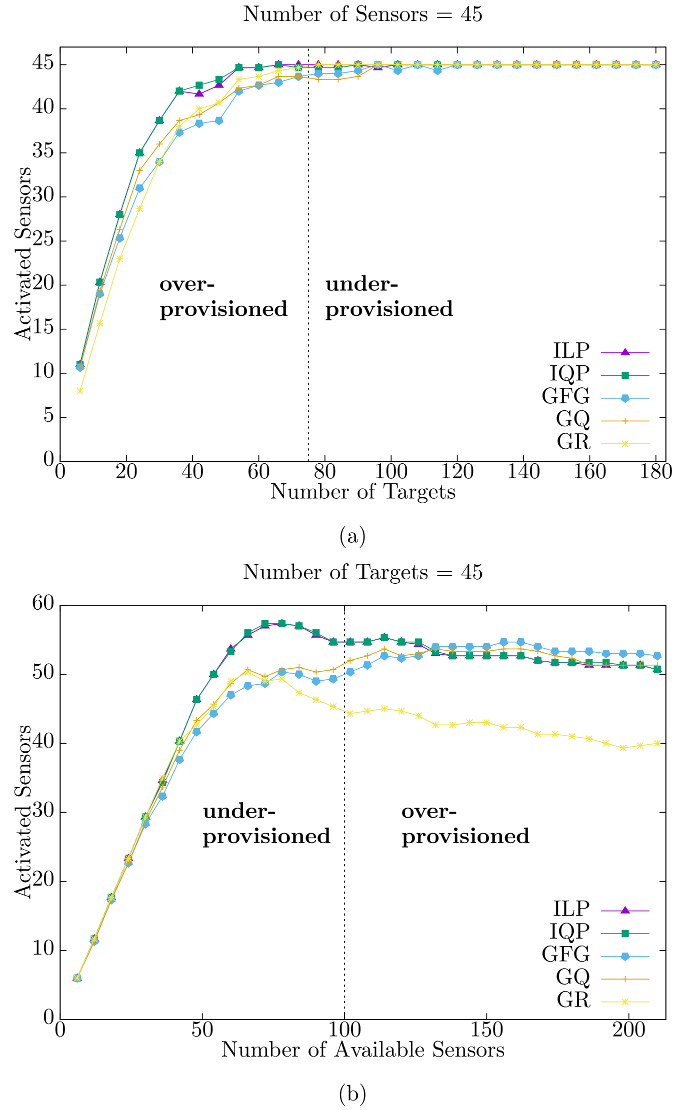

Fig. 16. For large scale network, uniform distribution of sensors: Sensor usage comparison of ILP, IQP, GQ, GFG and GR. (a) keeping sensor number fixed at 45, increasing the number of targets from 6 to 180 (b) keeping target number fixed at 45, increasing sensor number from 6 to 210.

at 45. Clearly, the network changes from under-provisioned to over-provisioned state. The result is shown in Fig. 15(b). The distance index becomes higher as the number of sensor increases and IQP performs the best. We see from Fig. 15(b) that with the increase of number of available sensors, distance index becomes saturated at a point of time. At such saturation, value of distance index becomes close to 1. From the equation of distance index (Eqn. (15)), we understand that value of distance index becomes 1 when all the targets receive their necessary coverages.

For the Greedy-Razali algorithm, $\mathcal{DI}$ index becomes saturated at below 1. Once all the targets get their quality requirements, increasing the number of sensors does not change the distance index value.

From Fig. 15 it is evident that for both kinds of simulations greedy algorithms are able to approximate the corresponding ILP and IQP formulations. More specifically, the Modified-Greedy (Fusco and Gupta) algorithm approximates ILP and Greedy-Quadratic approximates IQP. The Greedy-Razali performs worse than IQP in all cases. Hence, we conclude that IQP is better in achieving $\mathcal{DI}$ than other approaches and Greedy-Quadratic approximates the optimal solution very closely.

It is evident from Figs. 9 and 15 that, for large scale network, shapes of the curves are similar to those of the small scale network. Therefore our proposed approaches do not depend on scale of the network.

7.5.2. Sensor usage analysis

Fig. 16 shows the number of activated sensors for both kinds of simulation experiments in large network.

In the first kind when the number of targets increases (Fig. 16(a)) the number of activated sensors also increases. But when the network enters into under-provisioned state, all the sensor usage reaches saturation and further increase in targets does not change the shape. Also it is evident from Fig. 16(a) that all the approaches use almost same number of sensors in under-provisioned state.

When we increase the number of sensors (Fig. 16(b)) the number of activated sensors become fixed after a certain number of available sensors. At that point the number of sensors becomes good enough to meet all coverage or quality requirements and the network turns into over-provisioned system. As a result, further increase in sensor does not affect sensor usage. Also when the network enters into over-provisioned state, Greedy-Razali uses fewest sensors but it performs worse than IQP

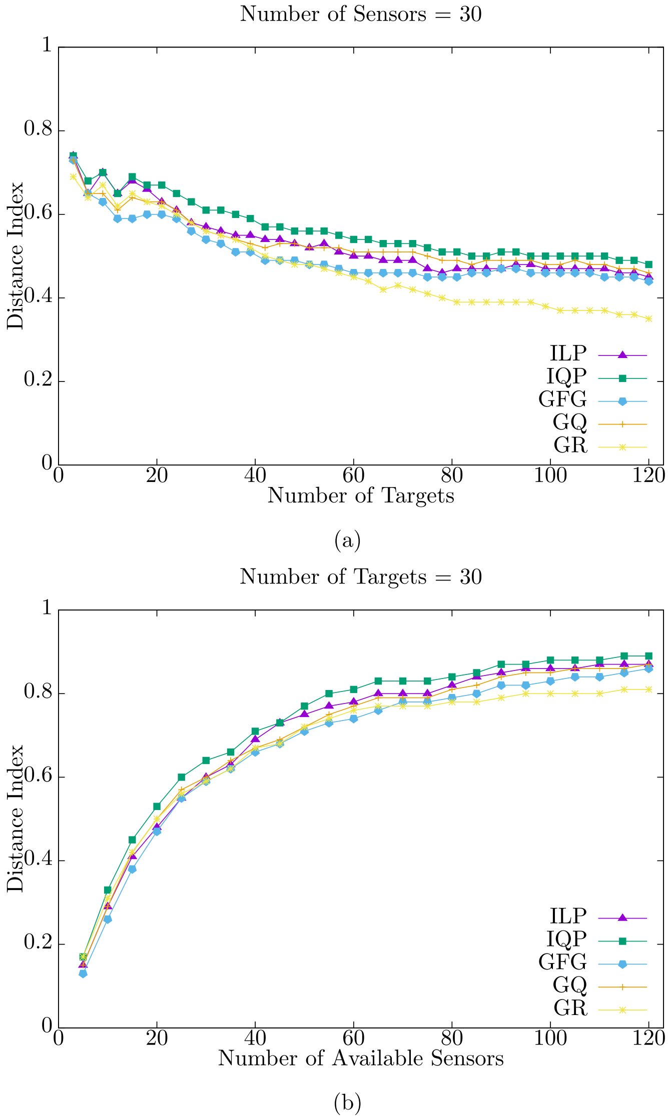

Fig. 17. For zipf distribution of sensors: Distance index comparison of ILP, IQP, GQ, GFG and GR (a) keeping sensor number fixed at 30, increasing the number of targets from 3 to 120 (b) keeping target number fixed at 30, increasing sensor number from 5 to 120.

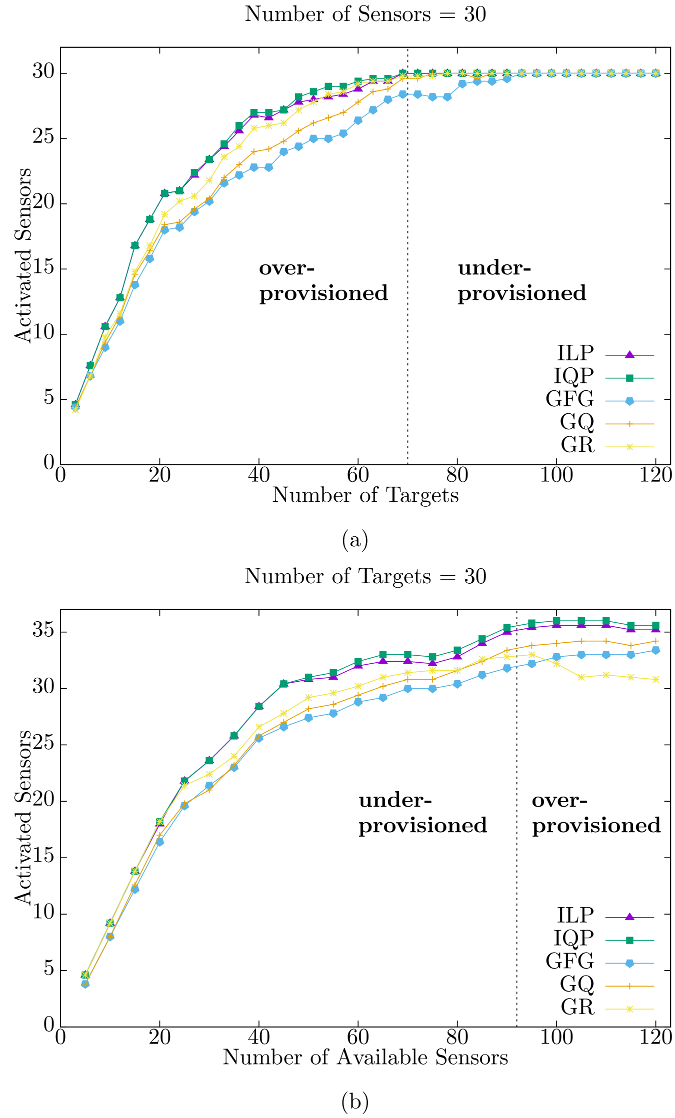

Fig. 18. For zipf distribution of sensors: Sensor usage comparison of ILP, IQP, GQ, GFG and GR. (a) keeping sensor number fixed at 30, increasing the number of targets from 3 to 120 (b) keeping target number fixed at 30, increasing sensor number from 5 to 120.

in achieving $\mathcal{DI}$.

7.6. Zipf distribution

In this subsection, we extend our simulations for zipf distribution of sensors. We perform Distance Index and Sensor Usage analysis in small network as following:

7.6.1. Distance index analysis

In first set of experiments, we gradually increase number of targets from 3 to 120, while keeping the number of sensors fixed at 30. As the number of targets gradually increases the network enters into under-provisioned system from over-provisioned system and the distance between achieved and required coverage vectors also gradually increases. The $\mathcal{DI}$ curve is shown in Fig. 17. From Fig. 17 (a), it is evident that the distance index decreases as the number of targets increases and IQP performs best among all approaches.

In the second set of experiments, we gradually increase the number of sensors from 3 to 120, while keeping the number of targets fixed at 30. Clearly, the network changes from under-provisioned to over-provisioned state. The result is shown in Fig. 17(b). The distance index becomes higher as the number of sensor increases and IQP performs the best. We see from Fig. 17(b) that with the increase of number of available sensors, distance index becomes saturated at a point of time. At such saturation, value of distance index becomes close to 1. From the equation of distance index (Eqn. (15)), we understand that value of distance index becomes 1 when all the targets receive their necessary coverages.

From Fig. 17 it is evident that for both kinds of simulations greedy algorithms are able to approximate the corresponding ILP and IQP formulations. More specifically, the Modified-Greedy (Fusco and Gupta) algorithm approximates ILP and Greedy-Quadratic approximates IQP. The Greedy-Razali performs worse than IQP in all cases. Hence, we conclude that IQP is better in achieving $\mathcal{DI}$ than other approaches and Greedy-Quadratic approximates the optimal solution very closely.

7.6.2. Sensor usage analysis

Fig. 18 shows the number of activated sensors for both kinds of simulation experiments. In the first kind when the number of targets increases (Fig. 18(a)) the number of activated sensors also increases.

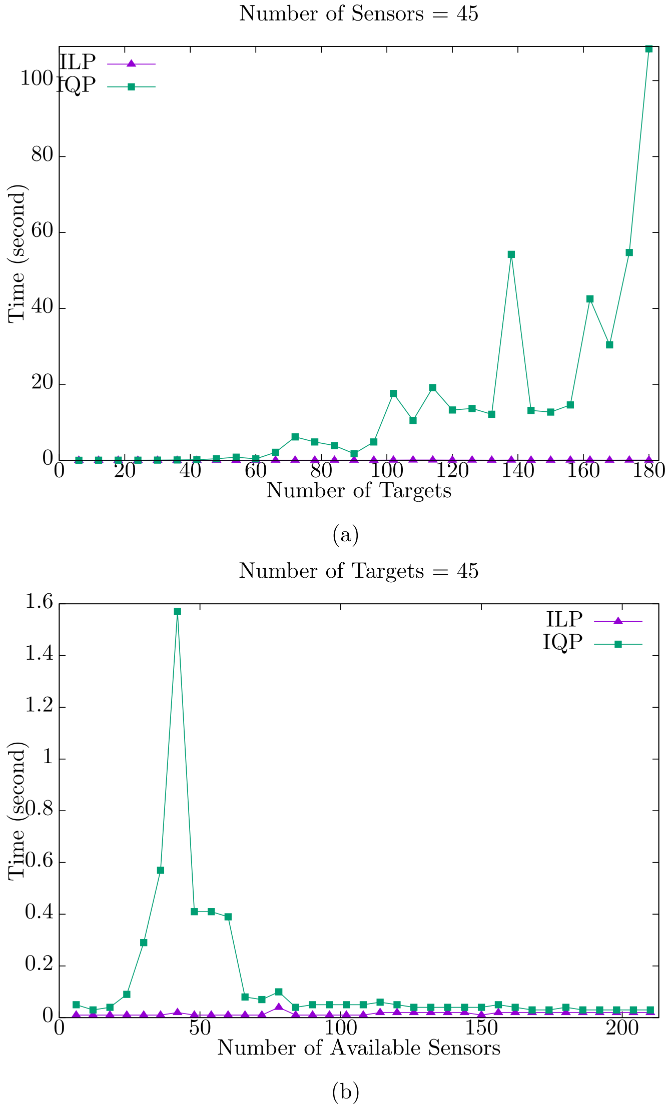

Fig. 19. Computational time comparison of ILP and IQP in large network with uniform sensor distribution. (a) keeping sensor number fixed at 45, increasing the number of targets from 6 to 180 (b) keeping target number fixed at 45, increasing sensor number from 6 to 210.

But when the network enters into under-provisioned state, all the sensor usage reaches saturation and further increase in targets does not change the shape. Also it is evident from Fig. 18(a) that all the approaches use almost same number of sensors in under-provisioned state.

When we increase the number of sensors (Fig. 18(b)) the number of activated sensors become fixed after a certain number of available sensors. At that point the number of sensors becomes good enough to meet all coverage or quality requirements and the network turns into over-provisioned system. As a result, further increase in sensor does not affect sensor usage.

7.7. Computation time analysis

As the network becomes more congested with sensor or target nodes, ILP and IQP formulations require more time to find the solution. Fig. 19 denotes the computation time of ILP and IQP approaches for large scale network with uniform sensor distribution.

In Fig. 19(a), number of targets is increased from 6 to 180 while keeping sensors fixed at 45. As the number of targets increases, the network enters into under-provisioned state and the coverage problem becomes more complex. Thus it requires more time to solve for IQP.

In Fig. 19(b), number of sensors is increased from 6 to 210 while keeping the number of targets fixed at 45. Here, the network gradually enters in over-provisioned from under-provisioned state after some point. As a result, initially IQP requires more time but after certain point, it takes fewer time.

From Fig. 19 it is clearly evident that IQP requires significant amount of time compared to ILP in under-provisioned system. Note that, the greedy algorithms require very insignificant amount of time compared to ILP and IQP.

Table 6
Summary of experimental results.

Approach	Result
IQP	Outperforms all the other approaches in terms of distance index
Greedy quadratic	Approximates the IQP but requires less time-complexity
ILP	Provides maximum total coverage
Modified greedy algorithm (Fusco and Gupta)	Approximates the ILP
Greedy-Razali	Uses lower number of sensors and consumes lower power than IQP. But it performs worse than IQP in terms of distance index.
pIQP	Provides higher distance index (i.e., higher coverage) in higher coverage group
Priority-based Greedy quadratic	Approximates the pIQP but requires less time-complexity
Reduced-variance IQP	Provides balanced coverage within each coverage group
Reduced-variance Greedy quadratic	Approximates the reduced-variance IQP but requires less time-complexity

7.8. Impact of network topology

Comparing the curves of large network (Figs. 15 and 16) with the respective small network curves (Figs. 9 and 10), it is evident that, our proposed approaches show similar result irrespective of the size of the network.

Similar argument can be drawn from comparison between uniform (Figs. 9 and 10) and zipf (Figs. 17 and 18) distribution.

Thus our proposed approaches do not depend on network topologies in terms of network size and node distribution.

All the simulation results are summarized in the Table 6.

8. Conclusion and future works

In this paper, heterogeneous target coverage problem for VSN, where targets have different coverage requirements, has been investigated. To this end, we propose integer linear/quadratic programming formulations (ILP, IQP) and corresponding greedy heuristics (greedy linear and greedy quadratic) that can be completed in polynomial time. IQP outperforms all the approaches in terms of fulfilling coverage requirements although its sensor usage and power consumption are higher than other approaches. In large networks, IQP is not suitable, hence its greedy version (i.e., greedy IQP) has been designed and it shows almost similar performance behavior of IQP. To tackle under-provisioned networks, we reformulate the problem as prioritized IQP, and reduced variance IQP. We also devise their corresponding greedy heuristics (priority based greedy quadratic and reduced variance greedy quadratic). Prioritized IQP is good at covering targets with higher coverage requirements and reduced variance IQP is suitable for coverage balancing within each coverage requirement group. In large networks, their corresponding greedy heuristics are recommendable as they show similar behavior of their respective formulations. In future we plan to explore the behavior of the formulations and heuristics under larger network settings. Also we want to extend our work for other two directions (*tilt* and *zoom*) of PTZ cameras. It will be good to see how the algorithms perform in mobile environments.

References

Ai, J., Abouzeid, A.A., 2006. Coverage by directional sensors in randomly deployed wireless sensor networks. J. Combin. Optim. 11 (1), 21–41.

Ammari, H.M., Das, S., 2010. A study of k-coverage and measures of connectivity in 3d wireless sensor networks. IEEE Trans. Comput. 59 (2), 243–257.

Bejerano, Y., 2008. Simple and efficient k-coverage verification without location information. In: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE, pp. 291–295.

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.

Cardei, M., Thai, M.T., Li, Y., Wu, W., 2005. Energy-efficient target coverage in wireless sensor networks. In: INFOCOM 2005. 24th Annual Joint Conference of the Ieee Computer and Communications Societies. Proceedings Ieee, vol. 3. IEEE, pp. 1976–1984.

Chaudhary, M., Pujari, A.K., 2009. Q-coverage problem in wireless sensor networks. In: International Conference on Distributed Computing and Networking. Springer, pp. 325–330.

Cheng, M.X., Ruan, L., Wu, W., 2005. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4. IEEE, pp. 2638–2645.

Du, X., Lin, F., 2005. Maintaining differentiated coverage in heterogeneous sensor networks. EURASIP J. Wirel. Commun. Netw. 2005 (4), 565–572.

Farzana, S., Papry, K.A., Rahman, A., Rab, R., 2016. Maximally pair-wise disjoint set covers for directional sensors in visual sensor networks. In: Wireless Days (WD), 2016. IEEE, pp. 1–7.

Feng, W.-C., Kaiser, E., Feng, W.C., Baillif, M.L., 2005. Panoptes: scalable low-power video sensor networking technologies. ACM Trans. Multimed Comput. Commun. Appl. (TOMM) 1 (2), 151–167.

Fusco, G., Gupta, H., 2009. Selection and orientation of directional sensors for coverage maximization. In: Sensor, Mesh and Ad Hoc Communications and Networks (SECON).

Gu, Y., Ji, Y., Li, J., Zhao, B., 2009. Qos-aware target coverage in wireless sensor networks. Wireless Commun. Mobile Comput. 9 (12), 1645–1659.

Hefeeda, M., Bagheri, M., 2007. Randomized k-coverage algorithms for dense sensor networks. In: INFOCOM.

IBM ILOG CPLEX Optimization Studio Community Edition, https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

Li, D., Liu, H., Lu, X., Chen, W., Du, H., 2012. Target q-coverage problem with bounded service delay in directional sensor networks. Int. J. Distributed Sens. Netw. 8 (10), 386093.

Lu, Z., Pitchford, T., Li, W., Wu, W., 2014. On the maximum directional target coverage problem in wireless sensor networks. In: Mobile Ad-hoc and Sensor Networks (MSN), 10th International Conference on. IEEE, pp. 74–79.

Lu, Z., Li, W.W., Pan, M., 2015. Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks. IEEE Trans. Veh. Technol. 64 (2), 714–727.

Malek, S.M.B., Sadik, M.M., Rahman, A., 2016. On balanced k-coverage in visual sensor networks. JNCA 72, 72–86.

Meguerdichian, S., Potkonjak, M., 2003. Low Power 0/1 Coverage and Scheduling Techniques in Sensor Networks. Tech. rep., UCLA Technical Reports 030001.

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 Pers. Commun. 79 (3), 2323–2338.

Rahman, A., Ahmed, M., Zerin, S., 2009. Analysis of minimum-energy path-preserving graphs for ad-hoc wireless networks. Simulation 85 (8), 511–523.

Razali, M.N., Salleh, S., Mohamadi, H., 2017. Solving priority-based target coverage problem in directional sensor networks with adjustable sensing ranges. Wireless Pers. Commun. 95 (2), 847–872.

Salleh, S., Marouf, S., Mohamadi, H., 2014. A new learning automata-based algorithm to the priority-based target coverage problem in directional sensor networks. In: International Conference on Mobile Networks and Management. Springer, pp. 219–229.

Sharmin, S., Nur, F.N., Razzaque, M.A., Rahman, M.M., 2016. Network lifetime aware coverage quality maximization for heterogeneous targets in dsns. In: Region 10 Conference (TENCON), 2016 IEEE. IEEE, pp. 3030–3033.

Sharmin, S., Nur, F.N., Razzaque, M.A., Rahman, M.M., Almogren, A., Hassan, M.M., 2017. Tradeoff between sensing quality and network lifetime for heterogeneous target coverage using directional sensor nodes. IEEE Access 5, 15490–15504.

Slijepcevic, S., Potkonjak, M., 2001. Power efficient organization of wireless sensor networks. In: Communications, 2001. ICC 2001. IEEE International Conference on, vol. 2. IEEE, pp. 472–476.

Tian, D., Georganas, N.D., 2002. A coverage-preserving node scheduling scheme for large wireless sensor networks. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02, ACM, New York, NY, USA, pp. 32–41, https://doi.org/10.1145/570738.570744.

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C., 2003. Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03, ACM, New York, NY, USA, pp. 28–39, https://doi.org/10.1145/958491.958496.

Wang, J., Niu, C., Shen, R., 2009. Priority-based target coverage in directional sensor networks using a genetic algorithm. Comput. Math. Appl. 57 (11), 1915–1922.

Yan, T., He, T., Stankovic, J.A., 2003. Differentiated surveillance for sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. ACM, pp. 51–62.

Yang, H., Li, D., Chen, H., 2010. Coverage quality based target-oriented scheduling in directional sensor networks. In: IEEE ICC, pp. 1–5.

L.-H. Yen, C. W. Yu, Y.-M. Cheng, Expected k-coverage in wireless sensor networks, Ad Hoc Netw. 4 (5).

Yetgin, H., Cheung, K.T.K., El-Hajjar, M., Hanzo, L.H., 2017. A survey of network lifetime maximization techniques in wireless sensor networks. IEEE Commun. Surv. Tutor. 19 (2), 828–854.

Zannat, H., Akter, T., Tasnim, M., Rahman, A., 2016. The coverage problem in visual sensor networks: a target oriented approach. J. Netw. Comput. Appl. 75, 1–15.

Zhang, H., Hou, J.C., 2005. Maintaining sensing coverage and connectivity in large sensor networks. Ad Hoc Sens. Wirel. Netw. 1 (1–2), 89–124.

Zhao, Q., Gurusamy, M., 2008. Lifetime maximization for connected target coverage in wireless sensor networks. IEEE/ACM Trans. Netw. (TON) 16 (6), 1378–1391.

Abdullah Al Zishan received his B.Sc. in Computer Science and Engineering from the Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 2017. He is currently pursuing his M.Sc. in CS department at University of Alberta, Edmonton, Canada. His research interests are in cyber-physical system, energy-informatics, optimization problem and machine-learning.

Imtiaz Karim is currently pursuing his Ph.D. at the department of CS in Purdue University, West Lafayette. He received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in 2017. His research interest span in the area of security and privacy. Currently, he is working on the security and privacy of cellular networks and IoT devices.

Sudipta Saha Shubha interestsis currently working as a Research Assistant in Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET). He received his B.Sc. in Computer Science and Engineering (CSE) from Bangladesh University of Engineering and Technology (BUET) in 2017. His research interests are in Big Data Analytics, Data Mining, Large Scale Graph Analysis (such as: Social Network), Large Scale Distributed Computing and Optimization Problems in Scalable Machine Learning. His current research works include Optimization Problems in Geo-Distributed System and Speech Recognition using Big Data.

Ashikur Rahman received his B.Sc. and M.Sc. degrees in Computer Science and Engineering from the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 1998 and 2001, respectively. He received his Ph.D. in 2006 from the Department of Computing Science, University of Alberta, Canada. After finishing his Ph.D., he worked as a postdoctoral researcher at Simon Fraser University (2007), University of Calgary (2010–2011), McGill University (2011), Canada and Binghamton University (2012), USA. His research interests include ad-hoc and sensor networks, peer-to-peer computing, swarm intelligence, data center networks, cyber physical systems, back-end compiler optimization and neural networks. He is currently working as a Professor in the Department of Computer Science and Engineering, BUET.