Coverage Problem for Sensors Embedded in Temperature Sensitive Environments

Arunabha Sen*, Nibedita Das*, Ling Zhou*, Bao Shen*, Sudheendra Murthy* and Prajesh Bhattacharya†
*Department of Computer Science and Engineering
Arizona State University, Arizona 85281
Email: {asen, nibedita.maulik, ling.zhou, bao, sudhi}@asu.edu
†G.W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology, Georgia 30332
Email: prajesh.bhattacharya@me.gatech.edu

***Abstract*— The coverage and connectivity problem in sensor networks has received significant attention of the research community in the recent years. In this paper, we study this problem for sensors deployed in temperature sensitive environments. This paper is motivated by the issues encountered during deployment of bio-sensors in a human/animal body. Radio transmitters during operation dissipate energy and raise the temperature of its surroundings. A temperature sensitive environment like the human body can tolerate such increase in temperature only up to a certain threshold value, beyond which serious injury may occur. To avoid such injuries, the sensor placement must be carried out in a way that ensures the surrounding temperature to remain within the threshold. Using a thermal model for heat distribution from multiple heat sources (radio transmitters), we observed that if the sensor nodes are placed sufficiently apart from each other, then the temperature of the surrounding area does not exceed the threshold. This minimum separation distance constraint gives rise to a new version of the sensor coverage problem that has not been studied earlier. We prove that both the optimization version and the feasibility version of the new problem are NP-complete. We further show that an ϵ-approximation algorithm for the problem cannot exist unless P = NP. We provide two heuristic solutions for the problem and evaluate the efficacy of these solutions by comparing their performances against the optimal solution. The simulation results show that our heuristic solutions almost always find near optimal solution in a fraction of the time needed to find the optimal solution. Finally, an algorithm for forming a connected sensor network with minimum transmission power in such a scenario is provided.**

I. Introduction

A biomedical sensor is a device, which is implanted in a human or animal body to monitor and transmit biological information such as retinal pressure [1], oxygen level on the surface of exteriorized tissues [2] etc. Recently, there has been an immense interest in new sensing, monitoring, wearable wireless devices and sensor networks for healthcare and clinical applications. This increased interest and importance of the emerging field is demonstrated by the successful conclusion of the third IEEE International Workshop on Wearable and Implantable Body Sensor Networks [3]. Among many other applications of biomedical sensor networks, the one that pertains to *artificial retina* [4] deserves special attention. The authors in [5] note that "organs that are especially sensitive to any temperature increase due to a lack of blood flow to them are prone to thermal damage (e.g., lens cataracts)". The authors in [6] also emphasize on the importance of considering possible health hazards for individuals exposed to EM field and identify the EM field values that is safe for human body. The authors in [1] note that heat build-up from the sensor electronics can jeopardize the implantation of the sensor, as elevated temperature may cause infection, especially when the implanted sensor becomes a haven for bacteria.

An example of a bio-medical sensor network currently used in clinical situations [7] is shown in figure 1. As seen in the figure, the geodesic Sensornet has a large number of wires connecting the sensors to the controller. This situation is clearly unwieldy. Efforts are currently underway to replace the wired sensor network by a wireless one in many universities and research laboratories. The formation of a network with implanted sensors on human or animal body poses a number of challenges. Firstly, the biomedical sensors cannot be implanted in any arbitrary location of the body. The placement of the sensors has to be confined within a set of potential locations. Secondly, the placement must be done in such a way that the increase in temperature due to the operation of the sensors (radio transmitters) is within an acceptable limit. Although such implantation poses many challenges, the researchers in our bioengineering department have already implanted such sensors in monkey brain. It is anticipated that such implantations will become fairly regular in the next few years.

Although the coverage and connectivity problems in sensor networks have received considerable attention from the research community in recent years [8]–[13], to the best of our knowledge, sensor placement and coverage problems for a temperature sensitive environment have not been studied earlier. Clearly, the area surrounding the location of a sensor will observe an increase in temperature due to the operation of the sensor and its radio transmitter. However, increase in temperature in such sensitive areas cannot be allowed to exceed a specified threshold. The introduction of this thermal constraint makes the coverage problem alone considerably more complex than similar problems studied in [8], [10], [12].

In the sensor network literature, there exists two different versions of the coverage problem [11]. In the *region-coverage* version, the problem is to find the optimal placement of the sensors so that the given region(s) of interest can be sensed. On the other hand in the *point-coverage* version, the problem is to find the optimal placement of the sensors so that a set of pre-specified points can be sensed. The version of the coverage problem discussed in this paper is different from either the region-coverage or the point-coverage problem. New constraints are imposed due to the fact that (i) the sensors cannot be placed in any arbitrary location if the sensing region happens to be the human/animal body and (ii) the sensors cannot be placed very close to each other as the increase in temperature due to their joint operation may exceed the acceptable threshold temperature and cause thermal impairment.

Fowler *et. al.* in [14] showed that sensor coverage problems are NP-complete. Hochbaum in [15] developed approximation schemes for these NP-complete problems. Coverage and connectivity problem was studied in an integrated fashion in [10], [12], [13]. The authors in [12] presented a Coverage Configuration Protocol that provides different degrees of coverage depending on the needs of the applications and explored the relationship between coverage and connectivity. Abrams *et. al.* in [8] developed a strategy for energy efficient monitoring of wireless sensor networks.

In this paper, we present our findings on the placement and coverage problem for medical bio-sensors implanted in human/animal body. We thoroughly analyzed the heat distribution phenomenon in a temperature sensitive environment like human body. Our analysis indicates that even if the temperature increase due to the operation of an isolated sensor remains below the allowable threshold, the surrounding temperature can still exceed the threshold if multiple sensors operate at close proximity of one another. However, if the sensors are placed sufficiently apart from each other (beyond a critical separation distance), then the temperature increase will remain below the threshold. Due to the page limitations and the focus of this conference, detailed discussion on the heat transfer process in the human body is not included in this paper. Interested readers are referred to our report [16]. The main conclusion of our heat transfer analysis is the following: *There exists a critical inter-sensor distance, d_{cr}, such that if the distance between any two deployed sensors is less than d_{cr}, then the temperature in the vicinity of the sensors exceeds the maximum allowable temperature $T_{threshold}$. Therefore, attention must be paid during sensor deployment to ensure that the distance between any two sensors is at least as large as d_{cr}.*

Fig. 1. Geodesic Sensor Network for measurement of EEG

This conclusion from our thermal analysis leads to the formulation of a new version of the sensor coverage problem. In this version, one would like to find out the fewest number of sensors that have to be placed in the region of interest, such that (i) the entire region of interest is sensed and (ii) the distance between any two sensors is at least as large as the critical separation distance. The introduction of the requirement (ii) adds a new dimension to the sensor coverage problem resulting in additional complexity.

The contributions of this paper are as follows:

1) We introduce a new version of the placement and coverage problem for sensors in a temperature sensitive environment.
2) We prove that both the optimization version and the feasibility version of the new problem are NP-complete.
3) We show that an ϵ-approximation algorithm for the problem cannot exist unless P = NP.
4) We provide an integer linear programming formulation for the optimal solution of the problem.
5) We provide two heuristic solutions for the problem and evaluate the efficacy of these solutions by comparing their performances against the optimal solution. The simulation results show that our

heuristic solutions almost always find near optimal solution in a fraction of the time needed to find the optimal solution.

6) We also propose an algorithm to maintain a connected network with the required coverage in such temperature sensitive environments.

Before proceeding further into the details of our work, we would like to draw the reader's attention on three points regarding the results presented in this paper. Firstly, in our simulation experiments, we have assumed a circular sensing region associated with each sensor. However, the heuristic solutions proposed in this paper do not assume the sensing region to be circular and can be used with any irregular shaped sensing region. Secondly, although the problem studied in this paper was motivated by an application of sensors in a temperature sensitive environment, our results are equally applicable in other domains that have a requirement for a minimum separation distance between any sensor node pair. Finally, we point out that due to the minimum separation distance constraint, performance bound of the form (O(log N)) given in [10] for the connected sensor cover problem cannot be obtained for this problem unless P = NP.

II. Problem Formulation for Sensor Coverage

In the sensor placement and coverage problem discussed in this paper, we have (i) a set of locations (or points p_i) to be sensed, (i) a set of potential locations (or points q_i) for placements of sensors and (iii) a minimum separation distance (d_{cr}) between each pair of sensors. The goal is to deploy as few sensors as possible in potential placement locations such that all points p_i are sensed and no two sensors are closer than d_{cr}. Like most of the previous studies in this area [9], [11], [13], we assume that each sensor is capable of sensing a circular area (disk) of radius r_{sen} with the location of the sensor being the center of the circle. We assume that the sensing radius of all the sensors are identical. An example of the coverage problem is shown in figure 2, where the circular nodes represent the points to be sensed (blue points), the square nodes represent the potential locations of the sensors (green points) and circles represent the area sensed by the sensor located at the center of the circle. A line connecting two square nodes indicates that the distance between them is less than the minimum separation distance d_{cr} and a sensor can be deployed in at most one of these two locations. A solution to the problem consists of selecting the locations b, c, d, f, g for sensor deployment whose coverage is shown by the shaded region in the figure. It may be noted that no two nodes in this set have a line between them, indicating that the distance between them is at least as large as d_{cr}.

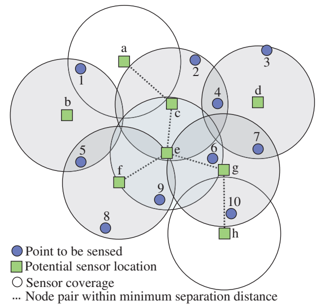

Fig. 2. Potential sensor locations, sensing points and sensing disk

Formally, the decision version of our sensor coverage problem can be stated as follows:

Problem 1: Sensor Coverage Problem (SenCov)

INSTANCE: Given (i) a set of points $P = \{p_1, \dots, p_m\}$ to be sensed (we will refer to these points as *blue* points)
(ii) a set of points $Q = \{q_1, \dots, q_n\}$, the potential placement locations of the sensors (we will refer to these points as *green* points)
(iii) sensing radius r_{sen} of the sensors
(iv) critical separation distance d_{cr}
(v) an integer K
QUESTION: Is there a subset $Q' \subseteq Q$, such that
(i) $|Q'| \leq K$,
(ii) $\forall p_i \in P$, $\exists q_j \in Q'$ such that $dist(p_i, q_j) \leq r_{sen}$,
(iii) $\forall q_i, q_j \in Q'$, $dist(q_i, q_j) \geq d_{cr}$
where $dist(x, y)$ represents the Euclidean distance between the points x and y.

In the optimization version of the problem, the input does not have the parameter K and the objective is to find the smallest K such that the conditions (ii) and (iii) are satisfied.

A. Sensor Coverage as Generalized Set Cover Problem

It may be noted that the sensor coverage problem described above can be viewed as a generalized version of the *set cover* problem as well as the *independent set* problem [17]. These two problems have been studied extensively. However, to the best of our knowledge, the generalized version of the problems as presented in this paper have not been studied before.

The sensor coverage problem can be transformed to the generalized set coverage in the following way. Each blue point p_i in the instance of the sensor coverage problem can be viewed as the element s_i to be covered in the instance of the set cover problem. Corresponding to green point q_j in the instance of the the sensor coverage problem, we can construct a subset S_j in the instance of the set cover problem. The set S_j will comprise of the elements s_i, if the $dist(p_i, q_j) \leq r_{sen}$. The generalized version of the set cover introduced in this paper also has a notion of a *a collection of incompatible subsets*. Two subsets S_i and S_j are said to be *incompatible*, if the Euclidean distance between the corresponding points q_i and q_j is less than the critical separation distance d_{cr}. Otherwise, the subsets S_i and S_j are *compatible*.

The generalized set cover problem is specified as follows:

Problem 2: Generalized Set Cover Problem (GSC)
INSTANCE: (i) a set of elements $S = \{s_1, \ldots, s_m\}$,
(ii) a collection of subsets $S_i \subseteq S, \mathcal{S} = \{S_1, \ldots, S_n\}$,
(iii) a collection of tuples of incompatible subsets $INC = \{(S_{i_1}, S_{j_1}), (S_{i_2}, S_{j_2}), \ldots, (S_{i_t}, S_{j_t})\}$
(iv) an integer K
QUESTION: Is there a subset $\mathcal{S}' \subseteq \mathcal{S}$, such that
(i) $|\mathcal{S}'| \leq K$
(ii) $\forall s_i \in S, \exists S_j \in \mathcal{S}'$, such that $s_i \in S_j$,
(iii) if $S_i, S_j \in \mathcal{S}'$ then $(S_i, S_j) \notin INC$.

B. Sensor Coverage as Generalized Independent Set Problem

The Sensor Coverage problem can also be viewed as a generalization of the Independent Set problem [17].

The sensor coverage problem can be transformed to the generalized independent set problem in the following way. From an instance of the sensor coverage problem, we can construct a graph $G = (V, E)$ where each node in V represents a green point q_i and two nodes $v_i, v_j \in V$ have an edge between them if the distance between the corresponding points q_i and q_j is less than the critical separation distance d_{cr}. In addition, with each node $v_i \in V$, we associate a list of blue points p_j. A blue point p_j will be in the list associated the node v_i (representing a green point q_i), if and only if the Euclidean distance between the points p_j and q_i is less than the sensing radius r_{sen}. The generalized independent set problem is specified as follows:

Problem 3A: Generalized Independent Set Problem (GIS)
INSTANCE: (i) a graph $G = (V, E)$
(ii) a set of elements $A = \{a_1, \ldots, a_m\}$ and
(iii) a subset $A_i \subseteq A$ associated with each node $v_i \in V$. We will refer to the subset A_i as list associated with v_i and will denote it by $L(v_i)$. We assume $\cup_{v_i \in V} L(v_i) = A$.
(iv) an integer K.
QUESTION: Is there an independent set V' in the graph $G = (V, E)$ such that
(i) $|V'| \leq K$ and
(ii) $\cup_{v_i \in V'} L(v_i) = A$

C. Feasibility issue in GSC and GIS Problems

In section II-B, we formulated the SenCov problem as the GSC problem. In the optimization version of the GSC problem, the goal is to find the smallest subset $\mathcal{S}' \subseteq \mathcal{S}$, such that (i) $\forall s_i \in S, \exists S_k \in \mathcal{S}'$, such that $s_i \in S_k$ and (ii) if $S_i, S_j \in \mathcal{S}'$ then $(S_i, S_j) \notin INC$. Since the optimization version of the SC problem is NP-complete [17], and GSC reduces to SC when $INC = \emptyset$, we can conclude that the GSC is also NP-complete. This conclusion leads us to look for approximation algorithms for GSC with guaranteed performance bound, especially because it is well known that such approximation algorithms with guaranteed performance bound exists for SC [15]. It may be noted however that although the problems SC and GSC are very similar, they have a major difference. To illustrate the point, we introduce the *feasibility* version of the SC (SCF) and GSC (GSCF) problems. The SCF and GSCF problems are special cases of the SC and GSC problems respectively, when $K = \infty$. In a similar fashion, we can consider the *feasibility* version of the GIS (GISF) and SenCov (SenCovF) problems as special cases of the GIS and SenCov problems respectively, when $K = \infty$.

It may be noted that the SCF problem can be solved easily by including in the set $\mathcal{S}'$ all the elements of the set $\mathcal{S}$ and checking if it covers all the elements of the set S. However, GSCF cannot be solved in such a trivial fashion because of the incompatibility among the elements of the set $\mathcal{S}$. In fact, in the next section we show that the GISF problem (and equivalently the GSCF and SenCovF problems) is NP-complete.

The NP-completeness of GISF (together with equivalent GSCF and SenCovF) problem puts a brake on our attempt to develop an approximation algorithm for the GSC problem with a guaranteed performance bound. In order to avoid the problem, we introduce a modified version of the GIS problem (GISM) whose goal is to find the smallest independent set V', whose $|\cup_{v_i \in V'} L(v_i)|$ is the largest. Informally, the goal of the GISM problem is to find an independent set that covers the largest number of blue points (points to be sensed) with smallest number of green points (sensors). It may be noted that unlike the GISF problem, feasibility is not an issue for the GISM problem. Before we formally define the GISM problem, we introduce the notion of *cost* of an independent set V' as follows:
$C(V') = (|\cup_{v_i \in V} L(v_i)| - |\cup_{v_i \in V'} L(v_i)| + \alpha) \times |V'|$,
where α is a number much smaller than $1/n$. (note: V is the set of nodes in the graph with $|V| = n$, V' is an independent set. α is introduced to handle the case when $|\cup_{v_i \in V} L(v_i)| = |\cup_{v_i \in V'} L(v_i)|$)

Problem 3B: Generalized Independent Set Problem (Modified) (GISM)
INSTANCE: (i) a graph $G = (V, E), (|V| = n)$
(ii) a set of elements $A = \{a_1, \ldots, a_m\}$ and
(iii) a subset $A_i \subseteq A$ associated with each node $v_i \in V$
We will refer to the subset A_i as list associated with v_i and will denote it by $L(v_i)$. We assume $\cup_{v_i \in V} L(v_i) = A$. In the optimization version of the GISM problem we try to find an independent set V' whose cost $C(V)$ is the smallest and in the decision version of the problem, we ask if there exists an independent set V' whose cost $C(V)$ is less than 1. Formally,
QUESTION: Is there an independent set V' in the graph $G = (V, E)$ whose *cost* is less than 1 (i.e., $C(V') < 1$)?

III. Computational Complexity of the Sensor Coverage Problem

In this section, we first show that the GISF problem is NP-complete.

Theorem 1: The GISF problem is NP-complete.

Proof: We give a transformation from the 3-Satisfiability (3-SAT) problem [17]. It may be noted that an instance of the 3-SAT problem is made by a set of variables $U = \{u_1, \ldots, u_g\}$ and a set of clauses $C = \{c_1, \ldots, c_h\}$ and want to find out if there is a truth assignment to the variables in U such that all the clauses in the set C are satisfied?

From an instance of the 3-SAT problem, we generate an instance of the feasibility version of the GISF problem and show that the GISF problem has a feasible solution, if and only if the instance of the 3-SAT problem is satisfiable. The graph in the instance of GISF is constructed in three phases.

Phase I: For each variable $u_i \in U$, we construct part of the graph G by introducing two nodes $V_i = \{u_i, \bar{u}_i\}$ and one edge $E_i = \{\{u_i, \bar{u}_i\}\}$, that is two nodes joined by a single edge. Each node in the instance of GIS will have a list of elements associated with it. In the instance we are creating, the list of elements associated with both the nodes u_i and $\bar{u}_i$ will contain a single element x_i, i.e., $L(u_i) = L(\bar{u}_i) = \{x_i\}$. It may be noted that any independent set will contain at most one node from the set $\{u_i, \bar{u}_i\}$.

Phase II: For each clause $c_j \in C$, we construct a part of the graph with three nodes and three edges.
$V'_j = \{a_1[j], a_2[j], a_3[j]\}$
$E'_j = \{\{a_1[j], a_2[j]\}, \{a_1[j], a_3[j]\}, \{a_2[j], a_3[j]\}\}$

In addition, we assign $L(a_1[j])$ = $L(a_2[j])$ = $L(a_3[j])$ = $\{y_j\}$.
It may be noted that an independent set will contain at most one node from the set $\{a_1[j], a_2[j], a_3[j]\}$.

Phase III: For each clause $c_j \in C$, let the three literals in c_j be denoted by $c_{j_1}, c_{j_2}, c_{j_3}$. In this phase, we do not introduce any nodes, but introduce three additional edges, corresponding to each clause c_j.
$E''_j = \{\{a_1[j], c_{j_1}\}, \{a_2[j], c_{j_2}\}, \{a_3[j], c_{j_3}\}\}$

This concludes the construction of our instance of the GIS, with graph $G = (V, E)$, where
$V = (\cup_{i=1}^{g} V_i) \cup (\cup_{j=1}^{h} V'_j)$ and
$E = (\cup_{i=1}^{g} E_i) \cup (\cup_{j=1}^{h} E'_j) \cup (\cup_{j=1}^{h} E''_j)$

where g and h corresponds to the number of variables and clauses of the instance of the 3-SAT problem respectively. The instance of the graph $G = (V, E)$ constructed using the rules specified above will have $2g + 3h$ nodes and $g + 6h$ edges. Figure 3 shows the instance of the GISF constructed from an instance of 3-SAT where $U = \{u_1, u_2, u_3, u_4\}$ and $C = \{\{u_1, \bar{u_3}, \bar{u}_4\}, \{\bar{u_1}, u_2, \bar{u}_4\}\}$. It may be noted that this construction can be completed in polynomial time.

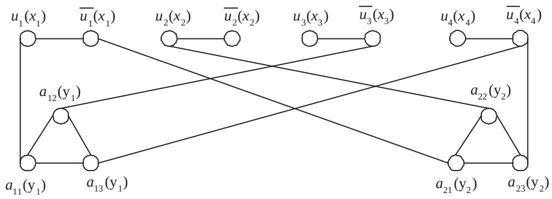

Fig. 3. Instance of the GISF constructed from an instance of 3-SAT where $U = \{u_1, u_2, u_3, u_4\}$ and $C = \{\{u_1, \bar{u_3}, \bar{u}_4\}, \{\bar{u_1}, u_2, \bar{u}_4\}\}$

Claim 1: The instance of the 3-SAT problem has a satisfying truth assignment if and only if the instance of the the GISF has a feasible solution.

Definition: The *node cover* of a graph $G = (V, E)$ is a subset $V' \subseteq V$, such that all edges in E have at least one end point in the set V'.

As a step towards showing that the instance of the 3-SAT problem has a satisfying truth assignment if and only if the instance of the the GISF has a feasible solution, we first establish that the instance of the 3-SAT problem has a satisfying truth assignment if and only if the instance of the the GISF has a node cover of size at most $g + 2h$.

Claim 2: The instance of the 3-SAT problem has a satisfying truth assignment if and only if the instance of the the GISF has a node cover of size at most $g + 2h$. Moreover, this node cover must contain exactly one node from each $V_i, 1 \leq i \leq g$ and exactly two nodes from each $V'_j, 1 \leq i \leq h$.

Proof of Claim 2: The proof of this claim is given in [17] (pages 54-56) and is not repeated here for brevity.

Proof of Claim 1: In any graph $G = (V, E)$, if V' is a node cover, then $V - V'$ must be an independent set. If the instance of the GISF has a node cover of size $g + 2h$

that contains exactly one node from each $V_i, 1 \leq i \leq g$ and exactly two nodes from each $V'_j, 1 \leq i \leq h$, then the instance of the GIS must also have an independent set of size $g+h$ that contains exactly one node from each $V_i, 1 \leq i \leq n$ and exactly one node from each $V'_j, 1 \leq i \leq m$. It can be easily verified that the union of the lists associated with the nodes belonging to this independent set will be equal to $\cup_{i=1}^{g} x_i \cup \cup_{i=1}^{h} y_i$. This independent set thus constitutes a feasible solution for the GISF. Thus we can conclude that if the instance of the 3-SAT problem has a satisfying truth assignment, the instance of the GISF problem has a node cover of size $g+2h$, which in turn implies that there is an independent set of size $g+h$ and the union of the lists associated with the nodes in this independent set is equal to $\cup_{i=1}^{g} x_i \cup \cup_{i=1}^{h} y_i$.

Conversely, if the instance of the 3-SAT problem has no satisfying truth assignment, then the size of the smallest node cover in the instance of the GISF problem is greater than $g+2h$. This implies that in the instance of the GISF problem, size of the smallest independent set is smaller than $g+h$. It may be observed that the union of the list associated with the nodes of an independent set of size smaller than $g+h$ cannot be equal to $\cup_{i=1}^{g} x_i \cup \cup_{i=1}^{h} y_i$. Thus, if the instance of the 3-SAT problem has no satisfying truth assignment, the instance of the GISF problem has no feasible solution.

Theorem 2: The GISM problem is NP-complete.

Proof: The transformation is from the GISF problem. The instance of the GISM problem constructed is exactly the same as the instance of the GISF problem. It is not difficult to verify that the instance of the GISF problem will have an independent set V' in the graph $G=(V,E)$ such that $\cup_{v_i \in V'} L(v_i) = A$, if and only if the instance of the GISM problem has an independent set V' whose *cost* is less than 1. This proves that the GISM problem is also NP-complete.

A. Hardness of approximation of the GISM problem

In this subsection we show that no polynomial time ϵ-approximation algorithm [17] can be developed for the GISM problem unless $P = NP$.

Theorem 3: Unless P = NP, no ϵ-approximation algorithm exists for the GISM problem with $\epsilon \leq \frac{1}{n\alpha}$ (α is a number much smaller than $1/n$).

Proof: Suppose that there exists a polynomial time ϵ-approximation algorithm APP for the GISM problem with $\epsilon \leq \frac{1}{n\alpha}$. This implies that for any instance I of the GISM problem, ratio between the approximate solution for the instance I, APP(I), and the optimal solution, $OPT(I)$, is bounded by ϵ.

Claim: If there exists a polynomial time ϵ-approximation algorithm APP for the GISM problem with $\epsilon \leq \frac{1}{n\alpha}$, then the GISF problem (proven to be NP-complete earlier) can be solved in polynomial time.

The optimization version of the GISM problem finds an independent set V' of minimum cost $C(V')$. The approximation algorithm APP returns an independent set V'' with a cost value $C(V'')$. If $C(V'') < 1$, we know that the instance of the GISF problem has an independent set V'' such that $\cup_{v_i \in V''} L(v_i) = A$. If $C(V'') > 1$, we know that the instance of the GISF problem has no independent set U such that $\cup_{v_i \in U} L(v_i) = A$.

The reason for the last statement is the following. Suppose (if possible) $C(V'')$ returned by the ϵ-approximation algorithm APP (with $\epsilon \leq \frac{1}{n\alpha}$) is greater than 1, but the instance of the GISF problem has an independent set V' such that $\cup_{v_i \in V'} L(v_i) = A$. If the instance of the GISF problem has an independent set V' such that $\cup_{v_i \in V'} L(v_i) = A$, then the optimal algorithm OPT would have returned an independent set with cost at most $n\alpha$, where n is the number of nodes in the graph $G=(V,E)$ and α is a number much smaller than $1/n$. In this case, the ratio between the objective value returned by the approximate algorithm APP (greater than 1) and the objective value returned by the optimal algorithm OPT (at most $n\alpha$), is not bounded by ϵ as $\epsilon \leq \frac{1}{n\alpha}$, contradicting the existence an ϵ-approximation algorithm.

IV. Optimal Solution for the Sensor Coverage Problem

From the discussion in section 2, it is clear that the Sensor Coverage problem is equivalent to Generalized Set Cover problem. The optimal solution for the Generalized Set Cover problem can be obtained by solving an Integer Linear Program (ILP). The ILP formulation of the GSC problem is given below.

From an instance of the GSC, first construct a $n \times k$ matrix A whose entries are either 0 or 1. $A(i,j)=1$ if the element s_i is a member of the subset S_j, otherwise $A(i,j)=0$.

We will use an indicator vector $\mathbf{x} = \{x_1, \ldots, x_k\}^T$ to indicate if the subset S_i is in the final solution set S' or not. Accordingly, $x_i = 1$ iff $S_i \in S'$ and $x_i = 0$ otherwise. The ILP formulation of GSC is

Minimize $\sum_{i=1}^{n} x_i$, subject to the constraints

(i) $A\mathbf{x} \geq 1$, and

(ii) $\forall x_i, x_j, x_i + x_j \leq 1$ if $(S_i, S_j) \in INC$.

The optimal solution to the GISM problem can be found by solving the following integer non-linear (quadratic) programming problem. Corresponding to every node v_i, we will have one binary variable x_i. The variable x_i takes value 1, if v_i is part of the independent set V' and 0 otherwise. Corresponding to every element $a_i \in A$, we will have one binary variable y_i. The variable y_i takes value 1, if $a_i \in L(v_i)$ and v_i is part of the independent set V'. The variable y_i is 0 otherwise. The IQP for the GISM is the following:

$Minimize \quad (m - \sum_{i=1}^{m} y_i + \alpha) \times (\sum_{i=1}^{n} x_i)$ (α is a number much smaller than $1/n$), subject to the following constraints

(i) $\sum_{i=1}^{n} x_i \geq 1$

(ii) $\forall i, j,\ x_i + x_j \leq 1$ if $(v_i, v_j) \in E$.

(iii) $\forall i,\ y_i \leq \sum_{v_i \in V_i} x_i$ where $V_i = \{v_i : a_i \in L(v_i)\}$.

(iv) $\forall i, x_i, y_i = 0/1$.

We can convert the above integer non-linear programming problem to an integer linear problem by introducing a new binary variable $z_{i,j}$ that is equal to 1 when $x_i = 1$ and $y_i = 1$, and 0 otherwise. The ILP is given next.

$Minimize \quad (m + \alpha) \times (\sum_{i=1}^{n} x_i) - \sum_{i=1}^{n} \sum_{j=1}^{m} z_{i,j})$, where α is a number much smaller than $1/n$.

subject to the following constraints

(i) $\sum_{i=1}^{n} x_i \geq 1$

(ii) $\forall i, j,\ x_i + x_j \leq 1$ if $(v_i, v_j) \in E$.

(iii) $\forall i,\ y_i \leq \sum_{v_i \in V_i} x_i$ where $V_i = \{v_i : a_i \in L(v_i)\}$.

(iv) $\forall i, j,\ x_i + y_j \geq 2z_{i,j}$.

(v) $\forall i, j, x_i, y_i, z_{i,j} = 0/1$.

V. Heuristics for the Sensor Coverage Problem

We showed that the Sensor Coverage Problem can be viewed as a Generalized Set Cover Problem. There exists a number of heuristic solutions for the Set Cover problem. We will tailor one such algorithm [18] to suit our needs to solve the sensor coverage problem stated in the form of Generalized Set Cover problem. It may be recalled that the Generalized Set Cover (GSC) Problem is specified by (i) a set of elements $S = \{s_1, \ldots, s_n\}$, (ii) a collection of subsets $S_i \subseteq S, \mathcal{S} = \{S_1, \ldots, S_k\}$, (iii) a collection of tuples of incompatible subsets $INC = \{(S_{i_1}, S_{j_1}), (S_{i_2}, S_{j_2}), \ldots, (S_{i_t}, S_{j_t})\}$ and the goal is to find the smallest subset $\mathcal{S}' \subseteq \mathcal{S}$, such that

(i) $\forall s_i \in S, \exists S_k \in \mathcal{S}'$, such that $s_i \in S_k$ and

(ii) if $S_i, S_j \in \mathcal{S}'$, then $(S_i, S_j) \notin INC$. From an instance of the GSC, we can construct an $n \times k$ matrix A whose entries are either 0 or 1. $A(i, j) = 1$ if the element s_i is a member of the subset S_j, otherwise $A(i, j) = 0$. We present two different greedy heuristics for the solution of the GSC problem. Both the heuristics select one column after another of matrix A with a goal of covering the largest number of element of the set S.

It may be noted during execution of the algorithm each column of matrix A (corresponding to each subset $S_1, \ldots, S_k$) can be in exactly one of the following three states: *selected, blocked, free*. A column is classified as *selected* if it is selected by the algorithm to be part of the cover. A column is classified as *blocked* if it is incompatible with a selected column. A column is *free* if it is neither selected nor blocked. The *benefit* associated with a column is measured by the number of uncovered elements $(s_1, \ldots, s_n)$ (at that time) that will be covered by the selection of that column. The *penalty* associated with a column is measured by the number of columns that will be blocked (at that time) by the selection of that column. Both the heuristics use *selection-merit* associated with a column for selecting the next column to be included in the cover. The selection-merit of a column used by the first heuristic is the same

Algorithm 1 heuristic_generalized_set_cover

1: **initialize**
   Set-of-Free-Columns $\Leftarrow \{S_1, \ldots, S_k\}$
   Set-of-Blocked-Columns $\Leftarrow \emptyset$
   Set-of-Selected-Columns $\Leftarrow \emptyset$
2: **repeat**
3: Find the column with highest selection merit in Set-of-Free-Columns
4: Identify the set of incompatible columns for this column
5: Move the highest selection merit column to Set-of-Selected-Columns
6: Move the incompatible columns to Set-of-Blocked-Columns
7: **if** addition of this new column to Set-of-Selected-Columns creates any redundant columns **then**
8: Move the redundant columns to Set-of-Free-Columns in decreasing order of their redundancy merit
9: Move the incompatible columns corresponding to the set of redundant columns from Set-of-Blocked-Columns to Set-of-Free-Columns
10: **end if**
11: **if** all elements of the set S are covered **then**
12: Go to step 21
13: **end if**
14: **if** Set-of-Free-Columns is empty **then**
15: Go to step 21
16: **end if**
17: **if** Set-of-Free-Columns contains only columns with zero merit **then**
18: Go to step 21
19: **end if**
20: **until** Set S is covered *or*
   Set-of-Free-Columns $= \emptyset$ *or*
   Set-of-Free-Columns contains only columns with zero merit
21: **print** Set-of-Selected-Columns and the elements of the set S covered by these columns

as its benefit value. The selection-merit of a column used by the second heuristic the ratio of its benefit to penalty, if penalty is greater than zero. If penalty is equal to zero the selection-merit is taken to be equal to its benefit.

During each iteration, the algorithm chooses the free column with the highest selection-merit to be included in the cover. This column is moved from the set of free columns to the set of selected columns. If in the set of free columns, there exists columns that are incompatible with this selected column, they are moved to the set of blocked columns. A column in the set of selected columns may become *redundant*, if addition of a new column during one iteration renders its presence in the set of selected columns unnecessary, i.e., all the elements covered by this column is covered by some other selected columns. A redundant column is removed from the set of selected columns and is returned to the set of free columns. In addition, some of the columns blocked due to inclusion of this column in the set of selected columns, may also be returned to the set of free columns at this time (if they are not blocked by some other columns, still part of the selected set). The *redundancy-merit* of a column is determined by the number of uncovered elements that can potentially be covered by the movement of the set of columns from the set of blocked columns to the set of free columns. In case of multiple redundant columns, the one with the highest redundancy merit is removed first. Redundant columns are removed one after another until no redundant columns are in the set of selected columns.

VI. Simulations Results and Discussion

We conducted extensive simulations to evaluate the efficacy of the two heuristics proposed in the section VI by comparing their performances with the optimal solution. For the simulation experiments, we generated two sets of uniformly distributed random points on a plane. In the absence of the information about the distribution of these points in practice, we have assumed a random distribution in our simulations. The first set represents the points to be sensed (blue points) and the second represents the potential location of the sensors (green points). In addition, we generated the sensing radius of the sensors r_{sen} and the minimum separation distance between the sensors d_{cr}. We are particularly interested in exploring the impact of d_{cr} in finding solution to the GISM problem and its relation with r_{sen}. If $d_{cr} = 2r_{sen}$, then we may have a situation where a part of the sensing region cannot be covered by any sensors. This is depicted in figure 4(b) where the area enclosed by the three points where the circles meet cannot be covered by any sensors. However, as shown in figure 4(a), this problem does not exist if $d_{cr} \leq \sqrt{3}r_{sen}$. For this reason, in our simulation experiments we have $d_{cr} \leq \sqrt{3}r_{sen}$.

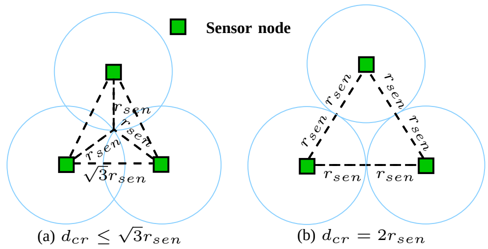

Fig. 4. Relationship of d_{cr} with r_{sen} (a) $d_{cr} = \sqrt{3}r_{sen}$ (b) $d_{cr} = 2r_{sen}$

We conducted the simulation experiments by varying the minimum separation distance parameter d_{cr} from 0 through 50 while keeping the sensing radius parameter (r_{sen}) fixed at 30. The number of potential sensor locations (green points) were considered to be significantly lower than that of the points to be sensed (blue points). In our experiments we kept the number of blue points to be thrice the number of green points. Five data sets (I1 through I5) were generated for each combination of (i) the number of blue points (B), (ii) the number of green points (G), (iii) r_{sen} and (iv) d_{cr}. The goal of the optimal algorithm as well as the two heuristics were to find the largest number of blue points that can be covered with the smallest number of green points. It may be noted that our optimal algorithm is guaranteed to find a solution that covers all the blue points with the smallest number of green points (subject to minimum separation distance constraint), if such a solution exists. The results of our simulation are presented in table I (next page). It may be noted that some entries in the table have a number within parentheses associated with it while many other entries do not. The implication of an entry (say X) having no number within parentheses associated with it is that X green points (sensor locations) are sufficient to cover all (150) blue points. If an entry (say Y) has a number (say Z) within parentheses associated with it, it implies that the largest number of blue points that can be covered is Z (not 150) and it can be done using Y green points.

The evaluation of the heuristics show that both of them produce near optimal solution most of the time. Moreover, they produce such high quality solution in a fraction of the time needed to find the optimal solution. While some problem instances needed more than 24 hours of computing time to find the optimal solution on a Pentium IV machine running CPLEX version 8.0, the heuristics produced near optimal solutions in only a few seconds. From their performance, we can conclude that both heuristics are quite effective. Between heuristics 1 and 2, the second heuristic is somewhat more “intelli-

TABLE I

PERFORMANCE COMPARISON OF OPTIMAL AND HEURISTICS SOLUTIONS WITH B = 150, G = 50 AND $r_{sen} = 30$. (IN THE TABLE, **OPT** = OPTIMAL, **H1** = HEURISTIC 1, **H2** = HEURISTIC 2.)

d_{cr}	Data Set 1			Data Set 2			Data Set 3			Data Set 4			Data Set 5		
	Opt	**H1**	**H2**	**Opt**	**H1**	**H2**	**Opt**	**H1**	**H2**	**Opt**	**H1**	**H2**	**Opt**	**H1**	**H2**
0	6	8	8	7	7	7	6	7	7	6	7	7	7	10	10
5	6	8	8	7	7	7	6	7	7	6	7	7	7	10	10
10	6	8	8	7	7	7	6	7	8	6	7	8	7	10	8
15	6	8	7	7	8	10	6	7	8	6	8	7	7	9	10
20	6	8	8	7	8	9	6	7	7	6	8	9	7	9	8
25	6	8	8	7	7(149)	9	6	7	7	6	8	8	7	8(146)	9
30	6	8	8	7	6(145)	8(149)	6	6(141)	9(144)	6	7	10	7	6(144)	8(147)
35	6	5(143)	7(142)	7	5(142)	7(148)	6	6(130)	6(144)	6	6(145)	7(143)	7(148)	5(141)	6(140)
40	6	5(143)	4(131)	6(149)	5(140)	6(145)	6	5(127)	6(143)	6(148)	5(129)	5(132)	6(147)	5(141)	5(128)
45	6(145)	4(130)	4(99)	6(143)	5(134)	5(140)	6(145)	3(105)	4(114)	6(146)	4(115)	5(130)	5(139)	4(127)	4(120)
50	4(138)	4(128)	4(98)	5(139)	4(121)	4(113)	6(143)	4(111)	4(112)	4(136)	4(115)	4(131)	4(134)	3(112)	4(125)

gent” in the sense that its benefit function not only takes into account the number of uncovered blue points being covered by the selection of a green point, but also takes into account the number of green points that are blocked from future consideration due to selection of this green point. However in practice, the seemingly “unintelligent” heuristic 1 seem to be outperforming heuristic 2 in many instances. Since this is somewhat counter-intuitive, we try to explain the phenomenon below.

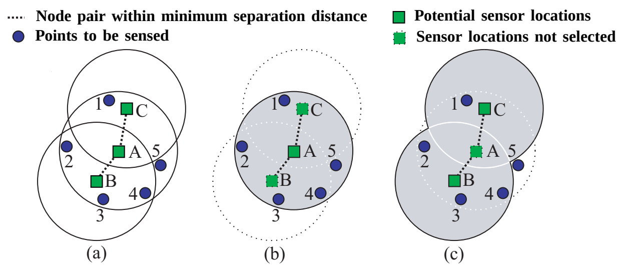

Fig. 5. Comparison of Heuristics 1 and 2

In the example shown in figure 5(a), we have $d_{cr} = 15$ and $r_{sen} = 30$. If we use heuristic 1, whose selection metric is the number of uncovered blue points, we have Selection-merit(A) = 5, Selection-merit(B) =3, Selection-merit(C) = 1. So, heuristic 1 chooses the green point A first, which is enough to cover all the blue points as shown in 5(b). This is the optimal solution for this problem instance. However if we use the heuristic 2, whose selection merit is the ratio of the number of uncovered blue points to the number of green points blocked, we have Selection-metric(A) = 5/2 = 2.5, Selection-merit(B) = 3/1 = 3, Selection-metric(C) = 1/1 = 1. In this case, the algorithm will choose green point B first, which only covers the blue points 2, 3 and 4, and blocks green point A. Then the only green point that the heuristic can choose is point C. This scenario is depicted in figure 5(c). So, the final solution using the heuristic 2 will comprise of green points B and C covering only blue points 1, 2, 3 and 4. The reason why heuristic 2 fails to find as good a solution as heuristic 1 in this case is the fact that the separation distance (d_{cr}) is much smaller than the sensing radius r_{sen}.

It may be recalled that both the heuristics use the selection-merit to identify the green point to be included in the cover. In case of heuristic 1, the selection-metric is equal to the benefit value and in case of heuristic 2, it is equal to the benefit to penalty ratio. When r_{sen} is much larger than d_{cr}, the benefit associated with a green point is much larger than the penalty and in such cases the heuristic 1 seems to be doing better than heuristic 2. On the other hand when d_{cr} is much larger than r_{sen}, the penalty associated with a green point is much larger than the benefit and in such cases the heuristic 2 seems to be doing better than heuristic 1. These observations are consistent with the simulation results.

VII. CONNECTED SENSOR NETWORK

The previous section presented optimal and heuristic solutions to find the minimum number of sensors required to cover a set of points in temperature sensitive environments. In applications where these sensors are not directly connected to the controller, they should form a connected network so that the data sensed by any sensor can be delivered to the controller (possibly by multiple hops through other sensor nodes). We provide a two-phase solution for such applications, where the first phase optimizes on the coverage (using the heuristics of the previous section) while the second phase determines the minimum transmission range necessary so that the selected sensors can form a connected communication network. We assume all sensors have same transmission range and two sensors can communicate if they are within the transmission range of one another. Since power consumption of the sensors is directly proportional to the transmission range, we form the connected communication network with smallest possible transmission range. This problem is formally stated as follows.

Transmission Range Problem: Given a complete graph G=(V, E) where V represents the set of sensors selected in the first phase and non-negative edge weights $w(e)$ for $e = (u, v)$, $u, v \in V$ representing the Euclidean distance between sensor u and v, find a spanning tree T of G such that $\max_{e \in T} w(e)$ is minimized. This minmax spanning tree problem is also known as the bottleneck spanning tree problem in the literature [19].

Algorithm 2 compute_smallest_transmission_range

Sort the edges of E such that $w(e_1) \leq \ldots \leq w(e_m)$
Set $E' = \{e_1\}$
for $i = 2$ to m **do**
    if $E' \cup \{e_i\}$ is acyclic **then**
        $E' = E' \cup \{e_i\}$
    end if
end for
return $\max_{e \in E'} w(e)$

The algorithm computes the edges E' that form the minmax spanning tree and returns the largest edge weight in the tree. The transmission range of the sensors set to the largest edge weight ensures that the resulting communication network is connected. The algorithm runs in O($|E| \log(|E|)$) time.

Theorem 2 Any spanning tree T^*=(V, E') where E'=$\{e'_1, e'_2, \ldots, e'_{n-1}\}$ constructed using the above algorithm produces a minmax spanning tree (MMST).

Proof: The proof is by contradiction. Suppose that for any spanning tree T of G other than T^*, $f(T)$ denotes the smallest value of i such that e'_i is not in T. Assume that T^* is not a MMST and T is a MMST such that $f(T)$ is as large as possible. Suppose that $f(T)$=k. This implies that $e'_1, e'_2, \ldots, e'_{k-1}$ are in both T and T^* but e'_k is not in T. Clearly, adding the edge e'_k to the edge set E' of the tree T creates a unique cycle C, i.e., $T + e'_k$ contains C. Suppose that e''_k is an edge in C that is in T but not in T^*. Since e''_k is not a cut edge of $T + e'_k$, $T + e'_k - e''_k$ is a connected graph with $m - 1$ edges and therefore another spanning tree of G. Clearly $w(T') = \max(w(T - e''_k), w(e'_k))$.

The algorithm chose the edge e'_k before it chose the edge e''_k, even though neither e'_k nor e''_k would have created a cycle with the edges $e'_1, e'_2, \ldots, e'_{k-1} e'_k$. This implies $w(e''_k) \geq w(e'_k)$. This observation, together with the fact that $w(T')$=$\max(w(T - e''_k), w(e'_k))$, concludes that $w(T') \leq w(T)$. Thus T' is also a MMST. However, $f(T') > k = f(T)$, contradicting the choice of T as the MMST with the largest $f(T)$. Therefore T=T^* and T^* is indeed a MMST.

VIII. Conclusion

In this paper, we have introduced a new version of the sensor placement and coverage problem. We have shown that both the optimization and the feasibility versions of the problem are NP-complete. Moreover, an ϵ-approximation algorithm for the problem cannot be developed unless P = NP. Our heuristics produce near optimal solution for most of the problem instances in a fraction of the time needed to find an optimal solution.

References

[1] L. Schwiebert, S. K. Gupta, P. Siy, G. Auner, and et al, "A biomedical smart sensor for the visually impaired," *IEEE Sensors*, 2002.

[2] B. J. Sargent and D. A. Gough, "Design and validation of the transparent oxygen sensor array," *IEEE Trans. on Biomedical Engineering*, vol. 38, no. 5, 1991.

[3] Body sensor networks. [Online]. Available: http://bsn.media.mit.edu

[4] L. Schwiebert, S. K. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *ACM/IEEE Conf. on Mobile Computing and Networking*, 2001.

[5] A. Hirata, G. Ushio, and T. Shiozawa, "Calculation of temperature rises in the human eye for exposure to em waves in the ism frequency bands," *IEICE Trans. on Communications*, vol. E83-B, no. 3, 2000.

[6] P. Bernardi, M. Cavagnaro, S. Pisa, and E. Piuzzi, "Sar distribution and temperature increase in an anatomical model ofthe human eye exposed to the field radiated by the user antenna in a wireless lan," *IEEE Trans. on Microwave Theory and Techniques*, vol. 46, no. 12, 1998.

[7] Geodesic sensor networks, egi corporation. [Online]. Available: http://www.egi.com

[8] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *IPSN*, 2004.

[9] S. Funke, A. Kesselman, F. Kuhn, and Z. Lotker, "Improved approximation algorithms for connected sensor cover," in *International Conf. on AD-HOC Networks and Wireless*, 2004.

[10] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution," in *MobiHoc*, 2003.

[11] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," in *WiOpt*, 2003.

[12] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *SenSys*, 2003.

[13] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," UIUC, Tech. Rep. UIUCDCS-R-2003-2351, 2003.

[14] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto, "Optimal packing and covering in the plance are np-complete," *Information Processing Letters*, vol. 12, no. 3, 1981.

[15] D. Hochbaum, *Approximation Algorithms*. PWS Publishing, 1997.

[16] A. Sen, N. Das, L. Zhou, B. Shen, S. Murthy, and P. Bhattacharya, "Coverage problem for sensors embedded in temperature sensitive environments," Arizona State University, Tech. Rep. TR-06-015, 2006.

[17] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman Press, 1979.

[18] E. Marchiori and A. Steenbeck, "An iterated heuristic algorithm for the set covering problem," in *Workshop on Algorithm Engineering*, 1998.

[19] J. Kleinberg and E. Tardos, *Algorithm Design*. Addison Wesley, 2005.