# Voronoi-Based Improved Algorithm for Connected Coverage Problem in Wireless Sensor Networks

Jie Jiang, Zhen Song, Heying Zhang, and Wenhua Dou

School of Computer
National University of Defense Technology
410073, Changsha, China
jiangjie@nudt.edu.cn

**Abstract.** In this paper, we consider the connected coverage problem and aim to construct a minimal connected cover set that is sufficient for a given query in wireless sensor networks. We propose a centralized, Voronoi tessellation (CVT) based algorithm to select the minimum number of active sensor nodes needed to cover the target region completely. The constructed sensor set proves to be connected when sensor node's communication range is at least twice of its sensing range. For other situations where the CVT algorithm alone cannot guarantee the network connectivity, we design a Steiner minimum tree (SMT) based algorithm to ensure the network connectivity. Theoretical analysis and simulation results show that our algorithm outperforms the greedy algorithm in terms of both the time complexity and the needed number of sensor nodes that must be kept active to respond to a given query.

## 1 Introduction

Because of advances in micro-sensors, wireless networking and embedded processing, wireless sensor networks (WSN) are becoming increasingly available for commercial and military applications, such as environmental monitoring, biological attack detection, and battlefield surveillance, etc [1, 2].

Minimizing energy consumption to prolong the network lifetime is a major design objective for wireless sensor networks since tiny sensor devices usually operate on limited battery power due to low cost and small size. Many research efforts have been made to improve the energy efficiency of wireless sensor networks [3–7].

In this paper we aim to improve the energy efficiency of wireless sensor network by constructing an energy-efficient network topology. The motivation of our work is the spatial query execution in sensor networks. In the process of spatial query, the sink node needs to gather sensor data within a specific target region in order to obtain the status of the monitored target region. An energy efficient way to respond to the query is to activate only a subset of sensor nodes to monitor the field and send back report data. The active node subset must satisfy two requirements: (1) coverage requirement. Each point in the monitored region must be covered by at least one sensor in this subset. (2) connectivity requirement. The communication network induced by active nodes must be connected.

For a given target region, the problem of finding a minimal subset of sensor nodes which meets the above two requirements at the same time is NP-hard. We propose a centralized Voronoi tessellation (CVT) based heuristic algorithm to compute the near optimal set of sensor nodes needed to cover the target region completely. In case of sensor's communication range is at least twice of its sensing range, the constructed sensor set is connected. In other cases where the CVT algorithm alone cannot guarantee the network connectivity, we design a Steiner minimum tree (SMT) based algorithm to ensure the communication connectivity. The proposed algorithm outperforms the greedy algorithm[9] in terms of both time complexity and the size of active sensor set.

## 2 Problem Statement

We consider static sensor networks in the 2D plane $\mathbb{R}^2$ and use binary sensing model to model sensor node's sensing capability. In this model, the coverage area of a sensor node $s_i$ is a circle centered at $s_i$ with radius $R_s$ (sensing radius). The circular sensing/coverage area is called $s_i$'s *sensing disk* and denoted by $S_i = \left\{p \in \mathbb{R}^2 | d\left(p, s_i\right) \leq R_s\right\}$, where $d$ is the Euclidean distance metric. Each senor node has a limited communication range. $N(i) = \{s_j | d(s_j, s_i) \leq R_c, j \neq i\}$ denotes $s_i$'s communication neighbor sets, where $R_c$ is sensor node's wireless communication radius.

**Definition 1.** *(Connected cover set) A set of sensors $S$ is a cover set for a region $R$ if each point in $R$ is covered at least by one sensor in $S$. If the communication graph induced by the cover set $S$ is connected, set $S$ is called a connected cover set (CCS) for $R$.*

For a given sensor network $S$ deployed in a target region $R$, our goal is to calculate the minimum set of active senor nodes that can both cover the region $R$ completely and form a connected communication network. This problem has been proved to be NP-hard and a greedy algorithm has been proposed in [9]. In this paper, we propose a new, novel algorithm based on Voronoi diagram.

## 3 Voronoi-Based Heuristic Algorithm

The minimal connected cover set is constructed in two steps. First, we construct the minimal cover set based on the Voronoi tessellation of the target region. When $R_c \geq 2R_s$, the constructed cover set proves to be connected. When $R_c < 2R_s$, we design a Steiner minimum tree (SMT) based algorithm to find the additional sensors needed to make the communication graph induced by the cover set connected. Not loosing generality, we assume that the target region is bounded and convex.

### 3.1 CVT Algorithm

The minimal cover set can be obtained by turning off the maximal number of redundant sensors. A sensor is redundant if its sensing coverage is completely subsumed by other sensors. Turning off a redundant sensor will not leave sensing hole in the target region.

**Definition 2.** *(Sensing hole) A sensing hole is a subarea of the target region $R$ where points are not covered by any sensor.*

According to definition 1 and 2, a target region $R$ is completely covered by a sensor network is equal to there is no sensing hole in $R$. The sensing hole is detected by utilizing the Voronoi tessellation [10] of $R$. Given a finite set of sensor nodes $S = \{s_1, \ldots, s_n\}$, the Voronoi region $V_i$ associated with node $s_i$ is defined by $V_i = \{p \in \mathbb{R}^2 | d(p, s_i) \leq d(p, s_j), j \neq i\}$, where $d$ denotes the Euclidean distance. In general, the Voronoi polygon associated with node $s_i$ is unbounded if $s_i$ is on the boundary of the convex hull of $S$ [10]. But when $R$ is bounded, the unbounded Voronoi polygons of sensors on the boundary of the convex hull of $S$ will be bounded by the boundary of $R$.

**Definition 3.** *(Bounded Voronoi tessellation) A Voronoi tessellation with all Voronoi polygons bounded is called a bounded Voronoi tessellation.*

Let $BVT(S, R)$ denote the bounded Voronoi tessellation of the bounded region $R$ when taking $S$ as the generator set, $BVC_i(S, R)$ denote the bounded Voronoi polygon associated with sensor $s_i$, $BVV_i(S, R)$ denote the set of vertices of $BVC_i(S, R)$ and $BVN_i(S, R)$ denote $s_i$'s Voronoi neighbors in $BVT(S, R)$. Since the Voronoi polygon is convex [10] and $R$ is also convex, all the Voronoi polygons in the bounded Voronoi tessellation are convex.

**Theorem 1.** *Consider a sensor network deployed in a bounded, convex region $R$. $R$ is completely covered by $S$ if and only if for each sensor $s_i \in S$, all the vertices of $BVC_i(S, R)$ are within $s_i$'s sensing disk $S_i$.*

*Proof.* (Sufficient condition) For any sensor $s_i \in S$, if all the vertices of $BVC_i(S, R)$ are within $S_i$, then $BVC_i(S, R) \subseteq S_i$. Since $R = \bigcup_{s_i \in S} BVC_i(S, R)$, we have $R = \bigcup_{s_i \in S} BVC_i(S, R) \subseteq \bigcup_{s_i \in S} S_i$. Therefore $S$ is a cover set for $R$.

(Necessary condition) If $S$ is a cover set for $R$, each point in $R$ is covered at least by one sensor in $S$. According to the definition of Voronoi tessellation, each point must be covered by its Voronoi generator. ■

Theorem 1 gives a method to judge whether a sensor node is redundant. Sensor node $s_i$ is redundant if $S \backslash s_i$ is still a cover set for $R$ ($S \backslash s_i$ represents the set of remaining sensors after removing $s_i$ from $S$). That is, for any sensor $s_j \in S \backslash s_i$, all the elements of $BVV_j$ in $BVT(S \backslash s_i, R)$ are within $S_j$. Furthermore, we observe that only $BVC_j(S \backslash s_i, R), j \in BVN_i(S, R)$ may be different from $BVC_j(S, R)$. Other sensors' bounded Voronoi polygons will not change after excluding sensor $s_i$ from $S$ because their Voronoi neighbors don't change. We conclude the above discussion as follows.

**Coverage Redundancy Condition:** Given a bounded convex region $R$ and a sensor network consisting of a set of sensors $S$, sensor $s_i$ is redundant if and only if for any sensor $s_j \in BVN_i\left(S, R\right)$, all the elements of $BVV_j$ in $BVT\left(S \backslash s_i, R\right)$ are covered by $S_j$.

In order to judge whether a sensor node is coverage redundant, the above condition needs to compute the Voronoi tessellation among all remaining $n-1$ sensor nodes. Therefore it will need $n$ times computation of such Voronoi tessellation of the target region $R$ so as to judge the redundant status of each of the $n$ sensor nodes. The time complexity of Voronoi tessellation with $n$ generators is about $O\left(nlogn\right)$ in the worst case. So the total time complexity is about $O\left(n^2logn\right)$ and it is computation expensive. To reduce the computation cost, we propose the following optimization method.

Suppose we are checking whether sensor node $s_i$ is coverage redundant. And $BVN_i\left(S, R\right)$ is $s_i$'s Voronoi neighbor set. For any sensor node $s_j \in BVN_i\left(S, R\right)$, $s_j$'s possible Voronoi neighbor set $BVN_j\left(S \backslash s_i, R\right)$ must be a subset of the union of $BVN_j\left(S, R\right)$ and $BVN_i\left(S, R\right)$ because of the nearest proximity characteristic of Voronoi tessellation, i.e., $BVN_j(S \backslash s_i, R) \subseteq \left(BVN_j\left(S, R\right) \backslash s_i\right) \cup BVN_i\left(S, R\right)$. This implies that $BVC_j(S \backslash s_i, R) = BVC_j\left(U_j, R\right)$, where $U_j = BVN_j\left(S, R\right) \backslash s_i \cup BVN_i\left(S, R\right)$. Therefore, it is sufficient to compute the Voronoi tessellation of $R$ taking $U_j$ as the generator set while determining whether $s_i$ is coverage redundant. Using this optimization, the time complexity can be reduced to $O\left(nklogk\right)$, where $k = \max\left(\|U_j\|\right) \ll n$.

We propose the optimized coverage redundant condition as follows.

**Optimized Coverage Redundancy Condition:** Given a bounded convex target region $R$ and a sensor network consisting of a set of sensors $S$, sensor $s_i$ is redundant if and only if for any sensor $s_j \in BVN_i\left(S, R\right)$, all the elements of $BVV_j\left(U_j, R\right)$ are covered by $s_j$, where $U_j = BVN_j\left(S, R\right) \backslash s_i \cup BVN_i\left(S, R\right)$.

After all redundant sensors have been identified, we further classify redundant sensors into two sub-categories: independently and dependently redundant sensor.

**Definition 4.** *(Independently/Dependently redundant sensor) An independently redundant sensor is a redundant sensor whose Voronoi neighbors are all non-redundant. A dependently redundant sensor is a redundant sensor whose Voronoi neighbors include at least one redundant sensor.*

Non-redundant sensors must be kept active and independently redundant sensors are safe to be turned off. If two dependently redundant sensors, which are Voronoi neighbors each other, are turned off simultaneously, an area between them may be left uncovered, thus a sensing hole will occur. So the problem of calculating the minimal set of the active sensors is equivalent to determining the maximal set of the dependently redundant sensors that can be turned off simultaneously. In our solution, we use Redundant Dependency Graph to resolve the dependency relationship.

**Definition 5.** *(RDG graph) A Redundant Dependency Graph,* $RDG = (V_r, E_r)$*, is an undirected graph, where* $V_r$ *is the set of all dependently redundant sensors and* $\forall v_{ri}, v_{rj} \in V_r, (v_{ri}, v_{rj}) \in E_r$ *if and only if* $v_{ri}$ *and* $v_{rj}$ *are Voronoi neighbors.*

The problem of determining the maximal set of dependently redundant sensors that can be deactivated simultaneously is equal to the problem of calculating the maximal independent set (MIS) of the RDG graph. In this paper, we adapt the greedy algorithm in [11] to calculate the MIS of the RDG graph.

When CVT algorithm terminates, all independently redundant sensors and all dependently redundant sensors selected into the MIS of the RDG graph form a Safe Set (*SS*), which includes all sensors that can be deactivated safely at the same time. And all other sensors form the final cover set for the target region and must be kept active to respond a specific query.

The pseudocode of the CVT algorithm is presented in Algorithm 1. The input of the CVT algorithm are the initial set of sensors $IS = \{s_1, \ldots, s_n\}$ and the target region $R$. Its output is a cover set ($CS$) for $R$. In the description, *IRS* denotes Independently Redundant sensor Set, *DRS* denotes Dependently Redundant sensor Set, *SS* denotes Safe sensor Set, *S* denotes temporary sensor Set and *MIS* denotes Maximal Independent Set.

**Algorithm 1** CVT Algorithm

$SS = CS = \emptyset, S = IS$
**if** $R \not\subset \bigcup_{s_i \in S} S_i$ **then**
Return $CS = \emptyset$
**end if**
**while** $S \neq \emptyset$ **do**
$IRS = DRS = MIS = \emptyset$
Identify all redundant sensor nodes in $S$
**for** each redundant sensor $s_i \in S$ **do**
**if** $s_i$ is independently redundant **then**
$IRS = IRS \bigcup \{s_i\}$
**else**
$DRS = DRS \bigcup \{s_i\}$
**end if**
**end for**
**if** $DRS \neq \emptyset$ **then**
Compute $MIS$ of the RDG graph
$SS = SS \bigcup IRS \bigcup MIS$
$S = IS - SS$
**else**
$SS = SS \bigcup IRS$
Break out the **while** loop
**end if**
**end while**
$CS = IS - SS$

The approximation performance of the CVT algorithm is determined by the greedy algorithm for the MIS problem. It has been shown that in any graph with $N$ vertices, $M$ edges and average degree $\delta$, the greedy algorithm can take at most $O(M)$ steps to find an independent set of size at least $N/(\delta+1)$[11].

In the following, we analyze the time complexity of the CVT algorithm.

**Lemma 1.** *The maximum possible number of Voronoi vertices in a two dimensional Voronoi tessellation is* $2n-5$*, and the maximum possible number of Voronoi edges is* $3n-6$*, where* $n$ *is the number of generators [10].*

**Lemma 2.** *The average number of Voronoi edges per Voronoi polygon does not exceed 6 [10].*

**Lemma 3.** *Any algorithm for constructing Voronoi tessellation requires at least* $O(nlogn)$ *time in the worst case, where* $n$ *is the number of generators [10].*

**Theorem 2.** *The worst-case time complexity of the CVT algorithm is less than* $O(mnk\log k)$*, where* $k$ *is the maximum possible number of Voronoi neighbor of each sensor,* $n$ *is the total number of sensors initially deployed in the network,* $m$ *is the maximum iterative times of the* ***While*** *loop of the CVT algorithm and* $m \leq \log_{7/6} n$*.*

*Proof.* Step $2 \sim 4$ can be completed in $O(n\log n)$. In the **While** loop, step 7 can be done in $O(nk\log k)$ using the optimized coverage redundancy condition, where $k$ is the maximum number of Voronoi neighbors. Step $8 \sim 14$ need $O(n)$ to classify redundant sensors. Step 16 needs $O(l)$ to calculate the MIS of RDG graph, where $l$ is the edge number of RDG and $l \leq 3n-6$. Therefore one pass of the **While** loop can be done at most in $O(nk\log k)$ in the worst case. Suppose the total iterative number of **While** loop is $m$. According to the approximation ratio of the greedy algorithm for the MIS problem, we can get $m \leq \log_{7/6} n$. The total time complexity of the CVT algorithm is dominated by step $5 \sim 23$ and thus is less than $O(mnk\log k)$. ■

### 3.2 SMT Based Connection Algorithm

It has been shown that the complete coverage of a convex region implies connectivity of the communication network induced by all active sensor nodes if $R_c \geq 2R_s$ [12]. Therefore the cover set constructed by the CVT algorithm is connected if $R_c \geq 2R_s$. But when $R_c < 2R_s$, the communication connectivity of the cover set can't be ensured. In this subsection, we propose a Steiner Minimum tree (SMT) based connection algorithm to make the cover set connected if necessary.

**Definition 6.** *(Primary/Assistant sensor) A sensor in the cover set is called a primary sensor. Otherwise it is an assistant sensor.*

**Definition 7.** *(Connected component) A connected component in the cover set is a connected subgraph of* $G_c$ *whose vertices only involve primary sensors, where* $G_c$ *is the communication graph induced by all initially deployed sensors.*

Suppose that sensors in the cover set constructed by the CVT algorithm form $m$ separated connected components in $G_c$ as shown in Fig.1. The connected component consists of only primary sensors (denoted by solid circular dots). These components are connected through assistant sensors (denoted by hollow circular dots). We abstract each connected component as one virtual vertex and use $v_i$ to represent $comp_i$. Then we attach to $v_i$ all edges in $G_c$ that connect one primary sensor in $comp_i$ and another assistant sensor. After this conversion, we get an abstract communication graph ${G_c}'$ (see Fig.2).

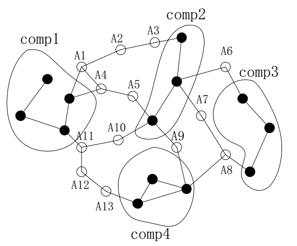


Fig. 1: Connected Components in $G_c$

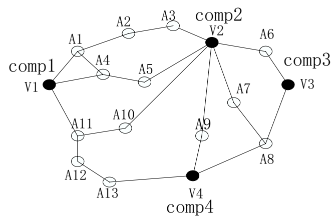


Fig. 2: Abstract Communication Graph ${G_c}'$

If we assign weight 1 to each edge in ${G_c}'$, selecting the minimal number of additional sensors needed to connect the components in $G_c$ is equivalent to solving the Steiner Minimum Tree problem in the 1-weighted graph ${G_c}'$, where the virtual vertices representing components are considered as "terminals" and the needed assistant sensors are those "Steiner points" that are needed to construct the Steiner Minimum Tree to connect all the terminals. We present the SMT-based connection algorithm in Algorithm 2.

**Algorithm 2** SMT-based Connection Algorithm

1: Create the 1-weighted abstract communication graph ${G_c}'$ as described
2: Calculate the Steiner Minimum Tree (SMT) to connect all virtual vertices in ${G_c}'$.
3: Return all the assistant sensors corresponding to the Steiner points in the computed SMT.

## 4 Performance Evaluation

To evaluate the performance of the proposed algorithms, we have constructed a simulator in C programming language under Linux operating system. The target region is a square of size $400 \times 400$. The default sensing radius is 30. We change the communication radius to evaluate the performance under different ratio of

$R_c$ to $R_s$. To ensure the target region is completely covered by the initial sensor network and the initial network is connected when all sensors are active, we first divide the region into grid cells with dimension of 40 and deploy one sensor at the center of each grid cell. In this deployment, the minimal communication range that can ensure network connectivity is 40. When the sensing range is 30, each sensor can only cover the grid where it is located. So we can expect the minimal size of the cover set is about 100 when the sensing range is 30. Besides the 100 fixed sensors, we also randomly deploy lots of sensors in the region.

### 4.1 Comparison with Greedy Algorithm

The greedy algorithm in [9] is also a centralized algorithm. This algorithm firstly needs to divide the target region into many subelements, where points in the same subelement are covered by the same set of sensors.

As stated in [9], the maximum number of subelements on a 2-demensional plane with $n$ circles is $n\left(n+1\right)$. When there are totally $K$ sample points needed to be examined, the preprocessing of generating subelements takes $O\left(Kn^3\right)$ [6]. The number of sample points is related to the size of the region and the grid cell. And the size of the grid cell is closely related to the coverage accuracy. Larger cell size can result in smaller $K$, thus less time is needed, but the coverage performance becomes worse. In application that requires every point is covered at least by one sensor, the size of the grid cell must be small enough. On the other hand, the region monitored by sensor networks is usually vast. Therefore the number of sample points is usually several orders of magnitude of the number of the sensors in the network, i.e., $K \gg n$. Therefore the time complexity of the greedy algorithm is much larger than that of CVT algorithm.

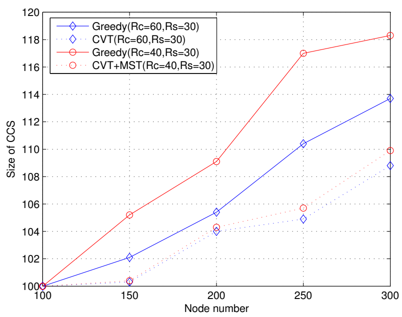


Fig. 3: Size of CCS

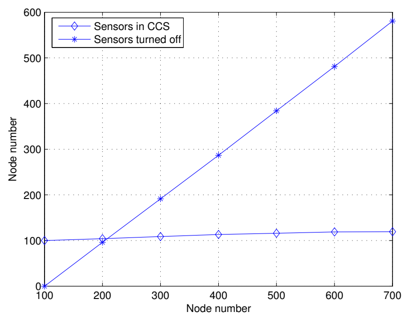


Fig. 4: Size of CCS and number of off-duty nodes vs. node density ($R_s = 30, R_c = 60$)

The comparison of the size of the connected cover set constructed by the greedy algorithm and our proposed algorithms is illustrated in Fig.3. It is clear that both the CVT and CVT+SMT algorithms generate much smaller CCS than the greedy algorithm when $R_s = 30$ and $R_c = 60, 40$ respectively. While the algorithm in [9] is globally greedy, our proposed algorithm can definitely identify all non-redundant sensors that must be kept active, all independently redundant sensors that can be safely turned off, and only apply the greedy algorithm to determine the maximal number of dependently redundant sensors that can be turned off simultaneously.

In summary, our proposed algorithm outperforms the greedy algorithm in terms of both the time complexity and the size of the final connected cover set (i.e., the number of sensors that must be kept active).

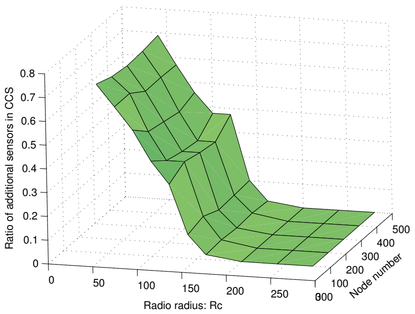

Fig. 5: Ratio of additional sensor in final CCS

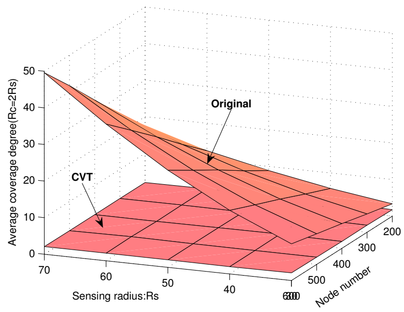

Fig. 6: Average coverage degree vs. node density

### 4.2 Simulation Results

Fig.4 shows how the size of CCS and the number of the sensors turned off vary when the number of deployed sensors changes from 100 to 700. Initially, when there are only 100 fixed sensors in the region, all sensors must be active to provide the coverage and connectivity. We observe that the size of CCS constructed by the CVT algorithm almost keeps constant and is very close to the expected number of active sensors, i.e., 100 when $R_c \geq 2R_s$. Note these 100 sensors are not the same as the 100 fixed sensor nodes.

When $R_c < 2R_s$, more additional sensors are needed to maintain the communication connectivity. Fig.5 shows the ratio of additional sensor nodes calculated by the SMT connection algorithm in the final CCS under different node density and sensing radius.

The average coverage degree can reflect the sensing redundancy of the network. To calculate the sensing degree, we divide the region into $1 \times 1$ unit cells and assume an event occurs at the center of each cell. The coverage degree is defined as the number of sensors that can detect the event. As illustrated in

Fig.6, although the original sensing degree varies from 3 to 50, our proposed CVT algorithm can result in about 2 degree when $R_c = 2R_s$.

## 5 Conclusions

In this paper, we have designed centralized algorithms to construct a minimal connected cover set for wireless sensor networks. The basic idea is to calculate the cover set firstly and then make it connected if needed. We propose a centralized, Voronoi tessellation (CVT) based algorithm to calculate the near optimal cover set for the target region. The constructed cover set is connected if $R_c \geq 2R_s$. In case of $R_c < 2R_s$ where the cover set constructed by the CVT algorithm is not connected, we design a Steiner minimum tree (SMT) based algorithm to compute the minimal set of additional sensors needed to make the cover set connected. Theoretical analysis and simulation results show that our proposed algorithm outperforms the greedy algorithm.

## References


1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Computer Networks (Elsevier). Vol.38 (4). March. 2002.
2. J. Elson and D. Estrin, Wireless Sensor Networks(Kluwer), 3-20 2004
3. S. Slijepcevic and M. Potkonjak. Power Efficient Organization of Wireless Sensor Networks. in Proc. of IEEE ICC'01, Helsinki, Finland, 2001.
4. F. Ye, G. Zhong, S. Lu, and L. Zhang. Peas: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks. in Proc. of ICDCS, 2003.
5. D. Tian and N. D. Georganas. A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks. in Proc. of WSNA'02, Atlanta, Georgia, USA, 2002.
6. H. Chen, H. Wu, and N. Tzeng. Grid-Based Approach for Working Node Selection in Wireless Sensor Networks. in Proc. of IEEE International Conference on Communications (ICC), Paris, France, 2004.
7. B. Carbunar, A. Grama, J. Vitek and O. Carbunar, *Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 377-386 (2004).
8. T. Yan, T. He, and J. Stankovic. Differentiated Surveillance Service for Sensor Networks. in Proc. of SenSys'03, Los Angels, CA, USA, 2003.
9. H. Gupta, S. R. Das, and Q. Gu. Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. in Proc. of MobiHoc'03, Annapolis, Maryland, USA, 2003.
10. A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. Spatial Tessellations: Concepts and Applications of Voronoi Diagram. John Wiley & Sons, New York, 1999.
11. D. S. Hochbaum. Approximation Algorithms for Np-Hard Problems. PWS Publishing Company, 1995.
12. H. Zhang and J. C. Hou. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Proceedings of NSF International Workshop on Theoretical and Algorihtmic Aspects of Sensors, Ad Hoc Wireless, and Peer-to-Peer Networks, 2004.