An Index-Based Sensor-Grouping Mechanism for Efficient Field-Coverage Wireless Sensor Networks

Yangfan Zhou Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
Email: {yfzhou, lyu}@cse.cuhk.edu.hk

Jiangchuan Liu
School of Computing Science
Simon Fraser University
Vancouver, Canada
Email: jcliu@cs.sfu.ca

Abstract— **This paper discusses a point-distribution index, ι, which measures the normalized minimum distance between sensors. Maximizing ι of a set of points causes the Delaunay triangulation graph of these points to be a net of equilateral triangles. Such a structure indicates the lowest redundancy of coverage if each point represents the center of a disc. Thus ι can serve as a promising measure for solving a critical problem in field coverage: How to group a set of sensor nodes into disjoint subsets so that each subset can cover the entire field? Based on the ι index, we develop an effective algorithm, MAXINE (MAXimizing-ι Node-redundancy Exploiting), for the sensor-grouping problem. We evaluate the performance of MAXINE through extensive simulations and compare it with existing algorithms. The results demonstrate the effectiveness of MAXINE and verify the superiority of employing ι for the sensor-grouping problem.**

I. Introduction

In a wireless sensor network (WSN), a large number of battery-powered sensor nodes are deployed to perform “field work”: collecting the data about some physical phenomena of interest. The low-cost implementation and unattended operational manner make WSNs especially well suited for environmental monitoring and surveillance [1][2]. Although sensor nodes in a WSN are critical devices to perform sensing tasks, their low-cost implementation and their field-working environments (which are sometimes even hostile, *e.g.*, in battlefield applications) suggest that they are subject to failures and permanent damages.

To enhance fault tolerance, a WSN often contains a large number of redundant nodes, which work in turn to ensure fault-tolerant operation and prolong the network lifetime. The scheduling should also ensure reliable field-coverage. In a basic 1-coverage system, each point in the network field should be covered by at least one active sensor node (*i.e.*, within this node’s sensing range).

This paper studies how to group a set of sensor nodes into as many disjoint subsets as possible, subject to the reliable field-coverage requirement (*i.e.*, each subset of sensors can ensure the entire field-coverage). We call this problem the sensor-grouping problem.

In our preliminary study [3], we proposed a novel point-distribution index called ι to present a normalized minimum distance, which is the minimum distance between each pair of points normalized by the average distance between each pair of points. Based on ι, we presented a preliminary algorithm for the sensor-grouping problem. This paper further provides a comprehensive study on the sensor grouping problem to show that maximizing the point-distribution index ι of active nodes in a WSN serves as a novel point of view for coverage-related problems.

The main contributions of this paper are two-fold. First, comparing with our preliminary study in [3], we provide a more detailed illustration of the features of ι and propose the ι-Thesis. More importantly, we conduct comprehensive experimental studies to compare our ι-based mechanism, *i.e.*, the MAXimizing-ι Node-redundancy Exploiting (MAXINE) algorithm with the most up-to-date algorithms in the literature, which serves as a strong evidence to show that our ι-based mechanism is a promising approach to attack the coverage-related problems. Second, reliable field-coverage of WSNs is studied. We investigate how to sample the coverage quality of a sensing field with discrete points in reliable field-coverage WSNs. Quasi-random sampling methods are then suggested.

The rest of the paper is organized as follows. Section II surveys related work. In Section III, we formulate the sensor-grouping problem with some discussions on sensing model and field-coverage sampling. Section IV elaborates the MAXINE algorithm for the sensor-grouping problem based on our point-distribution index ι and its features. Section V presents our comprehensive experimental studies. Section VI provides the conclusion remarks.

II. Related Work

A WSN generally contains a large number of redundant nodes so as to achieve fault tolerance. In order to save energy and extend network lifetime, a node sleeping/working schedule scheme is therefore highly desired to exploit the redundancy of working sensors and let as many nodes as possible work in sleep mode.

Much work in the literature is on this sensor sleeping/working scheduling issue, which can be classified into two streams. The first stream of work focuses on online distributed and localized algorithms, in which a sensor node determines its sleeping eligibility and the time it can sleep based on the

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

coverage requirement of its sensing field with the cooperation of its neighbors [4][5].

Another stream of research targets on grouping sensor nodes. Sensor nodes in a network are divided into disjoint sets. Each subset is able to maintain the required sensing tasks. The sensor nodes are scheduled according to the subset they belong to. These subsets work successively: At any time, only one subset of sensor nodes are working, while the rest of the sense nodes are sleeping. This paper belongs to this stream of work.

In [6] by Slijepcevic *et al*, a sensing field is divided into regions. Sensor nodes are grouped with the most-constrained least-constraining algorithm. It is a greedy algorithm in which the priority of selecting a given sensor is determined by how many uncovered regions this sensor covers and the redundancy caused by this sensor. Cardei *et al* [7] model the problem as disjoint dominating sets. The problem is known as NP-complete, and they thus propose a graph-coloring based approximation. A similar problem of covering target points are studied in [8], which is again NP-complete and mixed integer programming (MIP) approximation has been proposed. These algorithms are centralized solutions of sensor-grouping problem. In our work, instead of studying the problem with a graph theoretical formulation, we study this problem in a geometric point of view. We show that maximizing the ι index of a subset of sensors (*i.e.*, the normalized minimum distance between sensors of a subset) results in low redundancy of the subset. We thus propose a fast algorithm to maximize the ι index in order to find the low-redundancy subsets of sensor nodes. Moreover, this algorithm is easy to be extended to a distributed implementation.

III. Sensor-Grouping Problem

A. Problem Formulation

TABLE I

Description of symbols

n	The number in-network sensor nodes
$s_i \ (i = 1, 2, ..., n)$	Sensor nodes
$\mathcal{S}_j \ (j = 1, 2, ..., m)$	The jth subset of sensor nodes, where m is the number of disjoint subsets
$\mathcal{L}(s_i) \ (i = 1, 2, ..., n)$	The physical location of node s_i
ϕ	The area monitored by the network
R	The sensing radius of a sensor node

To facilitate our discussion, we list the major notations in Table I. Assume that a sensor node is responsible to monitor a circular area centered at the node with a radius equal to R. This circular area is called the *sensing field* of the node. We also assume that each sensor node can know its approximate physical location $\mathcal{L}(s_i)$, which is obtainable if each sensor node carries a GPS receiver or if some localization algorithms are employed (*e.g.*, [9]). These are general assumptions made in the current work on the sensor coverage, e.g., [6][8][10].

The sensor-grouping problem is formulated as follows.

Problem 1: Given:

- *The set of sensor nodes* $\{s_1, s_2, ..., s_n\}$ *and the location* $\mathcal{L}(s_i)$ *of each sensor node.*
- *A* sensing model *which quantitatively describes how a point* P *in area* ϕ *is covered by sensor nodes that are responsible to monitor this point. We call this quantity the* coverage *of* P.
- *The coverage requirement in* ϕ, *denoted by* τ. *When the coverage of a point is not smaller than this threshold, we say this point is* covered.

Maximize: m, *the number of disjoint subsets.*
Subject to:

- $\{s_1, s_2, ..., s_n\} \supseteq \mathcal{S}_1 \cup \mathcal{S}_2 \cup ... \cup \mathcal{S}_m$
- $\mathcal{S}_j \cap \mathcal{S}_k = \varnothing$ *($\varnothing$ denotes an empty set);* $\forall j, k = 1, 2, ..., m$ *and* $j \neq k$
- *Area* ϕ *can be covered by sensor nodes in each set* $\mathcal{S}_j$ $(j = 1, 2, ..., m)$ ■

In other words, the sensor-grouping problem is to address how to divide sensor nodes into as many disjoint subsets as possible, while each subset can maintain the coverage requirement of the entire network field.

B. The Sensing Model and Field-Coverage Sampling

For a complete description of the sensor-grouping problem 1, a concrete sensing model should be given. Also we need a criterion to determine whether an area ϕ is covered. We discuss these two issues in this subsection.

A *sensing model* describes how a point in the network field is covered can be modeled and quantified. In this paper, we focus on the classical *Boolean sensing model* [4][5]. In this model, it is assumed that a sensor node can always detect an event occurring in its responsible sensing field, which is a circular area centering at the sensor with radius R. It is the same as the classical disc-cover model studied in geometry literature (*e.g.* [11]). The model is described as follows.

For a point with physical location L,

$$\mathcal{C}_j(L) = \begin{cases} 1, & \text{if } \exists s_i \in \mathcal{S}_j \text{ and } \|\mathcal{L}(s_i) - L\| < R; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $\mathcal{C}_j(L)$ denotes the coverage quality of subset $\mathcal{S}_j$ at location L in the network field, and $||x_i - x_j||$ denotes the Euclidean distance between location x_i and location x_j. In this model, the required coverage τ is equal to 1. We adopt this model for it has been widely used in the literature and it largely captures the unique features of sensor networks[1].

An algorithm in solving Problem 1 needs to ensure reliable coverage of the entire network field. To evaluate the coverage of a sensing field, the field should be sampled by some discrete points in the field. And then given a sensing model, *e.g.*, that described in Equation (1), the coverage of these points can be quantified. The field is deemed reliably covered if the coverage quality of each point is not smaller than the requirement τ.

Sampling field coverage with discrete points is in fact an approximation approach to ensure that the entire field

[1] In our previous work [3], we also discuss a *collaborative sensing model* which captures the situations where sensor nodes are considered to detect events of interest in a collaborative manner.

is covered. Obviously, the larger the number of sampling points is, the better the approximation of the field-coverage sampling will be, which on the other hand also results in longer converging time of the algorithm in solving Problem 1. Such tradeoffs are investigated in our experimental study presented in Section V.

Besides the number of sampling points, how to generate the points is also an important issue. In this work, we sample field coverage with four methods: 1) regular-lattice-based sampling, 2) random sampling, 3) quasi-random sampling, and 4) sensor-node-based sampling. Details on these sampling methods and comprehensive experiments to study the performance of the algorithms in solving Problem 1 given these methods will be presented in Section V.

IV. Maximizing-ι Node-redundancy Exploiting Algorithm (MAXINE) for Sensor-grouping

A. Normalized Minimum Distance

The *normalized minimum distance*, namely ι, to evaluate the distribution of the points in order to measure how well the points separate from one another is defined as follows [3][2].

$$\iota = \frac{\min(||x_i - x_j||}{\mu} (\forall\, i, j = 1, 2, ..., n; and\, i \neq j) \qquad (2)$$

where the $\min(\cdot)$ function calculates the minimum distance between each pair of points, and μ is the average distance between each pair of points.

Let us suppose each x_i ($\forall i = 1, ..., n$) is a *variable*. How does the structure of these n points look like if ι is maximized? Given n =3, 4, 5, and 6, the following graphs in Figure 1 is the resulting structures when ι is maximized in 2-dimensional space.

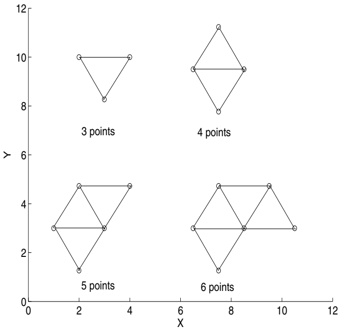

Fig. 1. Resulting structures of 3, 4, 5, and 6 points when ι is maximized

It is interesting to see the structures are all equilateral triangles. We have also numerically calculated the structures for greater n's with ι being maximized. Figure 2 demonstrates an resulting topology for $n = 40$.

More specifically, we draw the following thesis:

The ι-Thesis: *Given n points in a 2-dimensional space, ι of these points reaches the maximum value only if the Voronoi diagram*[3] *formed by these points is a net of equilateral hexagons,* i.e.*, these points forms a net of equilateral triangles.*

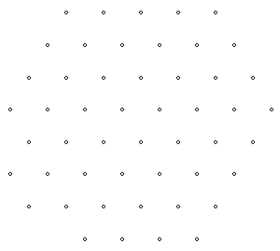

Fig. 2. Node number = 40, $\iota = 0.324970$

B. MAXINE algorithm for Sensor-grouping

In the disc-cover problem setting [13], *i.e.,* given a set of points representing the centers of a set of unit-diameter discs, it is proven that asymptotically when the Voronoi diagram of these points is a net of equilateral hexagons, the field covered by these discs achieve the maximum value [11]. In other words, if these points form a net of equilateral triangles, the network results in lowest redundancy and thus the coverage efficiency is the best. This result can be directly applied to the sensor-network coverage problem.

A sensor node can be deemed as the center of its circular sensing field with diameter equal to its sensing range. It is desirable that the active sensor nodes that are performing sensing task should separate from one another as much as possible to achieve low redundancy. Under the constraint that the entire sensing field should be covered, the more each node separates from the others, the less the redundancy of the coverage is. ι of these nodes indicates the quality of such separation. Particularly, in the sensor-grouping problem we formulated, it is necessary that ι of each subset should be maximized. This is the underlying thought of our Maximizing-ι Node-redundancy Exploiting Algorithm (MAXINE) algorithm.

MAXINE (See Figure 3 for the flow diagram) tentatively selects all ungrouped nodes[4] into the current subset (initially, the subset is an empty subset $\varnothing$). And then, one by one, it removes nodes from the subset until the deletion of any of the nodes in the subset will result in uncovered representative points. The selection criteria of the node to be removed are as follows.

- The deletion of the selected node does not result in any uncovered sampling points.
- The deletion of the selected node results in the maximum ι value of the current subset, comparing to the deletion of all other nodes in the subset.

In this way, a subset (*i.e.*, the resulting subset) that can covered the entire network field is successfully found.

[2] In this definition, n is larger than 2 and we do not consider the case that $x_i = s_j$, $\forall\, i, j = 1, 2, ..., n$.

[3] A Voronoi diagram formed by a set of nodes partitions a space into a set of convex polygons such that points inside a polygon are closest to only one particular node [12].

[4] In our following discussion, we call a node an *ungrouped node* if the node has not been grouped into any subset. Otherwise, we call the node a *grouped node*.

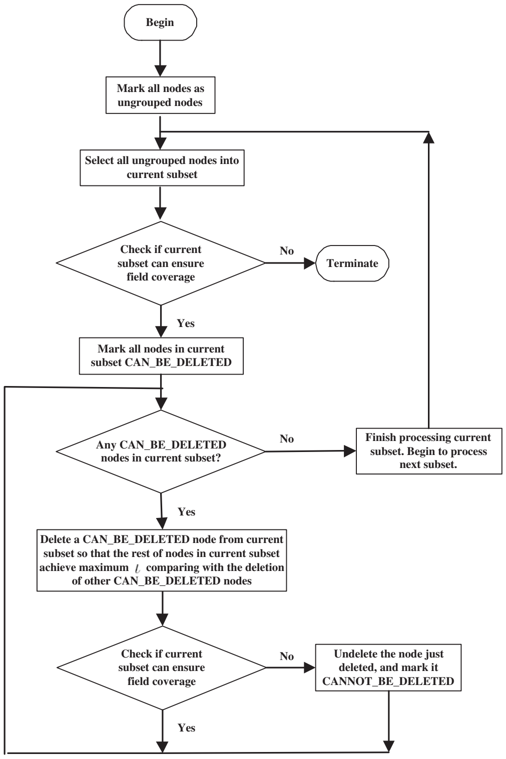
Begin
Mark all nodes as ungrouped nodes
Select all ungrouped nodes into current subset
Check if current subset can ensure field coverage
No
Terminate
Yes
Mark all nodes in current subset CAN_BE_DELETED
Any CAN_BE_DELETED nodes in current subset?
No
Finish processing current subset. Begin to process next subset.
Yes
Delete a CAN_BE_DELETED node from current subset so that the rest of nodes in current subset achieve maximum $\mathcal{L}$ comparing with the deletion of other CAN_BE_DELETED nodes
Check if current subset can ensure field coverage
No
Undelete the node just deleted, and mark it CANNOT_BE_DELETED
Yes

Fig. 3. The Flow Diagram of MAXINE

Iteratively, the algorithm begins to find the next subset with the same procedure until selecting all ungrouped nodes to the current subset eventually cannot guarantee that all sampling points are covered. Then the algorithm stops with all the successfully-found subsets.

C. Distributed Implementations

Although MAXINE is a centralized algorithm, it is easy to be extended to a distributed localized algorithm. In fact, MIND in our previous work [3] can be deemed as an example distributed version of an algorithm based on the $\mathcal{L}$-Thesis.

Also, note that a centralize algorithm like MAXINE requires to send only n (n is the number of in-network sensor nodes) messages, which is affordable for WSNs. The location of each node can be sent to a base station along the shortest-path tree rooted at the base station (Note that such a tree is generally a prerequisite for any WSN to report sensing-data from sensor nodes to a base station, and therefore can be utilized by a sensor-grouping mechanism) in the following way.

The leaf nodes in the tree send their locations first. When an in-network node collects all the location information of its offspring nodes, it encapsulates the locations of all its offspring nodes together with its own location in one packet and sends the packet to its parent node. In this way, each node needs to send only one packet and the base station can eventually know the locations of all in-network nodes. When the base station finishes performing MAXINE, it can directly send the grouping results to each in-network nodes if it is equipped with a powerful-enough antenna, or it can send the grouping results in a multi-hop manner along the reverse direction of the shortest-path tree.

V. Performance Study

As the algorithms for Problem 1 should ensure reliable field coverage, field-coverage is sampled with discrete points in the field. The field is deemed reliably covered if each sampling point is covered. We sample field coverage with the following four methods.

Regular Lattice (RL): Sampling points are selected in a regular manner. It divides the field into grids and the central point in each grid area is the sampling point for each grid.

Uniformly Distributed Random Points (UDRP): Sampling points are randomly selected in a uniform manner in the network field.

Quasi-Random Sequences (HTS and HMLS): Quasi-random sequences which have low *discrepancy* (a measure of uniformity for the distribution of the points) have been widely employed in Quasi Monte Carlo methods. This paper introduces quasi-random sequences to generating sampling points for evaluating coverage quality of WSNs. We consider two kinds of quasi-random sequences, Halton sequence [14] and Hammersley sequence [15]. Both sequences can achieve asymptotically optimal discrepancy and they are easy to construct. They are reasonably good sampling-point generators. In this work, we linearly map the 2-dimensional Halton sequence and the 2-dimensional Hammersley sequence into network field to generate the locations of the sampling points we need. We abbreviate the Halton-sequence-based and the Hammersley-sequence-based methods to HTS and HMLS respectively in our following discussion.

Sensor-node-based points (SNB): In this method, sampling points are generated on-line, based on the sensor nodes in each subset. We select in-network-field points on the border of each node's sensing field (*i.e.*, the perimeter of the circle center at the node with radius R)[5]. If all such points of each node in a subset are covered by other nodes in the same subset, the network field is considered covered by the subset. In the following discussion, in case that this SNB method is employed, we say a sampling point is covered if it is covered by other nodes in the same subset as the node that generates it.

In order to study the performance of our MAXINE algorithm, we also implement two other algorithms in solving sensor-grouping problem for comparisons. The first algorithm is a fast greedy algorithm and the second is based on the recent work in [8].

1) Greedy Algorithm (GA): The greedy algorithm we design selects an ungrouped node to the current subset (initially, the subset is an empty set $\varnothing$) when the gain of adding this node[6] is the maximum among all ungrouped nodes. This

[5] In the following discussion, we say the node *generates* the sampling points.

[6] If the field-coverage sampling method is not SNB, *the gain of adding a node* is the number of uncovered sampling points that can be covered by adding this node. Otherwise, the gain of adding a node is the sum of the sampling points the node can cover, which are generated by this node and the nodes in current subset but not covered by any other nodes in the subset

process is repeated until the subset can cover all the sampling points. In this way, a subset that can cover the entire network field is successfully found. Iteratively, the algorithm continues to find the next subset until the gain of adding any ungrouped node to the current subset is 0. Then it stops.

2) Algorithm Based on Mixed Integer Programming (MIPA): In [8], the sensor-grouping problem for target covering is formulated as a Disjoint Set Covers (DSC) problem. A bipartite directed graph G is constructed in which the vertex set is composed by sensor nodes and the targets. A vertex representing a sensor node has a directed edge to the vertex representing a target if the target is in the sensing field of the sensor node. Then draw k copies of this graph where k is the minimum in-degree of vertices representing the targets in graph G. With some auxiliary vertices and edges, these k copies of graph G is connected into a one-component graph G'. With proper capacity values assigned to the edges of graph G', the DSC problem is transformed to a maximum-flow problem with a mixed integer programming (MIP) formulation. With an MIP solver, the subsets can thus be obtained. By regarding each sampling point as a target to be covered, we can directly apply this algorithm.

In our following discussions, the algorithm name, followed by a hyphen and then the field-coverage sampling method, indicates a scheme we employ to solve the sensor-grouping problem. For example, MAXINE-HTS is a name for the scheme in which the MAXINE algorithm is employed and the field-coverage sampling method is the Halton sequence.

The simulations are performed at a Sun Blade 2500 computer with two 1.6GHz UltraSPARC-IIIi CPUs and 2GB RAM. The tool we employed to solve the MIP problem in the MIPA schemes are the ILOG CPLEX Interactive Optimizer 9.1.0. We simulate a stationary network in a $400m \times 400m$ field where sensor nodes are randomly deployed with a uniform distribution. We consider that the field to be monitored by the network is actually a $360m \times 360m$ field centered at the $400m \times 400m$ area. The sensing range of each sensor node R is equal to $60m$. We conduct each of our simulations 10 times with different random seeds. The results are averaged.

In our performance study, we first study field-coverage sampling methods. We then investigate the performance of the algorithms in solving the sensor-grouping problem given the Boolean sensing model.

A. The Comparisons of Area-Coverage Sampling Methods

In our experiments, to quantitatively study the performance of the subsets found by each scheme, we randomly let a number of events (10000 in our experiments) take place at the $360m \times 360m$ field with a uniform distribution. According to the sensing model, we calculate whether an event can be successfully detected when a subset (found by each sensor-grouping scheme) is taking charge of the event detection. We get the number of events that each subset cannot detect and thus obtain the percentage of the event-detection failures.

We compare the field-coverage sampling methods (UDRP, RL, HTS, and HMLS) in terms of the percentage of event-detection failures[7]. The total node number in this study is 200. We vary the number of sampling points from 16 to 100. For each setting, we run the GA, MIPA, and MAXINE algorithms.

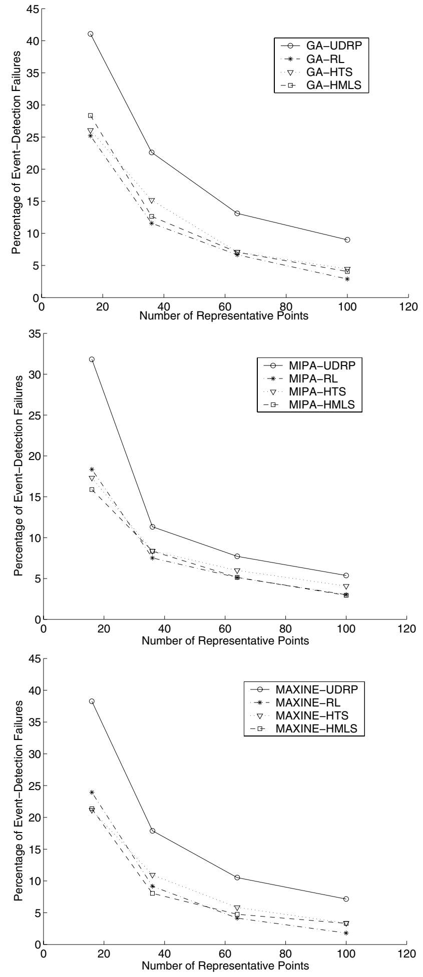

Fig. 4. Comparisons of UDRP, RL, HTS, and HMLS

Figure 4 demonstrates the average percentage of event-detection failures of the subsets found by the GA, MIPA, and MAXINE algorithms with different field-coverage sampling methods. We can see that an algorithm with the RL, HTS, and HMLS methods performs much better than that with the UDRP method. With the RL method, it performs the best. This is not surprising because the RL method generates representative points in a regular manner in which each representative point can be regarded as the expectation of the event locations, which is uniformly distributed, in a small field.

Discrete sampling points generated with quasi-random methods like HTS and HMLS have low discrepancy. As a result, the performance of an algorithm with quasi-random methods like HTS and HMLS is also good. Because of the quasi-random nature of the HTS and HMLS methods, they can be good alternatives to the RL method, especially in the

[7] We do not study the SNB method here because the number of representative points generated by this method is not deterministic. We will investigate this method in our later discussion.

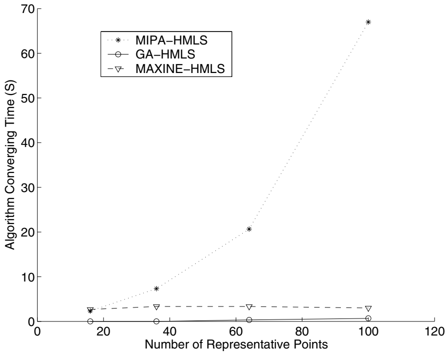

Fig. 5. The impact of sampling-point numbers on converging-time

application cases that sensor nodes are deployed with some deterministic scheme (*e.g.*, grid-based deployment), where the sampling points generated by RL method may be with high risk in providing wrong field coverage information as the pattern of sensor nodes and the pattern of sampling points may be highly related.

B. The Impact of Sampling-Point Numbers on Converging-Time

The larger the number of sampling points that is required to be covered by a subset is, the better the subset covers the entire field. This also requires more time for the algorithms that solve the sensor-grouping problem to converge. To study the converging-time of the GA, MIPA, and MAXINE algorithms, we deploy 250 sensor nodes in the network field. We vary the number of representative points from 16 to 100. Based on the above study, we adopt the HMLS method as the field-representation method. Other methods achieve similar simulation results and they are not included in this paper.

For each setting, we run the GA, MIPA, and MAXINE algorithms. The results generated are shown in Figure 5. It demonstrates that when the number of the representative points increases, the converging time of MIPA increases quickly, while the converging time of MAXINE and GA remains low in an almost constant manner. This is because the number of vertices grows linearly with the increase of the representative-point number in graph G in MIPA. And the number of the copies of G in graph G' also grows linearly with the increase of the representative-point number approximately. As a result, the edge number of G' grows with a second power of the number of the representative points approximately. Therefore, the scale of the maximum-flow problem, *i.e.*, the MIP formulated in the MIPA algorithm, increases very quickly as the number of the representative points increases. This is the main drawback of MIPA. Note that in our study, we conduct simulations on a fast workstation with an advanced commercial MIP solver. For comparison purpose, we have done simulations with GLPK [16], a very good *open-source* solver. The situation gets much worse when the number of the representative points increases.

In MAXINE and GA, sampling points are involved in testing whether a node should be selected/removed from a subset. This is not a major computational process. As a result, the sampling-point number does not have considerable impact on the converging time.

C. Performance of the Algorithms

We compare our MAXINE algorithm with the MIPA algorithm in terms of the number of subsets found by each algorithm with sampling points generated by the HMLS method. In this study, we vary the total number of sensor nodes n from 75 to 175 in order to study the impact of node density. The results are presented in Table II where x denotes the number of sampling points generated by the HMLS method.

TABLE II
NUMBER OF SUBSETS FOUND

	$x = 16$		$x = 36$	
n	MIPA	MAXINE	MIPA	MAXINE
75	2.00	2.00	1.67	1.33
100	4.00	4.00	4.00	3.33
125	5.33	5.33	4.67	4.33
150	7.67	7.33	6.67	5.33
175	9.00	8.33	8.67	6.67

	$x = 64$		$x = 100$	
n	MIPA	MAXINE	MIPA	MAXINE
75	1.33	1.33	1.33	1.33
100	2.67	2.67	3.00	2.67
125	4.00	3.33	3.67	3.00
150	6.00	5.00	5.67	4.67
175	7.00	6.00	7.33	5.00

	$x = 144$		$x = 196$	
n	MIPA	MAXINE	MIPA	MAXINE
75	0.67	0.67	1.00	1.00
100	3.00	2.67	3.00	2.33
125	3.67	2.67	4.00	3.00
150	6.00	4.67	6.00	4.67
175	6.00	4.33	6.33	4.67

When the node number is small, the performance of these two algorithms is close to each other in terms of the number of subsets found. When the node number is large, MIPA performs slightly better. In Figure 6, we show an example result of the impact of the total node numbers on the converging time of these two algorithms in case that the number of sampling points is 196.

We can see that in the large node number case, MIPA performs slightly better, but the converging time of MIPA increases quickly as the total number of sensor nodes increases. The reason is similar to the reason why the converging time of MIPA increases quickly as the number of sampling points increases, which has been presented in Section V-B.

We also investigate how the subsets perform in terms of the percentage of event-detection failures of the subsets. Figure 7 demonstrates the results in case that the number of representative points is 196. It shows that the performance of the subsets found by these two algorithms is comparable.

We compare GA and MAXINE algorithms in which the SNB method is employed and each sensor node generates 18

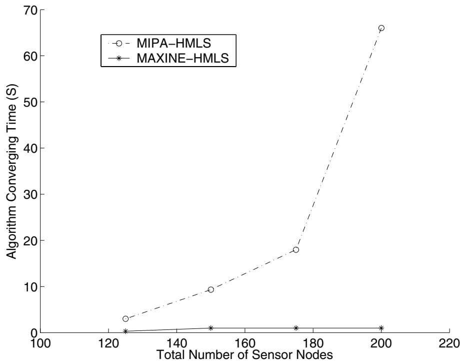

Fig. 6. The impact of total node numbers on converging time

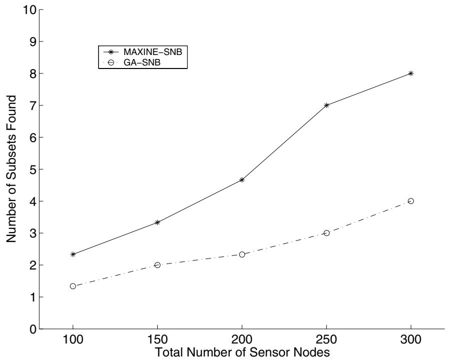

Fig. 8. The number of subsets found by GA and MAXINE

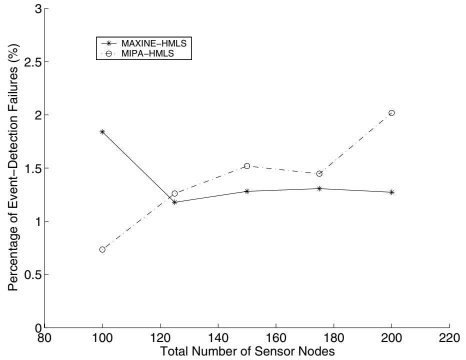

Fig. 7. Event-detection failures of the subsets found by MIPA and MAXINE

representative points. In this study, we vary the total number of sensor nodes from 100 to 300, and compare the number of subsets found by MAXINE and GA.[8] Figure 8 demonstrates the results. It shows that MAXINE performs much better than GA in terms of the number of subsets found.

These simulation studies show that an algorithm based on the ι-Thesis, which maximizes ι of the subsets, exhibits very satisfying results in terms of converging time, performance of the subsets, and the number of subsets found.

VI. Conclusion

In this paper, it is shown that maximizing ι results in low redundancy if each point represents a center of a disc in the disc-cover problem. This idea is employed in our algorithm called MAXINE (MAXimizing-ι Node-redundancy Exploiting), which serves as a novel approach to solve the sensor grouping problem. Comprehensive simulations are conducted and we verify the advantage of MAXINE by demonstrating its performance in converging time, field coverage of the subsets, and the number of subsets found.

Acknowledgement

The work was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4150/07E). J. Liu's work was supported in part by a Canadian NSERC Discovery Grant and an NSERC Strategic Project Grant.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[2] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. of the ACM SenSys*, Baltimore, MD, November 2004.

[3] Y. Zhou, H. Yang, M. R. Lyu, and E. C.-H. Ngai, "A point-distribution index and its application to sensor-grouping in wireless sensor networks," in *Proc. of the 2006 International Wireless Communications and Mobile Computing Conference (IWCMC)*, July 2006.

[4] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proc. of the ACM SenSys*, Los Angeles, CA, November 2003.

[5] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, 2005.

[6] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. of the IEEE ICC*, vol. 2, Helsinki, Finland, June 2001.

[7] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, "Wireless sensor networks with energy efficient organization," *Journal of Interconnection Networks*, vol. 3, no. 3-4, pp. 213–229, December 2002.

[8] M. Cardei and D. Du, "Improving wireless sensor network lifetime through power aware organization," *ACM Journal of Wireless Networks*, vol. 11, no. 3, pp. 333–340, May 2005.

[9] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communication*, October 2000.

[10] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proc. of the IEEE Infocom*, Miami, Florida, March 2005.

[11] R. Kershner, "The number of circles covering a set," *American Journal of Mathematics*, vol. 61, pp. 665–671, 1939.

[12] F. Aurenhammer, "Voronoi diagram - a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 2, pp. 345–405, September 1991.

[13] R. Williams, *The Geometrical Foundation of Natural Structure: A Source Book of Design*. pp. 51-52. New York: Dover Publications Inc., 1979.

[14] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, vol. 2, pp. 84–90, 1960.

[15] J. M. Hammersley, "Monte-Carlo methods for solving multivariable problems," *Annals of the New York Academy of Science*, vol. 86, pp. 844–874, 1960.

[16] Free Software Foundation, "The GLPK (GNU Linear Programming Kit)," http://www.gnu.org/software/glpk/.

[8] The percentage of event-detection failures of the subsets found in this study is less than 0.1%, which is very low. It will not be included for comparison.