Maximally Pair-wise Disjoint Set Covers for Directional Sensors in Visual Sensor Networks

Shahla Farzana*, **Khaleda Akther Papry**†, **Ashikur Rahman**‡
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka
Email: *treena908@gmail.com, †papry.05084@gmail.com, ‡ashikur@cse.buet.ac.bd

Raqeebir Rab
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology, Dhaka
Email: jishan005@gmail.com

***Abstract*—Directional sensors in visual sensor networks (VSNs) spend most of their energy for two major tasks: *sensing* and *communication*. As sensing regions of directional sensors often overlap, the sensing task can be rotationally scheduled among different sensors. Thus, efficient scheduling of directional sensors became one of the fundamental interesting problems attracting many researchers over the past decade. One possible scheduling scheme can be achieved by partitioning the directional sensors into a maximal number of disjoint set covers each of which is capable of monitoring all the targets using minimal number of sensors. The derived set covers can be activated successively in a round robin fashion. As a result, the larger the number of disjoint sets, the longer the lifetime of the network. Moreover, *fault tolerance* can be easily achieved by activating more than one set covers at the same time. In this paper, we solve the problem of finding maximum possible disjoint set covers using minimal number of directional sensors. We formulate the problem as an Integer Linear Programming (ILP) problem. The ILP partitions the sensors into *optimum* number of disjoint set covers. As ILP does not scale well over large problem instances, we design a new target oriented heuristic and modify couple of existing heuristics for generating *near-optimal* disjoint set covers. We compare the performance of all three heuristics using extensive simulations.**

I. Introduction

A visual sensor network consists of a set of (smart) directional sensors capable of self-controlling their orientations in order to monitor a set of points of interest known as *target* points. Recently visual sensor networks have drawn considerable attention of researchers due to their enormous applicability in real-world scenarios like surveillance system, environment monitoring, smart traffic controlling system etc, to name a few.

Usually in visual sensor networks, smart mechanically operated PTZ (Pan, Tilt and Zoom) cameras are deployed as sensors. The user of a PTZ camera typically has the ability to control the pan (left and right), tilt (up and down) and zoom of the camera with a hardware device or software [7]. Usually large shopping malls and public areas would find a good use of PTZ cameras as using fixed cameras to monitor such large space would not be cost effective. Interestingly, one can program auto-tour functions into PTZ cameras in order to maneuver the camera in a predetermined way so that it can capture multiple points of interest. For instance, the camera can be programmed to capture the front door for few seconds, and then can be rotated to capture the parking lot or the back entrance for another few seconds and so on. This sequence might even be auto repeated as needed.

We envision two kinds of visual sensor networks: (i) over-provisioned networks, and, (ii) under-provisioned networks. We call a VSN is over-provisioned if the number of sensors is sufficient to cover all the targets and under-provisioned otherwise. Although, a directional sensor in over-provisioned VSNs (like PTZ cameras) has always directional coverage that can sense only in the direction of its orientation, it is highly likely that certain sensors often share common sensing regions. In such scenarios, some sensors can be switched into sleep state to conserve energy while other sensors sharing the same sensing area can keep on monitoring.

In this paper, we address the problem of finding maximum number of disjoint sets with minimum number of directional sensors (more applicable for over-provisioned networks) such that every set covers all the target points with known locations. We construct disjoint sets of sensor-pan pairs and schedule them in round robin fashion which minimizes power consumption, extends network lifetime and performs load balancing among the sensors. Moreover, generating multiple disjoint set covers improves fault tolerance as every set is capable of monitoring all the targets. For instance, we can activate k sets simultaneously and the system automatically provides k-coverage of each targets.

The major contributions of the paper is summarized below.

(i) We study the problem of directional senor coverage, in particular, coverage maximization problem using minimum number of sensors and find maximal possible disjoint set covers such that all sensors in a set can be activated to monitor all the targets and the number of sensor usage is minimal.

(ii) We develop Integer Linear Programming (ILP) formulation of the problem to better understand the optimal solution.

(iii) As the ILP is computationally expensive and does not scale well, we introduce a polynomial time *target-oriented* heuristic for finding approximate maximum disjoint set covers. The heuristic groups all of the deployed directional sensors into a number of (disjoint) subsets, each of which covers all of the targets.

(iv) We extend two other existing algorithms capable of generating a single set cover and augment those to iteratively generate multiple (disjoint) set covers.

(v) Finally, we compare the performance of the heuristics in terms of network lifetime, average power consumption and the number of generated disjoint set-covers.

II. Related Works

Over the past decade, a large number of researchers have been addressing fundamental issues of visual sensor networks such as the optimal placement of sensors, energy efficient scheduling to extend network lifetime [4], and fault tolerant network design to ensure secured connectivity of the network. Among these issues, covering all target objects using minimum number of sensors is of prime importance. Ai and Abouzeid [3] first tackled the issue by formulating the problem as an integer linear programming (ILP) that lead to finding an optimal solution to the problem. They also developed practically-implementable Centralized Greedy Algorithm (CGA) and its relaxed version Distributed Greedy Algorithm (DGA). Along the same direction, Munishwar and Abu-Ghazaleh [11] proposed Centralized Forced Algorithm (CFA) and its relaxed version Distributed Forced Algorithm (DFA) where they focused on ensuring maximum coverage. However, they did not attempt to minimize the number of sensor usage. Cardei et. al. [6] proposed a method to extend the sensor network life time by organizing the sensors into a maximal number of set covers that are activated successively. However, their work only considers *omni-directional* sensors. Ahn and Park [2] proposed mathematical formulation and heuristics for maximum set cover problem using the similar *omni-directional* sensors. In another work, Munishwar and Abu-Ghazaleh [10] proposed target oriented heuristic for coverage maximization where they cover the critical targets first and greedily choose a sensor which covers most of the targets along with the critical target. Ding et. al. [8] proposed algorithms for reducing power consumption rate in coverage by alternating the sensors between active and sleep mode. None of the above mentioned works addressed the issue of finding maximal pair-wise disjoint cover sets for directional sensors in VSN domain which is the main focus of our work.

III. System Overview, definitions and motivation

In this section, we present the system overview and describe the motivation of the problem dealt in the paper.

A. Directional Sensing Model

Different from conventional sensing models where an omnisensing area centers on the sensor, we employ a directional sensing model as shown in Fig. 1(a). The sensing model is a 2-D model where the sensing area of a sensor s_i is a sector denoted by 4-tuple (L, R_s, $\overrightarrow{V_{ij}}$,θ). For PTZ cameras, the sensing sector is known as *pan*. Here L is the position of the sensor in 2-D plane, $\overrightarrow{V_{ij}}$ is the unit vector which cuts the sensing sector into half which defines the direction it is looking at, and θ is the cutoff angle of the field of view (FoV) on both sides of $\overrightarrow{V_{ij}}$. R_s is the sensing range (FoV range). To make the model tractable, we assume that a directional sensor might have a finite set of non-overlapping sensing sectors (pans). For instance, in the example shown in Fig. 1(a), a directional sensor with FOV, $2\theta = \frac{\pi}{4}$ can pick eight mutually disjoint pans/orientations. These sectors can be combined to generate the full circular view of an omnidirectional sensor.

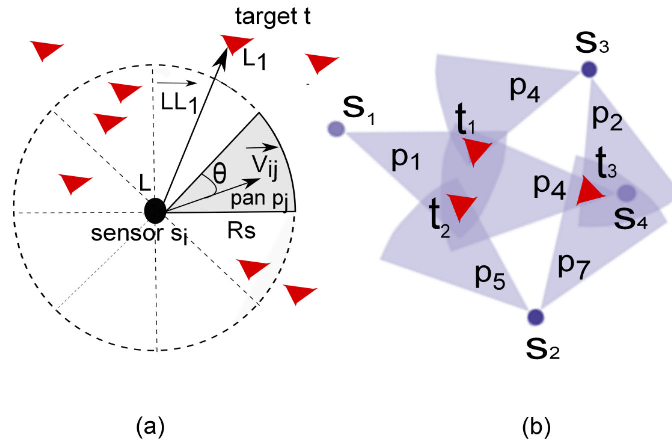


Fig. 1. (a) A directional sensor with a finite set of pans and the shaded area is the current sensing sector. A directional sensor can be active in one sensing sector at any time instant. (b) Example with three targets $T = \{t_1, t_2, t_3\}$ and four directional sensors $S = \{s_1, s_2, s_3, s_4\}$

A target point t at position L_1 is said to be covered by a sensor s_i centered at position L if and only if the following conditions are met:

1) d(L, L_1) $\leq R_s$, where d(L, L_1) is the Euclidean distance between the location of sensor (L) and target point (L_1).
2) The resulting distance vector $\overrightarrow{LL_1}$ is within the FOV $[-\theta, \theta]$ of the directional vector $\overrightarrow{V_{ij}}$

B. Definitions

A simpler way to judge if a target point t is covered by sensor s_i in pan p_j is as follows: if $\left|\overrightarrow{LL_1}\right| \leq R_s$ and $\overrightarrow{LL_1}.\overrightarrow{V_{ij}} \geq |\overrightarrow{LL_1}||\overrightarrow{V_{ij}}| \cos\theta$, then target t is covered. Running this test for all targets we can easily construct a coverage matrix $\mathcal{A}_{ij}^t$ such that each element of the matrix, a_{ij}^t, represents whether the sensor s_i in pan p_j can cover target t.

$$a_{ij}^t = \begin{cases} 1 & \text{if sensor-pan pair } (s_i, p_j) \text{ covers target } t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Definition 1: The set of targets covered by a specific sensor-pan pair (s_i, p_j) is denoted by Φ_{ij} and is defined by relation 2:

$$\Phi_{ij} = \{t | a_{ij}^t = 1\} \quad (2)$$

For example, in Fig. 1(b), $\Phi_{11} = \{t_1, t_2\}$ and $\Phi_{32} = \{t_3\}$.
Definition 2: critical target and sensors: The target covered by the least number of sensors is called *critical target*. The set of sensors covering the critical target is called *critical sensors*.
Definition 3: The set of sensor-pan pairs covering target t is denoted by $\Phi^{-1}(t)$ and is defined by relation 3:

$$\Phi^{-1}(t) = \{(s_i, p_j) | a_{ij}^t = 1\} \quad (3)$$

In Fig. 1(b), $\Phi^{-1}(t_1) = \{(s_1, p_1), (s_3, p_4), (s_4, p_4)\}$.
Definition 4: cardinality of a target: The number of sensor-pan pairs covering a target t is called cardinality $D(t)$ of the target t and is defined by relation 4:

$$D(t) = |\Phi^{-1}(t)| = |\{(s_i, p_j) | a_{ij}^t = 1\}| \quad (4)$$

Note that, with the definition 4, a critical target is the target with minimum cardinality.

C. Motivation

The major motivation of this work is to design the sensor scheduling mechanism so that the power consumption behavior and the fault tolerance capability of the network get improved. There are two ways to organize the sensors for rotational scheduling. In the first approach, we can organize the sensors in *pair-wise disjoint sets* so that each set is capable of monitoring all the targets. In the other approach, the sensors are allowed to participate in more than one sets while each set still being capable of monitoring all the targets independently. A research work based on the second approach for *omnidirectional* sensors are presented in [5]. However, we incorporate the first approach for *directional* sensors due to the following reasons. Consider the example scenario of Fig. 1(b). There are three targets $t_1, t_2,$ and t_3 and four sensors $s_1, s_2, s_3,$ and s_4 each having eight pans/sensing sectors $p_1, p_2, p_3, p_4, p_5, p_6, p_7,$ and p_8. The coverage relationship between sensors and targets can be represented as: $\Phi^{-1}(t_1) = \{(s_1, p_1), (s_3, p_4), (s_4, p_4)\}$, $\Phi^{-1}(t_2) = \{(s_1, p_1), (s_2, p_5), (s_4, p_4)\}$, and $\Phi^{-1}(t_3) = \{(s_2, p_7), (s_3, p_2), (s_4, p_4)\}$. In the first approach, it is possible to construct at most two disjoint sets covering all the targets and using the minimum number of sensors. These sets are $C_1 = \{(s_1, p_1), (s_2, p_7)\}$, and $C_2 = \{(s_4, p_4)\}$. In the second approach, which allows *sensor overlaps* between sets, three sets are possible, $C_1 = \{(s_1, p_1), (s_2, p_7)\}$, $C_2 = \{(s_1, p_1), (s_3, p_2)\}$, and $C_3 = \{(s_4, p_4)\}$. Now let us compare the power consumption behavior of two approaches. Suppose the system is homogeneous where all sensor nodes have the same energy level which allows each of them to be active for 1 unit of time consuming P unit of power. That is, if all sets are active for 1 unit of time successively, then the network lifetime (N) is equal to the number of set covers. Consequently, increasing number of set covers prolongs the network lifetime proportionally.

In the first approach, if both sets run for 1 unit of time successively, then the network lifetime becomes $N = 2$ consuming $\frac{|C_1|P+|C_2|P}{N} = \frac{3P}{N}$ unit of power.

In the second approach, sensor s_1 is included in two sets, each set being operational for different time intervals, e.g., $C_1 = \{(s_1, p_1), (s_2, p_7)\}$, for 0.5 unit of time, $C_2 = \{(s_1, p_1), (s_3, p_2)\}$ for 0.5 unit of time, and $C_4 = \{s_4\}$ for 1 unit of time. This will result in a network lifetime of 2 and the consumed power becomes: $\frac{|C_1|P\times 0.5+|C_2|P\times 0.5+|C_3|P}{N} = \frac{2\times P\times 0.5+2\times P\times 0.5+P}{N} = \frac{3P}{N}$ unit.

Although the network life time remains the same, the first approach is more advantageous when we consider the fault tolerance aspect. If the first approach generates K disjoint set covers, we can activate k sets simultaneously ($k \leq K$) and the system automatically provides k-coverage of each targets. For instance we can schedule C_1 and C_2 simultaneously and the system automatically provides 2-coverage.

IV. MDSC Problem Definition

In this section we formally define the problem that we solve in this paper. The problem is named as *maximal disjoint set cover* problem or MDSC in short and is defined as follows:
Given: Suppose we have a set of targets $T = \{t_1, t_2 ... t_m\}$, a set of homogeneous directional sensors $S = \{s_1, s_2 ... s_n\}$ each of which has a set of possible non-overlapping pans $P = \{p_1, p_2 ... p_q\}$. Suppose, each sensor s_i oriented in pan p_j covers subset of targets $\Phi_{ij} \subset T$. Since we are interested in monitoring all targets, a set cover C_k is defined as a set of sensor-pan pairs (s_i, p_j) such that all sensor-pan pairs together cover all the targets, i.e.,

$$\bigcup_{(s_i,p_j)\in C_k} \Phi_{ij} = T \tag{5}$$

Problem: The problem is to find maximal possible disjoint set covers such that all sensors in a set can be activated (and appropriately oriented) to monitor all the targets while minimizing the number of activated directional sensors.

V. Optimum solution of the MDSC Problem

In this section, we provide Integer Linear Programming (ILP) formulation of the MDSC problem which can be solved to generate maximum disjoint set covers using minimum number of sensors.

A. ILP Formulation

The parameters used for the formulation can be summarized as follows. n: the number of directional sensors; m: the number of targets, q: the number of orientations available for each directional sensor. Let us assume that $C = \{C_1, C_2, ..., C_r\}$ is the collection of all possible sets, each of which covers all the targets. $\mathcal{X}_{ij}^k$ is a variable that indicates whether sensor-pan pair (s_i, p_j) is in the set cover C_k. That is:

$$\mathcal{X}_{ij}^k = \begin{cases} 1 & \text{if sensor-pan pair } (s_i, p_j) \in C_k \\ 0 & otherwise \end{cases} \tag{6}$$

The variable in the ILP formulation is as follows.

$$\mathcal{Y}_k = \begin{cases} 1 & \text{if set cover } k \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Then the goal of ILP is to:
maximize:

$$\sum_{k=1}^{r}\sum_{i=1}^{n}\sum_{j=1}^{q}(1-\rho\mathcal{X}_{ij}^k)\mathcal{Y}_k \tag{8}$$

subject to:

$$\sum_{k=1}^{r}\sum_{j=1}^{q}\mathcal{X}_{ij}^k\mathcal{Y}_k \leq 1, i = 1...n; \tag{9}$$

$$\mathcal{Y}_k \in \{0, 1\}, k = 1...r \tag{10}$$

Remarks:

- The objective function in Equation 8 maximizes the number of disjoint sets, each covering all the targets and minimizes the number of sensors used in each set by imposing a penalty by multiplying the number of directional sensors to be activated by a positive penalty coefficient ρ whose value must be small enough ($\rho \leq 1$).

- There are r variables and (n +r) constraints for the ILP.
- Equation 9 imposes the condition that one directional sensor is used in at most one set cover.
- Equations 10 state the binary integer requirements on the variables.

VI. Centralized algorithms for the MDSC Problem

Although the ILP formulation mentioned in the previous section provides the optimal solution for this problem, it is not feasible for large problem instances. Therefore, we present heuristic based polynomial-time algorithms to approximate solution of the MDSC problem.

We consider a number of targets with known locations that need to be continuously observed. A large number of directional sensors each having finite pans/sensing sectors are randomly deployed close to the targets. The base station (BS) has the information about the coordinates of the targets. It also knows the positions of the directional sensors and their sensing range. Therefore, for each sensor it is able to compute it's target coverage matrix Sensed data might be processed locally by the sensors or at the BS, from where it is aggregated and forwarded to the user. We assume the sensors are equipped with some location determination capabilities (i.e., via GPS).

As we are considering an over provisioned system that is the number of sensors deployed in the field is generally larger than the optimum needed to perform the required task, an important energy-efficient method consists in scheduling the disjoint sets containing different directional sensors to alternate within *active* state, *idle* state and *sleep* state.

The set scheduling mechanism can be accomplished in the following steps: (i) sensors send their location information to the BS, (ii) BS executes one of the disjoint set maximization algorithms (described in the subsequent subsections) and broadcasts the schedule in which set each sensor belongs and when it needs to be active, (iii) every sensor schedules itself for active/idle/sleep intervals. For generating disjoint sets, BS executes one of the following three algorithms: (i) Iterative Centralized Greedy Algorithm (ICGA), (ii) Iterative Centralized Force-directed Algorithm (ICFA), and (iii) Iterative Target-Oriented Algorithm (ITOA).

A. Iterative Centralized Greedy Algorithm (ICGA)

ICGA is a greedy approach which iteratively constructs pair-wise disjoint set covers. Initially, all sensors are inactive (i.e., not selected). At each step, an inactive sensor-pan pair (s_i, p_j) is selected which maximizes the number of uncovered targets. Once a sensor-pan pair is selected, it is included in the current set cover and the newly selected sensor is excluded from the set of available sensors. All newly covered targets are removed from the target set. When there are no more targets to be covered, computation of a set is complete and a new set cover is formed in the same way. The iteration continues until no more sensors are available or no set covers can be created with the remaining available sensors covering all the targets. Algorithm 1 describes ICGA. The condition in line 4 ensures that the inner loop generates disjoint set covers until each target is covered by at least one sensor from the set of *SENSORS*.

Algorithm 1 Iterative Centralized Greedy Algorithm (ICGA)

Input: S= set of sensors; T= set of targets; P= set of discrete pans; K= No. of set covers
Output: C= Collection of Set Covers C_1, C_2....C_K
1: *SENSORS* $= S$
2: /**SENSORS* keeps track of the list of unused sensors */
3: $k = 0$
4: **while** *SENSORS* $\neq \emptyset$ **do**
5:     /* a new set cover C_k will be formed */
6:     $k = k + 1$
7:     C_k=$\emptyset$
8:     *TARGETS*= T
9:     /**TARGETS* contains the uncovered targets */
10:     **while** *TARGETS* $\neq \emptyset$ **do**
11:         /* selects (s_i, p_j) covering maximum targets */
12:         $(s_i, p_j) \leftarrow \arg\max_{s_i \in SENSORS,\ 1 \leq j \leq q} \Phi_{ij} \cap TARGETS$
13:         **if** $|\Phi_{ij} \cap TARGETS| = 0$ **then**
14:             **goto** marker
15:         **else**
16:         $C_k = C_k \cup \{(s_i, p_j)\}$
17:         *SENSORS* $=$ *SENSORS* $\setminus \{s_i\}$
18:         *TARGETS* $=$ *TARGETS* $\setminus \Phi_{ij}$
19:         **end if**
20:     **end while**
21:     $C = C \cup C_k$
22: **marker**: return the collection of set covers C

Let us analyze the time complexity of ICGA. Since the inner *while* loop (Line 10) runs over all the targets, it may require at most m iterations. In each iteration, Line 12 creates the coverage matrix which requires mnq operations to check over all sensor-pan pairs and all targets. Then to select the desired (s_i, p_j) that provides maximum coverage nq comparisons are needed. Thus, the worst case complexity of the inner *while* loop is $O((m + 1)mnq)$. The outer *while* loop runs over all $SENSORS$, so there can be at most n iterations. Consequently the worst case time complexity of ICGA is $O((m + 1)mn^2q)$ which eventually becomes $O(m^2n^2q)$.

B. Iterative Centralized Force-directed Algorithm (ICFA)

ICFA is based on CFA proposed in [11]. CFA introduces a concept of *force* and assigns weights to the pans depending on the force exerted by targets on them in order to determine the likelihood of a given pan to be selected. The force in a pan P_k for a camera C_i is computed as the ratio of number of targets coverable by P_k to the total number of targets coverable by the camera C_i. Mathematically:

$$F_{ik} = \frac{|\Phi_{ik}|}{|\Phi_i|} \tag{11}$$

Algorithm 2 Iterative Centralized Force-directed Algorithm (ICFA)

Input: S= set of sensors; T= set of targets; P= set of discrete pans; K= No. of set covers
Output: C= Collection of Set Covers C_1, C_2....C_K

1: *SENSORS*= S
2: /**SENSORS* keeps track of the list of unused sensors */
3: $k = 0$
4: **while** *SENSORS* $\neq \emptyset$ **do**
5: /* a new set cover C_k will be formed */
6: k= k + 1
7: C_k=ϕ
8: *TARGETS*= T
9: /**TARGETS* contains the targets that still have to be covered by the current set cover C_k */
10: **while** *TARGETS* $\neq \emptyset$ **do**
11: /* more targets have to be covered */
12: Calculate F_{uv} for each $s_u \in$ *SENSORS*, $p_v \in P$;
13: Let a sensor-pan pair (s_i, p_j) covers Φ_{ij} targets with force F'_{ij} such that:
14: $F'_{ij} = \text{argmax}_v F_{uv}; \forall s_u \in S; \forall p_v \in P; \forall s_i \in S; \forall p_j \in P;$
15: **if** $\left|F'_{ij}\right| = 0$ **then**
16: **goto** marker
17: **else**
18: $C_k = C_k \cup \{(s_i, p_j)\}$
19: *SENSORS* $=$ *SENSORS* $\setminus \{s_i\}$
20: *TARGETS* $=$ *TARGETS* $\setminus \{\Phi_{ij}\}$
21: **end if**
22: **end while**
23: $C = C \cup C_k$
24: **end while**
25: **marker**: return the collection of set covers C

where $|\Phi_{ik}|$ is the number of targets covered by camera C_i on pan P_k, and $|\Phi_i|$ is the number of coverable targets by camera C_i in all possible pans. CFA selects a camera-pan pair that has the highest force.

ICFA is presented in Algorithm 2. Inside inner *while* loop (Line 10) the pan with the highest *force* is always selected at each iteration. The selected camera and the covered targets are removed (like CGA) and the process continues until no more cameras can be selected or all the targets are covered. The outer *while* loop runs iteratively until no new disjoint set covers can be formed.

Let us analyze the time complexity of ICFA. The outer *while* loop is bounded by the number of sensors which is n, and the inner *while* loop is bounded by the number of targets which is m. The most costly step of the inner while loop is step 12 which needs to go over all targets for each camera-pan pair (out of $n \times q$ pairs) in order to calculate forces of all pan-pairs. So this step requires $O(mnq)$ checking. Therefore, the cost for inner while loop becomes $O(m^2nq)$. As the outer loop executes n times in worst case the time complexity of ICFA is $O(m^2n^2q)$.

Algorithm 3 Iterative Target-Oriented Algorithm (ITOA)

Input: S= set of cameras; T= set of targets; P= set of discrete pans; K= No. of set covers
Output: C=Collection of Set Covers C_1, C_2....C_K

1: *SENSORS*= S
2: /**SENSORS* keeps track of the list of unused sensors */
3: $k = 0$
4: **while** *SENSORS* $\neq \emptyset$ **do**
5: /* a new set cover C_k will be formed */
6: k= k + 1
7: C_k=$\emptyset$
8: *TARGETS*= T
9: /**TARGETS* contains the uncovered targets */
10: **while** *TARGETS* $\neq \emptyset$ **do**
11: /* more targets have to be covered */
12: Find $D_{min} = \text{min}D(t) : t \in T$
13: **if** $|D_{min}| = 0$ **then**
14: /**TARGETS* contains some uncoverable targets*/
15: **goto** marker
16: **else**
17: $S' = \emptyset$
18: Find the targets having cardinality equals D_{min} and create a set T'
19: **for each** $t \in T'$
20: **for each** $(s_i, p_j) \in \Phi^{-1}(t)$
21: $S' = S' \cup \{(s_i, p_j)\}$
22: Find the most contributing (s_u, p_v) such that:
23: $(s_u, p_v) \leftarrow argmax_{s_u \in S', 1 \leq v \leq q} \Phi_{uv} \cap$ *TARGETS*
24: /* selects (s_u, p_v) covering maximum targets */
25: In case of tie, create a subset $S'' \subseteq S'$ with all sensor-pan pairs having same contribution
26: Let $(s_i, p_j) \in S''$ has the maximum force F'_{ij} among all sensor-pan pairs in S'', i.e.:
27: $F'_{ij} = \text{argmax}_v F_{uv}; \forall s_u \in S; \forall p_v \in P; \forall s_i \in S; \forall p_j \in P;$
28: $C_k = C_k \cup \{(s_i, p_j)\}$
29: *SENSORS* $=$ *SENSORS* $\setminus \{s_i\}$
30: *TARGETS* $=$ *TARGETS* $\setminus \{\Phi_{ij}\}$
31: **end if**
32: **end while**
33: $C = C \cup C_k$
34: **end while**
35: **marker**: return the collection of set covers C

C. Iterative Target-Oriented Algorithm (ITOA)

While ICGA and ICFA looks into the *sensors* first and selects the most contributing sensor-pan pairs, ITOA looks into the *targets* instead. At each iteration, it finds the least covered target (the critical target) and selects the sensor-pan pair covering it. There could be more than one critical targets, so ITOA finds a set of critical targets T' having the same (minimum) cardinality and a set of sensor-pan pairs S' covering at least one of the critical targets in T'. Then, from S', ITOA selects the most contributing sensor (like CGA), i.e., the sensor having a pan that covers maximum targets. Unlike CGA, if there are more than one sensors having the same highest contribution, ITOA chooses the one with highest force (i.e., highest coverage ratio) as in CFA. We continue this approach until each target is covered by at least one sensor generating a disjoint set cover. After this iteration all the selected sensors are eliminated from the sensor's set (S) and all the targets are restored to the set of targets (T). The same procedure is run iteratively to generate all possible disjoint set covers. The pseudo-code of ITOA is shown in Algorithm 3.

Let us see the time complexity of ITOA. The outer *while* loop is bounded by the number of sensors which is n, and the inner *while* loop is bounded by the number of targets which is m. Step 18 of the inner while loop incurs most cost because it needs to go over all the targets for each camera-pan pair (out of $n \times q$ pairs) for finding target(s) with minimum cardinality. So this step requires $O(mnq)$ operations. Therefore, the cost for inner while loop becomes $O(m^2nq)$. As the outer loop executes n times in worst case the time complexity of ITOA is $O(m^2n^2q)$.

VII. Performance Analysis

In this section, we compare the performance of the solutions generated by ILP formulation and three proposed heuristics with extensive simulation results.

A. Simulation Environment

We consider a stationary network consisting of sensors and target points randomly deployed over an area of 100×100 units. The positions of sensors and target points are defined as 2-D points (x, y) in Cartesian co-ordinate system. We assume that all sensors in the network have equal sensing range (i.e., a homogeneous system). In the experiments the following parameters are tuned to observe the performance:

- n, the number of sensors. n is varied between 10 and 80 to observe the effect of sensor density on performance.
- m, the number of target points to be covered. m is kept fixed to 10 which are randomly deployed over the region.
- R_s, the sensing range. R_s is varied between 15 to 30 unit with the number of discrete non-overlapping pans $q = 8$.

The simulation code base was developed in Java programming language using JDK 1.7, NetBeans IDE 7.2. ILP formulation was solved using the LP library *lpsolve-v5.5.2.0* [1].

Scenario generation. The static networks are generated by varying the number of randomly deployed sensors (n) over the fixed deployment area of 100×100 units. n varies from 10 to 80 from one scenario to another with an increment of 5 sensors. Additional sensors were added in a manner so that the smaller scenario is the subset of the larger one. This makes the evaluation of the effect of increased population of sensors *consistent* as it retains all the "features" of the smaller scenario while making it better.

We also vary the sensing range between $R_s = 15$ to $R_s = 30$ unit with an increment of 2 unit at each step keeping the number of pans constant to 8 to evaluate the effect of increasing coverage area of the directional sensors. The number of targets is kept constant ($m = 10$) while changing targets' positions randomly over the deployment area.

For a certain value of n, R_s and m a VSN is randomly created and the optimal solution along with the proposed heuristics are simulated over this VSN. For each VSN size, we have generated 400 instances over which the simulations are performed. Performance measures are reported as an average of these 400 random scenarios (unless specified otherwise). We have followed two different approaches to evaluate the performance during simulation:–(i) by varying n while keeping m and R_s constant, and (ii) by varying R_s while keeping n and m constant.

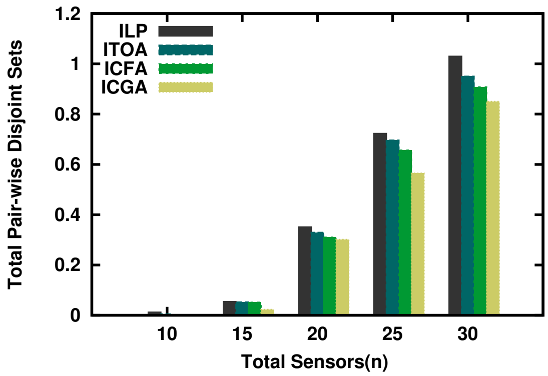
B. Evaluation Criteria

We analyze the performance of the proposed heuristics over three evaluation criteria:–(i) network lifetime, (ii) fault tolerance, and (iii) power consumptions that we describe next.

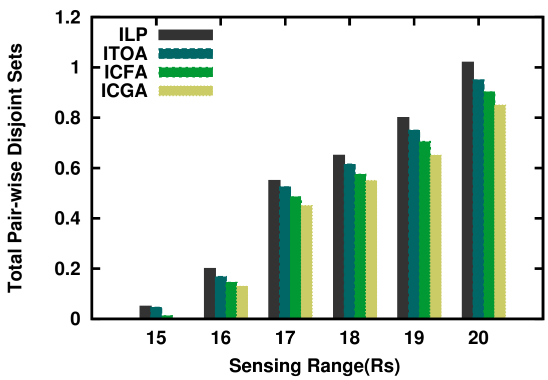
Network Lifetime Analysis (N). We consider homogeneous sensors and all of them have same residual energy. We also assume that available energy keeps sensors alive for t time unit. That is, if all sensors are active at the same time continuously, then the network lifetime becomes $N = t$. As the sensors are organized in disjoint set covers, maximizing number disjoint set covers extend network lifetime proportionately. Thus, if we have K disjoint set covers and we schedule them successively each set being active for t time unit, the network lifetime becomes $N = tK$. In other words, network lifetime, $N \propto K$.

Fault Tolerance Analysis. With the increase in number of pair-wise disjoint set covers K, fault tolerance of the network also improves. If any sensor malfunctions, it affects only the set cover containing that sensor. All other set covers are still good for monitoring all target points. Moreover, by generating K disjoint set covers we can achieve k'-coverage ($k' \leq K$) by scheduling k' sets simultaneously and each each target point becomes automatically covered by at least k' sensors. Thus, we can activate $\lfloor \frac{K}{k'} \rfloor$ different sets ensuring k'-coverage. For example, our solution gives K sets with 1-coverage and $\lfloor \frac{K}{2} \rfloor$ sets with 2-coverage.

Power Consumption Analysis (P_w). We assume that a visual sensor can be in one of the three states: *active*, *idle*, and *sleep*. In active state, the sensor captures the image in cooperation with other hardware components and consumes most power. In idle state, the sensor is still turned ON but no specific task is performed; it simply runs background OS processes. The sleep state is the least power consumption state where all hardware components are deactivated. As presented in [9], a Panoptes video sensor node consumes $p_i = 1.473W$ in idle state, $p_a = 5.268W$ in active state and $p_s = 0.058W$ while in sleep mode. Further we can divide the active power into two parts. For camera and network activation, power consumption is about $p_{cn} = 4.28W$ and while capture the scenario, it takes about $p_{cr} = (5.268 - 4.28) = 0.988W$. Thus, total active power is $p_a = (p_{cn} + p_{cr})W$. For certain values of p_a, p_i and p_s, the equation for consumed power (P_w) of a VSN can be derived as follows.

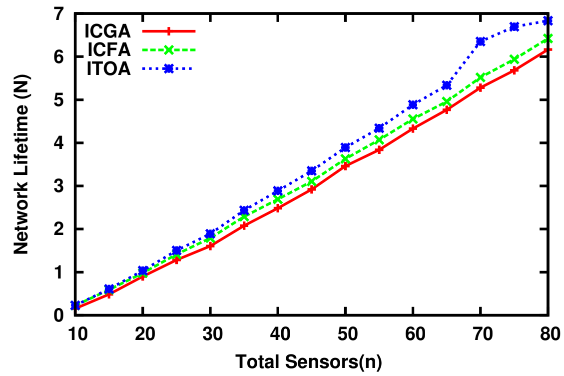


(a) $R_s = 20, q = 4$

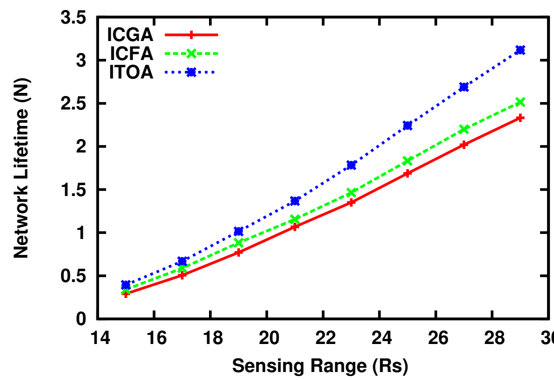


(b) $n = 30, q = 4$

Fig. 2. The average number of disjoint solutions' comparison between ILP, ITOA, ICFA and ICGA with, (a) $R_s = 20$, and (b) $n = 30$



(a) $R_s = 30, q = 8$



(b) $n = 40, q = 8$

Fig. 3. Network Lifetime (a) by varying the number of sensors, (b) by varying the sensing range

Suppose, the generated set of disjoint set covers $C = \{C_1, C_2, \ldots, C_K\}$ is scheduled for $\mathcal{T}$ seconds for monitoring all targets and $|C| = K$. The number of sensors in a cover set is $|C_i|$. So, the total number of sensors used in C is $\sum_{i=1}^{K} |C_i|$.

As we schedule all the set covers successively in a round robin fashion, every set cover will remain active for $\frac{\mathcal{T}}{K}$ seconds and in sleep state for $\frac{\mathcal{T}(K-1)}{K}$ seconds. So, total power consumption in active mode is $P_A = \frac{\sum_{i=1}^{K} |C_i| p_a}{K}$ unit. Similarly, in sleep mode, the total power consumption, $P_S = \frac{(K-1)\sum_{i=1}^{K}(|C_i|)p_s}{K}$ unit. All unused sensors are in idle state and deactivated for $\mathcal{T}$ seconds. Therefore, the total power consumption in idle state becomes $P_I = (n - \sum_{i=1}^{K} |C_i|)p_i$ unit. Adding all three components, the total power consumption of the network becomes, $P_w = (P_A + P_S + P_I)$ unit.

C. Simulation Results

In the first set of experiments, we compare the average number of disjoint sets (K) generated by the heuristics with the optimum number produced by ILP for the same scenario. Fig. 2 shows the comparison at a glance. In Fig. 2(a) we vary n between 10 and 30 while keeping $R_s = 20$, the number of pans, $q = 4$ and $m = 10$ fixed. On the other hand, in Fig. 2(b) we vary R_s from 15 to 20 units while keeping $n = 30$, $q = 4$ and $m = 10$ constant. The y-axis shows the value of K which is an average of 20 random instances of VSNs. Although ILP maximizes number of disjoint sets and outperforms all three heuristics, the number of sets generated by ITOA, ICFA, and ICGA is very close to the number generated by ILP. ITOA shows better performance compared to the other two heuristics and ICGA shows worst performance. Although ILP maximizes the number of set covers, its run-time complexity is much higher compared to three heuristics. This is due to the exponential nature of ILP. In Table I, we compare the actual running time of ILP, ITOA, ICFA and ICGA for the same scenarios. As we increase sensing range, each sensor covers more and more targets in every pan and the problem instance becomes larger. The run time for finding optimal solution (ILP) exponentially increases with the problem size whereas the run time of the proposed heuristics are much more scalable.

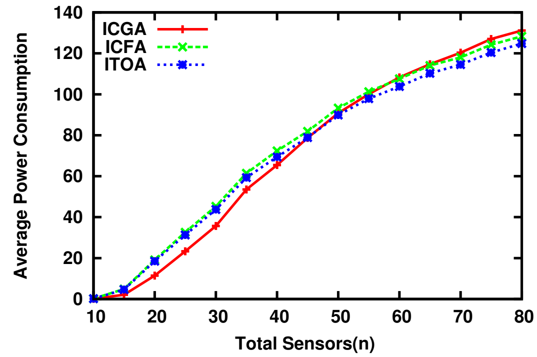
In Fig. 3, we plot the network life time (N) which is proportional to the number of average disjoint sets. Fig. 3(a) varies the number of sensors and Fig. 3(b) varies the sensing range. Clearly the network life time increases with the sensor density and the sensing range. When there are more sensors than the targets the probability of generating more cover sets also increases. Therefore, with the increase in sensor density the system becomes more and more over-provisioned and generates more cover sets. Similarly, by extending sensing range we can increase the coverage capability of the sensors; consequently the number of disjoint solutions also increases. In both figures, ITOA outperforms ICFA and ICGA. As ITOA covers the critical targets first and considers more contributing sensors while generating the set covers, it generates more cover

TABLE I
RUN TIME OF ILP, ITOA, ICFA AND ICGA

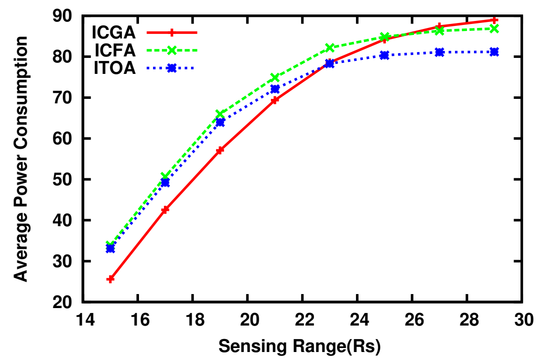
Range	ILP		ITOA		ICFA		ICGA	
	Life time	Run time (ms)	Life time	Run time (ms)	Life time	Run time (ms)	Life time	Run time (ms)
15	0.050	245.1	.045	2.35	.012	0.75	0.0	1.1
16	0.202	330.0	.165	2.40	.145	0.8	0.132	1.4
17	0.55	11342.5	.525	2.45	.485	0.8	0.45	1.8
18	0.651	16352.6	.615	4.65	.579	0.85	0.55	2.4
19	0.805	33548.0	.75	6.36	.70	1.65	0.65	3.9
20	1.02	46467.2	.95	7.0	.90	2.35	0.85	4.7

sets. On the other hand, as ICGA chooses sensors greedily the sensor(s) covering the critical targets might get used up in the earlier iterations, thereby lowering the number of cover sets.

Fig. 4 shows the average power consumption of all three heuristics. Fig. 4(a) varies the number of sensors and Fig. 4(b) varies the sensing range. With the increase in R_s (the FoV range), the lens of the PTZ camera is focused at a distance yielding the increase in depth-of-field of FoV which incurs more power to be consumed by the sensors. We assume that the power consumption model for active sensors increases quadratically with R_s that is, $p_{cr} \propto R_s^2$. This is a reasonable assumption because the coverage area within a pan of a sensor is a sector of a circular region (i.e., $\frac{\pi r^2}{q}$) although any other power model can be used. In both figures ICGA initially consumes least amount of power compared to the other approaches because, it generates less disjoint set covers and ends up using lower number of sensors compared to the other two approaches. But gradually its power consumption increases as its ratio of sensor usage increases but the number of solution sets does not increase in that proportion. On the other hand, ITOA maximizes disjoint set covers while minimizing the number of sensor usage in each set. So, when sensor density is high (i.e., many redundant sensors exist) it consumes lower power compared to ICFA and ICGA.



(a) $R_s = 20, q = 8$



(b) $n = 40, q = 8$

Fig. 4. Power consumptions, (a) by varying the number of sensors, (b) by varying the sensing range

VIII. CONCLUSION

In this paper, we study the coverage problem of randomly deployed directional sensors in VSNs. We intend to construct energy efficient fault tolerant solutions for target coverage problem. At first we present an exact solution to this problem using Integer Linear Programming (ILP) where we add necessary constraints to minimize power consumption using minimum number of sensors. Then, we present three iterative practically implementable heuristics for finding a set of disjoint set covers that individually covers all the targets. Finally, we compared performance of proposed solutions on the basis of few important performance metrics. The simulation result reveals the fact that for a certain number of targets, the average disjoint solution that is network lifetime of ILP and all three heuristics increases with the number of sensors and the sensing range. There is a trade-off between the higher lifetime value and the increase in the runtime of ILP for large VSNs. Compared to the optimal solution, our proposed heuristics are more scalable to large VSNs.

REFERENCES

[1] LP solver. http://web.mit.edu/lpsolve/doc/Java/README.html. [Online; accessed 10-10-2015].

[2] N. Ahn and S. Park. A new mathematical formulation and a heuristic for the maximum disjoint set covers problem to improve the lifetime of the wireless sensor network. *Ad Hoc & Sensor Wireless Nets.*, 13(3-4):209 – 225, 2011.

[3] J. Ai and A. A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11(1):21–41, 2006.

[4] W. Cai, Y.and Lou, M. Li, and X. Li. Target-oriented scheduling in directional sensor networks. In *Proceedings of IEEE INFOCOM*, pages 1550–1558, May 2007.

[5] M. Cardei, M. T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, pages 1976–1984, March 2005.

[6] M. Cardei and J. Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Computer Communications*, 29(4):413 – 420, 2006.

[7] Cdr Jacob Koottummel. PTZ Camera Deployments The Right and the Wrong? https://www.linkedin.com/pulse/ptz-camera-deployments-yes-cdr-jacob-koottummel.

[8] L. Ding, W. Wu, J. Willson, J. Wu, Z. Lu, and W. Lee. Constant-approximation for target coverage problem in wireless sensor networks. In *Proceedings of IEEE INFOCOM, FL, USA*, pages 1584–1592, March 2012.

[9] Wu-Chi Feng, Ed Kaiser, Wu Chang Feng, and Mikael Le Baillif. Panoptes: Scalable low-power video sensor networking technologies. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(2):151–167, May 2005.

[10] V. Munishwar and N. B. Abu-Ghazaleh. Target-oriented coverage maximization in visual sensor networks. In *Proceedings of MobiWac '11*, pages 175–178, 2011.

[11] V. Munishwar and N. B. Abu-Ghazaleh. Coverage algorithms for visual sensor networks. *ACM Transactions on Sensor Networks. (TOSN)*, 9(4):45, 2013.