Transforming Complete Coverage Algorithms to Partial Coverage Algorithms for Wireless Sensor Networks

Yingshu Li, *Senior Member*, *IEEE*, Chinh Vu, *Student Member*, *IEEE*, Chunyu Ai, *Member*, *IEEE*, Guantao Chen, *Member*, *IEEE*, and Yi Zhao

Abstract—The complete area coverage problem in Wireless Sensor Networks (WSNs) has been extensively studied in the literature. However, many applications do not require complete coverage all the time. For such applications, one effective method to save energy and prolong network lifetime is to partially cover the area. This method for prolonging network lifetime recently attracts much attention. However, due to the hardness of verifying the coverage ratio, all the existing centralized or distributed but nonparallel algorithms for partial coverage have very high time complexities. In this work, we propose a framework which can transform almost any existing complete coverage algorithm to a partial coverage one with any coverage ratio by running a complete coverage algorithm to find full coverage sets with virtual radii and converting the coverage sets to partial coverage sets via adjusting sensing radii. Our framework can preserve the characteristics of the original algorithms and the conversion process has low time complexity. The framework also guarantees some degree of uniform partial coverage of the monitored area.

Index Terms—Partial coverage, wireless sensor networks, energy efficiency.

1 Introduction

Researchers have spent lots of effort to design algorithms to completely cover an area—*complete coverage* problem. Most of the coverage-related works concern how to prolong network lifetime through different techniques. One of the techniques which recently attract researchers' attention is to reduce the coverage quality to trade for network lifetime. In some applications, the required coverage quality may even be different at different points of time. For example, *forest fire monitoring* applications [13], [14], and [15] may require complete coverage in dried seasons, while they only require 80 percent of the area to be covered in rainy seasons. Thus, to extend network lifetime, we can lower the coverage quality if it is acceptable. The problem to cover only a portion of an area is referred as the *"partial coverage"* problem. The requirement for the partial coverage problem is that the ratio of the covered area over the whole monitored area is no less than a predefined value. This value is a user-specified parameter. In this work, we use the notation α to refer to this parameter. Consequently, the partial coverage problem is also referred as α-coverage problem of which the objective is to cover only α-portion of the area (the formal definition of this problem is given in Definition 2). Moreover, it is always desirable to schedule the sensors such that the area is uniformly covered. It is clearly undesired if the network only covers some particular subregions of the area while uncovers the other large and continuous subregions. To evaluate coverage quality, a metric named as *Sensing Void Distance* (SVD) [11], [18] is used (the formal definition of this metric is given in Definition 6).

In this work, we solve the α-coverage problem by transforming various well-known complete coverage algorithms to partial coverage algorithms. Our framework has four strategies: two general strategies and two extended strategies. The two general strategies are designed for networks where sensors have fixed sensing ranges (fixed-sensing-range network) and the other two are for networks where sensors can adjust their sensing ranges (adjustable-sensing-range network). For any particular α, the two general strategies also guarantee a constant bound of SVD.

2 Related Work

The problem of partial coverage is recently investigated in the literature under many alias such as "α-lifetime" [8], [9], "p-percentage coverage" [10], [18], "θ-coverage" [11], or "q-portion coverage" [16]. The problem requires the network to cover at least "p percent," "α portion," or "q portion" of an area. In other words, if $\alpha = \theta = q = \frac{p}{100}$, those problems are actually the same. To be consistent with the name when this problem was first proposed [8], [9] and to make the term self-explaining, we use the name "α-coverage" (which will be formally defined later in Definition 2) to refer to this problem and α is referred as *coverage ratio*.

The work in [8] shows the upper bound of network lifetime when only α-portion of the whole area is covered. It shows that network lifetime may increase up to 15 percent for

- *Y. Li, C. Vu, and C. Ai are with the Department of Computer Science, Georgia State University, Atlanta, GA 30303. E-mail: {yli, chinhvtr, chunyuai}@cs.gsu.edu.*
- *G. Chen and Y. Zhao are with the Department of Mathematics and Statistics, Georgia State University, Atlanta, GA 30303. E-mail: {matgtc, matyxz}@langate.gsu.edu.*

Manuscript received 5 Dec. 2009; revised 24 Mar. 2010; accepted 28 Apr. 2010; published online 2 June 2010.
Recommended for acceptance by J. Xiaohua.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-12-0601.
Digital Object Identifier no. 10.1109/TPDS.2010.124.

99 percent coverage and 25 percent for 95 percent coverage. Then, in [9], the authors proposed a centralized algorithm to solve the α-coverage problem of which the increment of network lifetime is close to the upper bound derived in [8]. In [16], a centralized algorithm based on the Garg-Könemann method for q-portion coverage is proposed. The algorithm has a performance ratio of $(1+\epsilon)(1+ln\frac{1}{1-q})$, for any $\epsilon > 0$.

In [12], percentage coverage instead of complete coverage is selected as the design goal, and a location-based Percentage Coverage Configuration Protocol (PCCP) is developed to assure that the proportion of the area after configuration to the original area is no less than a desired percentage.

Liu and Liang [11] presented a centralized algorithm which takes both coverage and connectivity into account. Their work is the first one to analyze partial coverage properties in order to prolong network lifetime. Initially, active sensors are randomly selected. In each iteration, nodes on a chosen candidate path with the maximum gain are chosen. The algorithm continues until the whole area is θ-covered. This method is also employed by [10]. The work in [10] proposed two algorithms, one centralized and one distributed, for the same problem. To provide different coverage qualities at different locations of the monitored area, the area is partitioned into a number of clusters and the algorithms partially cover the clusters one by one.

For the α-coverage problem, we always want to know how uniformly the subregions are covered. To evaluate this, adapted from [11], the work in [18] uses *SVD* which is the distance from an uncovered point to a nearest covered point. The authors in [18] also claimed that their CDS-based distributed algorithm CpPCA-CDS can provide a constant-bounded SVD. However, the value of p is too small that even a subset of a CDS can provide p-percentage coverage; thus, coverage redundancy is high to guarantee a bounded SVD. In other words, in [18], the coverage redundancy is the price paid for a bounded SVD.

To the best of our knowledge, most proposed algorithms for α-coverage are centralized ones. There are distributed algorithms discussed in [10], [18]. However, those algorithms work in a distributed manner but not a parallel fashion, i.e., each sensor has to wait for the value of α to be calculated by its neighbors to decide whether to be active or to sleep. So, the time complexity may be very high. In the worst case, the time complexity of a nonparallel (consequential) algorithm may be of the order of the network size.

The rest of the paper is organized as follows: Section 3 states the motivation of our work. Section 4 introduces some preliminary concepts and supporting knowledge. In Section 5, we explain our framework in details. We then evaluate the effectiveness of the proposed framework in Section Simulation of *Supplementary File*, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.124. We conclude our work in Section 6.

3 Motivation

Most of the existing algorithms for the α-coverage problem are greedy on a so-called *contribution* (or *gain*). *Contribution* of a set X of sensors is a parameter that mainly depends on the current uncovered area that can be covered by X. For the simplest case where sensors' sensing regions are assumed to be perfect disks, that region usually is the part of union of disks that is not covered by some other overlapping disks (neighbors). However, calculating the area of this union region is not trivial even for the case that all the sensors have the same sensing range. The previous works usually ignore to explain how to do that calculation. To the best of our knowledge, there is no existing method to calculate the exact area for such region.

Besides, most of the current works directly solve the partial coverage problem. The common method is to greedily (on *contribution*) add sensors until at least α portion of the area is covered. As a result, most of the existing algorithms are centralized. The rest, few distributed algorithms have high time complexity due to the fact that they have to scan through all the sensors one by one until α-portion of the area is covered. That means the sensors cannot work in a parallel manner. In a sense, we claim that the α-coverage problem is an *impossible-to-directly-solve* problem in a distributed and parallel manner due to the fact that α is a global parameter which cannot be acquired in a parallel manner.

Moreover, there are a great deal of existing algorithms designed for complete coverage which motivate us to find a way to utilize them for partial coverage. In the literature, there do exist many fully distributed and parallel algorithms which do not depend on *contribution* such as the algorithms in [2], [3], [4], [5], [6], [7]. It is a significant contribution if we somehow convert those algorithms to solve partially (instead of completely) area cover problems. The resulting algorithms have to guarantee some level of coverage quality as required by users. The conversion should preserve the characteristics of the original algorithms. Moreover, the conversion should be simple and fast enough to not increase time complexity too much. Importantly, the conversion process should work with most of the existing complete coverage algorithms.

In this work, we propose a framework that satisfies all the requirements of an algorithm conversion framework we just mentioned above. The framework consists of four strategies (two general strategies and the other two extended strategies) which are designed for different kinds of networks. Two (one general and one extended strategy) are designed for fixed-sensing-range WSNs and the other two are for adjustable-sensing-range WSNs. Amazingly, for any certain desired coverage ratio α and a particular WSN, the resulting algorithm of two general strategies for partial coverage can guarantee a constant-bounded SVD for all the original algorithms for complete coverage. In other words, the resulting algorithms of two general strategies uniformly cover the area. Also, the general strategies work for almost all complete coverage algorithms.

4 Preliminary

We dedicate this section to introduce some concepts and definitions.

Definition 1 (α-cover). *Given a real number α where $0 < \alpha < 1$, a two-dimensional region A and set $\mathcal{S}$ of n sensors s_i for $i = 1 \dots n$. Sensor s_i's sensing region is S_i. $\|A\|$ denotes the area of region A. A subset $\mathcal{C} \subseteq \mathcal{S}$ α-covers area A ($\mathcal{C}$ is an α-set-cover of A) if:*

$$\left\| \left(\bigcup_{s_i \in \mathcal{C}} S_i \right) \bigcap A \right\| \geq \alpha \|A\|. \tag{1}$$

By this definition, the traditional set cover is also called *1-set-cover* in this work. Now, we define the *α-coverage problem* as following:

Definition 2 (α-coverage problem). *Given a real number α where $0 < \alpha < 1$, a two-dimensional region A and set $\mathcal{S}$ of n sensors s_i for $i = 1 \dots n$. Find a set of α-set-covers $C_1, \dots, C_l$ of $\mathcal{S}$ for region A.*

Definition 3 (γ-virtual network). *For a particular real number γ where $0 < \gamma$ and a WSN $\mathcal{S}$ with n sensors $s_1, \dots, s_n$.*

- *If $\mathcal{S}$ is a fixed-sensing-range WSN where sensor s_i ($i = 1 \dots n$) has a sensing range R_i, then the γ-virtual network of $\mathcal{S}$, denoted by $\mathcal{S}^\gamma$, is the network where sensor s_i has a sensing range $\frac{R_i}{\sqrt{\gamma}}$. This sensing range is called the virtual sensing range.*
- *If $\mathcal{S}$ is an adjustable-sensing-range WSN where sensor s_i ($i = 1 \dots n$) has the maximum sensing range of $MaxR_i$, then:*
  - *The γ-virtual network of $\mathcal{S}$, denoted by $\mathcal{S}^\gamma$, is the adjustable-sensing-range network where sensor s_i has maximum sensing range of $\frac{MaxR_i}{\sqrt{\gamma}}$ for $i = 1 \dots n$. This maximum sensing range is called virtual maximum sensing range.*
  - *The fixed-γ-virtual network of $\mathcal{S}$, denoted by $\mathcal{S}^\gamma_{fixed}$, is the fixed-sensing-range network where sensor s_i has sensing range of $\frac{MaxR_i}{\sqrt{\gamma}}$. This sensing range is called the virtual sensing range.*

Because we have to deal with two types of networks in this paper (real and virtual networks), it is necessary to distinguish two types of set covers corresponding to the two types of networks as follows:

Definition 4 (VSC^γ: γ-virtual-set-cover). *Given a real number γ ($0 < \gamma$) and set $\mathcal{S}$ of n sensors s_i ($i = 1 \dots n$), the γ-virtual network of $\mathcal{S}$ is $\mathcal{S}^\gamma$. A 1-set-cover of $\mathcal{S}^\gamma$ is called γ-virtual-set-cover, denoted by VSC^γ.*

Definition 5 (μ-RSC: μ-real-set-cover). *Given a real number μ ($0 < \mu$) and set $\mathcal{S}$ of n sensors s_i ($i = 1 \dots n$). The μ-real-set-cover, denoted by μ-RSC, of a virtual set cover C_j (C_j is not necessarily a VSC^γ) is the set cover where every sensor has a sensing range of $\sqrt{\mu}$ times of its sensing range in C_j. Furthermore, a μ-RSC is feasible if either of the following conditions is true:*

- *If the network is a fixed-sensing-range network, then every sensor of μ-RSC must have its predefined sensing range.*
- *If the network is an adjustable-sensing-range network, then every sensor of μ-RSC must have its sensing range no larger than its predefined maximum sensing range.*

5 A Framework for α-Coverage Algorithms

The framework requires three inputs: 1) the coverage ratio α, 2) a complete coverage algorithm $\mathbb{A}$ (sometimes referred as "*original algorithm*"), and 3) the network $\mathcal{S}$ consisting of n sensors $\mathcal{S} = \{s_1, .., s_n\}$. Our framework transforms the input original algorithms to the ones that can generate a set of α-set-covers. We sometimes refer the obtained algorithms as "*α-coverage algorithms*."

5.1 Assumptions and Notations

We assume that the sensing region of a sensor s_i is a disk centered at s_i. If s_i has a sensing range of R_i, denoted by $s_i(R_i)$, then that disk has a radius of R_i.

For an input algorithm $\mathbb{A}$, conventionally, the result of $\mathbb{A}$ to completely cover a sensor network $\mathcal{S}$ is a set of $\{C_j, t_j\}$ pairs where each C_j is a 1-set-cover and t_j is its working schedule. Each set cover C_j is a set of sensors (with their sensing ranges) that will be turned on to provide complete coverage. The schedule t_j of set cover C_j may include the starting time points and the durations that the set cover C_j will be activated. It is worth emphasizing that an algorithm $\mathbb{A}$ does not always explicitly create a set of all desired set covers and return them as an output. For instance, the family of scheduling algorithms that work in rounden, of which the network working time is divided into equal length rounds and the algorithms create a set cover for each round, they just create one set cover at a time for each round. We make the assumption about the result of the original algorithms only for clear and easier explanation of our framework.

For an adjustable-sensing-range WSN $\mathcal{S} = \{s_1, .., s_n\}$, we assume that each sensor s_i is able to smoothly adjust its sensing range under some upper cut-off range. This assumption is employed by most works concerning adjustable-sensing-range WSNs such as [3] and [4].

Our framework has no restriction on the type of WSNs. In terms of sensors' sensing ranges and initial energy, the WSNs may be heterogeneous or homogeneous, and the network may be fixed-sensing-range or adjustable-sensing-range networks. However, the algorithms generated by our framework have the same restriction as the original algorithms.

5.2 Basic Idea of the Framework

Before explaining our strategies in detail, it is necessary to emphasize the essences of our framework. Our framework essentially converts a complete coverage algorithm to a resulting algorithm for the α-coverage problem. The framework first makes the original algorithm to be executed on a virtual network. The result of that execution is a set of virtual set covers C_j and their schedules t_j. These set covers are for the virtual network, so the sensors of those set covers may have virtual sensing ranges which might be bigger than their real maximum sensing ranges. The framework then modifies the original algorithm so that the final result is a set of real set covers $\overline{C_j}$, where each set cover $\overline{C_j}$ is α-RSC of C_j. Though in Section 5.3.1, Section 5.3.2, and Section Extended strategies of *Supplementary File*, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.124, our strategies are actually the general guidelines on how to modify an original algorithm (for complete coverage) to obtain a desired algorithm for α-coverage. When modifying an original algorithm, every step does not have to be exactly the same as shown in the pseudocodes:

- The process to modify the algorithm $\mathbb{A}$ is not always the same. For example, instead of creating a new virtual network, our framework suggests that the original algorithm should be modified in the way that anytime sensors' sensing ranges are referenced by the algorithm $\mathbb{A}$, and the strategies just have to replace the original sensing ranges with the corresponding virtual sensing ranges.

- Since there exist complete coverage algorithms that do not generate all set covers at once, their resulting algorithms for the α-coverage problem do not have to explicitly identify all α-set-covers. Most of the time, when the algorithm $\mathbb{A}$ generates a set cover for the virtual network, the algorithm $\mathbb{A}$ will be modified in the way that right before a virtual set cover is returned, it will be converted to a real set cover.

5.3 General Strategies

5.3.1 General Strategy for Fixed-Sensing-Range WSNs—*Strategy G-1*

Assume that a sensor s_i $(i = 1, \ldots, n)$ has a fixed sensing range R_i. Strategy G-1 is given in Algorithm 1 of *Supplementary File*, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.124.

This strategy works for both types of original algorithms $\mathbb{A}$:

- **Type 1**: Algorithm $\mathbb{A}$ is designed for fixed-sensing-range WSNs. Strategy G-1 first runs the original algorithm $\mathbb{A}$ on α-virtual network $\mathcal{S}^\alpha$ where sensor s_i has a sensing range of $\frac{R_i}{\sqrt{\alpha}}$ to get a set of virtual set covers C_j and their schedules t_j. For a sensor $s_i \in C_j$, its virtual sensing range in C_j is $\frac{R_i}{\sqrt{\alpha}}$. From virtual set cover C_j, the set cover $\overline{C_j}$ is created of which each sensor s_i has a sensing range of R_i, which is s_i's real sensing range. It can be seen that each set cover C_j is a VSC^α and set cover $\overline{C_j}$ is an α-RSC of C_j. The final result is a set of set covers $\overline{C_j}$ where sensors use their real sensing ranges R_i and their schedule t_j for $j = 1 \ldots l$.
- **Type 2**: Algorithm $\mathbb{A}$ is designed for adjustable-sensing-range WSNs. Strategy G-1 runs the original algorithm $\mathbb{A}$ on α-virtual network $\mathcal{S}^\alpha$ where sensor s_i has the *maximum* sensing range of $\frac{R_i}{\sqrt{\alpha}}$ to get a set of $\{C_j, t_j\}$ pairs. Denote the sensing range of a sensor s_i in C_j as R_i^j and $R_i^j \le \frac{R_i}{\sqrt{\alpha}}$. In the final resulting set covers $\overline{C_j}$, sensor $s_i \in \overline{C_j}$ uses its real sensing range R_i (i.e., we intentionally ignore the virtual sensing ranges assigned by the algorithm $\mathbb{A}$) and schedule t_j. Usually, it is not practical to execute Type 2 algorithms on fixed-sensing-range WSNs. We just list this case here to illustrate the universality of our framework. We do not recommend Strategy G-1 to be applied for adjustable-sensing-range algorithms on fixed-sensing-range WSNs.

Since Strategy G-1 forces sensors of the final resulting set covers $\overline{C_j}$ to use their original sensing ranges (without considering the sensors' sensing ranges in the virtual set cover C_j), the following lemmas are directly derived:

Lemma 1. *Given a real number α $(0 < \alpha < 1)$ and a fixed-sensing-range WSN, for any original (complete coverage) algorithm as an input, the α-coverage algorithm derived by Strategy G-1 generates feasible set covers whose sensors' sensing ranges are their original sensing ranges.*

Lemma 2. *The sensing range of a sensor in a final result's set cover $\overline{C_j}$ is no smaller than $\sqrt{\alpha}$ times of its sensing range in the corresponding virtual set cover C_j.*

5.3.2 General Strategy for Adjustable-Sensing-Range WSNs—*Strategy G-2*

We assume that each sensor s_i's sensing range has a predefined upper bound $MaxR_i$. Strategy G-2 is given in Algorithm 2 of *Supplementary File*, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.124, and works for both types of original algorithms.

- **Type 1**: The original algorithm $\mathbb{A}$ is designed for fixed-sensing-range WSNs. Strategy G-2 runs the original algorithm $\mathbb{A}$ on fixed-β-virtual network $\mathcal{S}^\beta_{fixed}$ where $\alpha \le \beta$. That is, execute algorithms $\mathbb{A}$ on a virtual network where a sensor s_i has a fixed sensing range of $\frac{MaxR_i}{\sqrt{\beta}}$. The result of this execution is a set of virtual set covers C_j, and each C_j is a VSC^β. Then, it adjusts each sensor's the sensing range from $\frac{MaxR_i}{\sqrt{\beta}}$ in $VSC^\beta C_j$ to
$$R_i = \sqrt{\alpha} \times \frac{MaxR_i}{\sqrt{\beta}} = \frac{\sqrt{\alpha}}{\sqrt{\beta}} \times MaxR_i$$
in α-RSC $\overline{C_j}$. Since $\alpha \le \beta$, $R_i \le MaxR_i$, which makes the final resulting set covers $\overline{C_j}$ feasible. If we choose $\beta = \alpha$, then this strategy is similar as Strategy G-1.
- **Type 2**: The original algorithm $\mathbb{A}$ is designed for adjustable-sensing-range WSNs. Strategy G-2 runs the original algorithm $\mathbb{A}$ on β-virtual network $\mathcal{S}^\beta$, where $\alpha \le \beta$ to decide sensors' sensing ranges and their schedules. Let R_i^j be the sensing range that $\mathbb{A}$ decides for sensor s_i in a virtual set cover C_j, clearly $R_i^j \le \frac{MaxR_i}{\sqrt{\beta}}$. Then, each sensor's sensing range is adjusted to $\sqrt{\alpha}R_i^j$ in the final resulting set cover $\overline{C_j}$. Because
$$\alpha \le \beta, \sqrt{\alpha}R_i^j \le \sqrt{\alpha} \times \frac{MaxR_i}{\sqrt{\beta}} \le MaxR_i.$$
Thus, the set cover $\overline{C_j}$ is a feasible set cover.

Based on the explanation above, the following lemmas hold:

Lemma 3. *Given a real number α where $0 < \alpha < 1$ and an adjustable-sensing-range WSN, for any original complete coverage algorithm, Strategy G-2 generates feasible set covers whose sensors' sensing ranges are no larger than their predefined upper bound.*

Lemma 4. *In a final resulting set cover $\overline{C_j}$, each sensor's sensing range is exactly $\sqrt{\alpha}$ times of its sensing range in the corresponding virtual set cover C_j.*

5.3.3 Analysis

Before showing the correctness of the two general strategies, we first introduce a supporting lemma:

Lemma 5. *Given a real number α $(0 < \alpha < 1)$, an infinite monitored area, and a WSN where sensors' sensing regions are disks, assume that the network can completely cover the monitored area. If the sensing ranges of all the active sensors shrink down with the ratio $\sqrt{\alpha}$, then the network with the new sensing range assignment can α-cover the monitored area.*

Proof. Let $\delta = \sqrt{\alpha}$, then this lemma is a direct result of the δ-*compression* theorem (Theorem 11 in Section 5.3.4). □

Theorem 6 (Correctness). *Given a real number α $(0 < \alpha < 1)$ and an infinite monitored area, the proposed two general strategies work correctly.*

Proof. To prove the correctness of the two general strategies, we need to prove the following:

- The final resulting set covers are feasible: According to Lemma 1 and Lemma 3, the resulting set covers of both strategies are feasible.
- The final resulting set covers can α-cover the area: According to Lemma 2 and Lemma 4, the sensing ranges of sensors of final real set covers are *no smaller* than $\sqrt{\alpha}$ times of those of the corresponding virtual set covers. Since the virtual set covers can completely cover the area, according to Lemma 5, the final set covers can α-cover the area.

The resulting set covers of α-coverage algorithms of the both strategies are feasible and can α-cover the monitored area, which proves the correctness of both strategies. □

For the partial coverage problem, it is always desirable to evaluate how uniformly the network α-covers the area. A good metric for such an evaluation is *Sensing Void Distance* defined in [11] and [18] as follows:

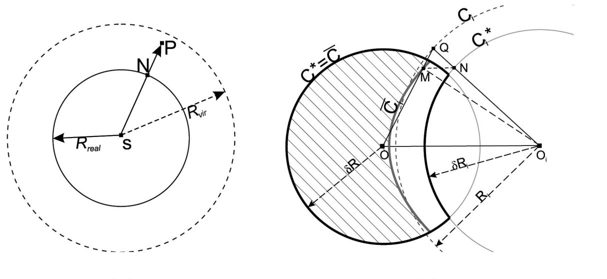
Definition 6 (SVD). *Sensing Void Distance is the maximum distance from a point that is not covered by any active sensor to the nearest point that is covered by an active sensor.*

Based on the definition of SVD, the upper bound of SVD of any α-coverage derived by the two general strategies is given in the following theorem:

Theorem 7. *Let R_{max} be the maximum sensing range a sensor may have. That is, for fixed-sensing-range WSNs, R_{max} is the largest sensing range of all the sensors and for adjustable-sensing-range WSNs, R_{max} is the largest one among all the possible maximum sensing ranges of all the sensors. For any original algorithm, the SVD of α-coverage derived by the two general strategies is bounded by the following values:*

- $$SVD \le \frac{1-\sqrt{\alpha}}{\sqrt{\alpha}} R_{max}$$

  for Strategy G-1.



(a) (b)

Fig. 1. (a) Sensing void distance. (b) Case 2.2.

- $$SVD \le \frac{1-\sqrt{\alpha}}{\sqrt{\beta}} R_{max}$$

  for Strategy G-2.

Proof. Assume that P is the uncovered point whose distance to the nearest covered point is SVD. Since P is covered by a virtual network (α-virtual network for Strategy G-1, β-virtual network, or fixed-β-virtual network for Strategy G-2), there exists a sensor s such that P is covered by $s(R_{vir})$, where R_{vir} is the virtual sensing range of s in the virtual network. In Fig. 1a, the inner solid circle is s's actual sensing region, i.e., s works with its real sensing range R_{real}. The outer dashed circle with radius of R_{vir} is s's sensing region in the virtual network. We have $R_{real} = \sqrt{\alpha} R_{vir}$.

Denote $|sP|$ as the euclidean distance from sensor s to point P, then $|sP| \le R_{vir}$. Assume that the line sP intersects with the inner circle at point N, clearly, $|sN| = R_{real}$. Because of the way we choose P, we have

$$\begin{aligned} SVD \le |NP| = |sP| - |sN| = |sP| - R_{real} \le R_{vir} - R_{real} \\ = R_{vir} - \sqrt{\alpha} R_{vir} = R_{vir}(1 - \sqrt{\alpha}). \end{aligned}$$

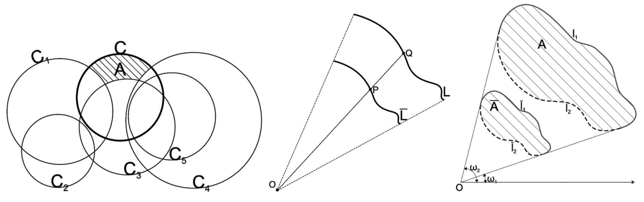
For Strategy G-1, the original algorithms work with α-virtual-network; thus, $R_{vir} \le \frac{R_{max}}{\sqrt{\alpha}}$. Therefore, $SVD \le \frac{1-\sqrt{\alpha}}{\sqrt{\alpha}} R_{max}$.

For Strategy G-2, the original algorithms work with β-virtual-network; thus, $R_{vir} \le \frac{R_{max}}{\sqrt{\beta}}$. Therefore,

$$SVD \le \frac{1-\sqrt{\alpha}}{\sqrt{\beta}} R_{max}.$$

□

For complete coverage ($\alpha = 1$), we have $\mathrm{SVD} = 0$. Clearly, our SVD bound is more meaningful than the one given in [18] which is $r_{tmax} + r_{smax} - r_{smin}$ (this bound is the same for any α) where r_{tmax}, r_{smax}, and r_{smin} are the maximum communication range, the maximum sensing range, and the minimum sensing range, respectively. For Strategy G-2, the larger the value of β, the smaller the value of SVD. Thus, users may use a large β to get a better coverage uniformity. However, if β is too large, the virtual network might not be able to completely cover the monitored area, since the sensors' sensing ranges in the virtual network are too small.



(a) (b) (c)

Fig. 2. *Private region* and (O, δ)-*compression*. (a) Private region of disk C. (b) (O, δ)-*compression* for a curve. (c) (O, δ)-*compression* for a region.

Theorem 8. *The time complexity of any resulting α-coverage algorithm derived by the two general strategies is the same as that of the original algorithm.*

Proof. The proposed general strategies essentially adjust the source code of an original algorithm in a way that the resulting algorithm can α-cover the area. Basically, the general strategies carry out two tasks:

1. Let the original algorithm work with a virtual network where every sensor s has its virtual sensing range. In this step, the framework (which also includes two extended strategies discussed in Section 5.3) only considers any reference to s's sensing range and merely changes s's real sensing range to a virtual sensing range. Thus, our framework does not change the time complexities of the original algorithms since it just conducts a simple calculation operation (a division) to the original algorithms.
2. Convert the virtual set covers to real set covers. Essentially, before returning the results, i.e., set covers with schedules, the two general strategies simply modify a piece of code in the original algorithm such that instead of returning the virtual sensing ranges, it returns real sensing ranges which are usually $\sqrt{\alpha}$ times of the virtual sensing ranges. Understanding the general strategies in this way, the time complexity of an original algorithm is not changed.

Thus, the time complexity of the resulting algorithm derived by the two proposed general strategies is the same as that of the original algorithm. □

5.3.4 δ-Compression Theorem

We dedicate this section to prove an important theorem showing the correctness of our framework. We first introduce some notations and definitions.

Notations:

$C(O, R)$ denotes a circle C with radius R centered at point O. The region enclosed by C is denoted by $D(C)$ (D stands for *disk*). We use $\|A\|$ to denote the area of region A. For a line XY, $|XY|$ is its length.

For 2 two-dimensional regions $\mathcal{A}$ and $\mathcal{B}$, we say $\mathcal{A} \subseteq \mathcal{B}$ if for every point $P \in \mathcal{A}$, we also have $P \in \mathcal{B}$. In other words, $\mathcal{A}$ is fully covered by $\mathcal{B}$. We say $\mathcal{A} \nsubseteq \mathcal{B}$ if there exists a point $P \in \mathcal{A}$, but $P \notin \mathcal{B}$. For example, as shown in Fig. 2a, we have $D(C_5) \subseteq D(C_4)$, $D(C_5) \nsubseteq D(C)$. $\mathcal{A} - \mathcal{B}$ includes any point P such that $P \in \mathcal{A}$, but $P \notin \mathcal{B}$.

Definitions:

Definition 7 (Neighbors). *In the plane, given a circle C and l other circles $C_1, C_2, \ldots, C_l$, a circle C_i ($i = 1 \ldots l$) is said to be a neighbor of C if C and C_i overlap.*

For example, in Fig. 2a, C_1, C_3, C_4, C_5 are all the neighbors of C, while C_2 is not.

Definition 8 (Private region). *The region of disk $D(C)$ that is not covered by any of its neighbors is called a private region.*

For example, the shaded region A in Fig. 2a is a private region of $D(C)$. $A = D(C) - \bigcup_{i=1}^{5} D(C_i)$.

Definition 9 ((O, δ)-compression). *Given a real number δ ($0 < \delta < 1$) and a point O, a curve $\mathcal{L}$, and a convex region $\mathcal{A}$ in the plane, we define:*

- *(O, δ)-compression of curve $\mathcal{L}$ is the curve $\overline{\mathcal{L}}$ consisting of every point P such that $\frac{|OP|}{|OQ|} = \delta$ for all a point $Q \in \mathcal{L}$.*
- *(O, δ)-compression of region $\mathcal{A}$ is the region $\overline{\mathcal{A}}$ consisting of every point P such that $\frac{|OP|}{|OQ|} = \delta$ for a point $Q \in \mathcal{A}$.*

As shown in Fig. 2b, $\overline{L}$ is (O, δ)-*compression* of L. In Fig. 2c, $\overline{A}$ is (O, δ)-*compression* of A. The following lemma can then be easily proved:

Lemma 9. *Given a real number δ ($0 < \delta < 1$), a point O and two convex regions $\overline{\mathcal{A}}$ and $\mathcal{A}$ in the plane, $\overline{\mathcal{A}}$ and $\mathcal{A}$ are enclosed by curves $\overline{\mathcal{L}}$ and $\mathcal{L}$, respectively. $\overline{\mathcal{A}}$ is (O, δ)-compression of $\mathcal{A}$ if and only if $\overline{\mathcal{L}}$ is (O, δ)-compression of $\mathcal{L}$.*

Lemma 10. *If $\overline{\mathcal{A}}$ is (O, δ)-compression of $\mathcal{A}$, then $\|\overline{\mathcal{A}}\| = \delta^2 \|\mathcal{A}\|$.*

Proof. We have two cases:

Case 1. O is outside of $\mathcal{A}$, i.e., $O \notin \mathcal{A}$. We partition $\mathcal{L}$ enclosing $\mathcal{A}$ into two curves l_1 and l_2, as shown in Fig. 2c. Similarly, we partition $\overline{\mathcal{L}}$ enclosing $\overline{\mathcal{A}}$ into two curves $\overline{l_1}$ and $\overline{l_2}$. Clearly, $\overline{l_1}$ and $\overline{l_2}$ are (O, δ)-compression of l_1 and l_2, respectively.

If we consider regions under the polar coordinate, then from [1], we have

$$\begin{aligned}\|\overline{\mathcal{A}}\| &= \frac{1}{2}\int_{\omega_1}^{\omega_2} \left(\overline{l_1}^2 - \overline{l_2}^2\right) d\omega = \frac{1}{2}\int_{\omega_1}^{\omega_2} ((\delta l_1)^2 - (\delta l_2)^2) d\omega \\ &= \frac{1}{2}\delta^2 \int_{\omega_1}^{\omega_2} \left(l_1^2 - l_2^2\right) d\omega = \delta^2 \|\mathcal{A}\|.\end{aligned}$$

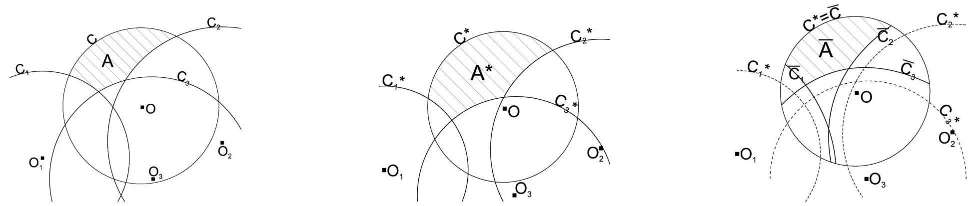
Case 2. O is inside of $\mathcal{A}$, i.e., $O \in \mathcal{A}$. Let l and $\bar{l}$ be the curves enclosing $\mathcal{A}$ and $\overline{\mathcal{A}}$, respectively. We have

$$\begin{aligned}l\|\overline{\mathcal{A}}\| &= \frac{1}{2}\int_{0}^{2\pi} (\bar{l}^2) d\omega = \frac{1}{2}\int_{0}^{2\pi} (\delta l)^2 d\omega = \frac{1}{2}\delta^2 \int_{0}^{2\pi} l^2 d\omega \\ &= \delta^2 \|\mathcal{A}\|.\end{aligned}$$

□

The δ-*compression* theorem:

Theorem 11 (δ-compression theorem). *In the plane, given n circles $C_i(O_i, R_i)$ ($i = 1 \ldots n$), the n corresponding disks $D(C_i)$ ($i = 1 \ldots n$) may overlap. If we "shrink" all the radius of the n circles by ratio δ ($0 < \delta < 1$), we will have n new circles $C_i^*(O_i, R_i^*)$, where $R_i^* = \delta R_i$ for $i = 1 \ldots n$. If we denote $\mathcal{A} = \bigcup_{i=1}^{n} D(C_i)$ and $\mathcal{A}^* = \bigcup_{i=1}^{n} D(C_i^*)$, then $\|\mathcal{A}^*\| \geq \delta^2 \|\mathcal{A}\|$.*



(a) (b) (c)

Fig. 3. *Private regions.* (a) The *private region* before shrinking. (b) The *private region* after shrinking. (c) Compressed *private region*.

Proof. We prove this theorem by induction on the number of the disks:

Basic step: $n = 1$: Trivial.

Inductive hypothesis: Assume that the theorem holds for $k = n - 1$ for some $n \geq 2$.

Inductive step: Prove for $k = n$. Among the n circles, assume that circle $C(O, R)$ is the one with the *smallest* radius. Denote other $n-1$ circles as $C_i(O_i, R_i)$ $(i = 1 \dots n-1)$. We have $R \leq R_i$ for $i = 1 \dots n-1$.

Let

$$\mathcal{A}_{n-1} = \bigcup_{i=1}^{n-1} D(C_i), \tag{2}$$

$$\mathcal{A} = \left[\bigcup_{i=1}^{n-1} D(C_i)\right] \bigcup D(C) = \mathcal{A}_{n-1} \bigcup D(C). \tag{3}$$

Intuitively, $\mathcal{A}$ is the union region of n disks $D(C)$ and $D(C_i)$ $(i = 1 \dots n-1)$, while $\mathcal{A}_{n-1}$ is the union region of $n-1$ disks $D(C_i)$ $(i = 1 \dots n-1)$ not including $D(C)$.

After n circles shrink with ratio δ, we have n new circles $C_i^*(O_i, R_i^*)$ $(i = 1 \dots n-1)$ and $C^*(O, R^*)$, where $R_i^* = \delta R_i$ and $R^* = \delta R$.

Let

$$\mathcal{A}^*_{n-1} = \bigcup_{i=1}^{n-1} D(C_i^*), \tag{4}$$

$$\mathcal{A}^* = \left[\bigcup_{i=1}^{n-1} D(C_i^*)\right] \bigcup D(C^*) = \mathcal{A}^*_{n-1} \bigcup D(C^*). \tag{5}$$

Intuitively, $\mathcal{A}^*$ is the union region of n disks, while $\mathcal{A}^*_{n-1}$ is the union region of $n-1$ disks not including $D(C^*)$.

With all those notations, the *inductive hypothesis* can be rewritten as $\|\mathcal{A}^*_{n-1}\| \geq \delta^2 \|\mathcal{A}_{n-1}\|$. We need to prove $\|\mathcal{A}^*\| \geq \delta^2 \|\mathcal{A}\|$. There are two cases:

Case 1. Before shrinking, if $D(C)$ is completely covered by all of its neighbors, i.e., $D(C) \subseteq \bigcup_{i=1}^{n-1} D(C_i) = \mathcal{A}_{n-1} \Rightarrow \mathcal{A} = \mathcal{A}_{n-1} \bigcup D(C) = \mathcal{A}_{n-1}$, then we are done because $\|\mathcal{A}^*\| \geq \|\mathcal{A}^*_{n-1}\| \geq \delta^2 \|\mathcal{A}_{n-1}\| = \delta^2 \|\mathcal{A}\|$.

Case 2. Otherwise, $D(C)$ is not fully covered by all of its neighbors.

Without loss of generality, assume that before shrinking C intersects with l other circles $C_i(O_i, R_i)$ $(i = 1 \dots l)$. For example, in Fig. 3a, we have $l = 3$. By assumption, we have $R \leq R_i$ for $i = 1 \dots l$.

Let A be the *private region* of $D(C)$. The region that can be covered by all the n disks is $\mathcal{A} = \mathcal{A}_{n-1} + A$. Hence, $\|\mathcal{A}\| = \|\mathcal{A}_{n-1}\| + \|A\|$.

After all the disks shrink as shown in Fig. 3b, let A^* be the new *private region* of $D(C^*)$. After shrinking, the region that can be covered by all the n disks is $\mathcal{A}^* = \mathcal{A}^*_{n-1} + A^*$. Hence, $\|\mathcal{A}^*\| = \|\mathcal{A}^*_{n-1}\| + \|A^*\|$.

We need to prove $\|\mathcal{A}^*\| \geq \delta^2 \|\mathcal{A}\| \Leftrightarrow \|\mathcal{A}^*_{n-1}\| + \|A^*\| \geq \delta^2(\|\mathcal{A}_{n-1}\| + \|A\|)$. From *inductive hypotheses*, we have $\|\mathcal{A}^*_{n-1}\| \geq \delta^2 \|\mathcal{A}_{n-1}\|$. So, we only need to prove $\|A^*\| \geq \delta^2 \|A\|$.

In Fig. 3c, we introduce a new concept which is *compressed private region*. We create (O, δ)-*compression* $\overline{C_i}$ of circles C_i (original circles before shrinking shown in Fig. 3a) for $i = 1 \dots l$, where O is the center of C and C^*. It is necessary to emphasize that $C^* \equiv \overline{C}$, i.e., C^* and $\overline{C}$ are the same circle. We only consider the border portions of $\overline{C_i}$ which are inside disk $D(\overline{C})$. From now on, we also use $\overline{C_i}$ to denote those portions. The new *private region*, denoted by $\overline{A}$, created by $D(\overline{C})$ and curves $\overline{C_i}$ is called a *compressed private region*.

Each curve $\overline{C_i}$ partitions disk $D(C^*)$ into two subregions (halves). We use $\overline{A_i}$ to denote the subregion that contains the *compressed private region* $\overline{A}$. That is, $\overline{A_i} = D(C^*) - D(\overline{C_i})$. We always have $\overline{A} \subseteq \overline{A_i} \subseteq D(C^*)$.

Similarly, we define $A_i^* = D(C^*) - D(C_i^*)$. If $D(C_i^*)$ overlaps $D(C^*)$, circle C_i^* also partitions disk $D(C^*)$ into two halves, A_i^* is the half that contains the *private subregion* A^*. Otherwise, if $D(C_i^*)$ does not overlap $D(C^*)$, then $A_i^* = D(C^*)$. Then, we have

$$\overline{A} = \bigcap_{i=1}^{l} \overline{A_i} \quad \textit{and} \quad A^* = \bigcap_{i=1}^{l} A_i^*. \tag{6}$$

It is easy to see that $\overline{A}$ is (O, δ)-*compression* of A. By Lemma 10, we have $\|\overline{A}\| = \delta^2 \|A\|$. Thus, instead of proving $\|A^*\| \geq \delta^2 \|A\|$, we are going to prove $\|A^*\| \geq \|\overline{A}\|$.

We claim that for $i = 1 \dots l$, $\overline{A_i} \subseteq A_i^*$. In other words, $\overline{A_i}$ is completely inside of A_i^*. This and (6) lead to the consequence that $\overline{A} \subseteq A^*$, i.e., $\overline{A}$ is also completely inside of A^*. Hence, we have $\|\overline{A}\| \leq \|A^*\|$. Thus, if the claim is proved, our theorem is consequently proved.

Now, we prove our claim. It is easy to see that if $D(C_i)$ overlaps $D(C)$, then $\overline{C_i} \neq \emptyset$. Thus, for all $i = 1 \dots l$, we have $\overline{C_i} \neq \emptyset$. Based on that fact and since C^* has the *smallest* radius among all the disks, for a particular i $(1 \leq i \leq l)$, there exist only two cases:

- *Case 2.1*: C_i^* does not intersect $D(C^*)$. Then, $\overline{A_i} \subseteq D(C^*) = A_i^*$. Hence, $\overline{A_i} \subseteq A_i^*$.
- *Case 2.2*: C_i^* does intersect $D(C^*)$. We will prove that $\overline{A_i} \subseteq A_i^*$, i.e., in Fig. 1b, the shaded region ($\overline{A_i}$)

is completely inside of the thickened region (A_i^*). To prove this, we only have to prove that $\overline{C_i}$ is completely inside of A_i^*.

Consider an arbitrary point $M \in \overline{C_i}$. We will prove that $M \in A_i^*$. Line OM intersects C_i at Q, and line O_iQ intersects C_i^* at N. We denote $\angle MNO_i$ as the angle formed by two rays NM and NO_i. We have $\frac{|OM|}{|OQ|} = \delta = \frac{|O_iN|}{|O_iQ|}$. Thus, line MN is parallel with line OO_i. Consequently, $\angle MNO_i + \angle NO_iO = \pi$.

Since $\overline{C_i}$ is inside of $D(C^*)$ and $M \in \overline{C_i}$; thus, $M \in D(C^*) \Rightarrow |OM| \leq \delta R$. Hence, $|OQ| = \frac{|OM|}{\delta} \leq R \leq R_i = |O_iQ|$. Consider triangle QOO_i. Since $OQ \leq O_iQ$ and $\angle QO_iO \leq \angle QOO_i$, we have $\angle QO_iO \leq \frac{\pi}{2}$. It means that $\angle MNO_i \geq \frac{\pi}{2}$. Since

$$\angle MNO_i + \angle NMO_i + \angle NO_iM = \pi,$$

we have $\angle NMO_i < \frac{\pi}{2} \leq \angle MNO_i$. Thus, $|O_iM| > |O_iN| = \delta R_i$.

$|O_iM| > \delta R_i$ means that point M is outside of disk $D(C_i^*)$, i.e., $M \notin D(C_i^*)$. Also, $M \in D(C^*)$, so $M \in [D(C^*) - D(C_i^*)] = A_i^*$. Since M is an arbitrary point in $\overline{C_i}$, from $\overline{C_i} \subseteq A_i^*$, we have $\overline{A_i} \subseteq A_i^*$.

For all the cases, we have $\overline{A_i} \subseteq A_i^*$, which proves our claim, and consequently, completes our proof. □

Section Extended Strategies, Advantages of Our Framework, and Simulation are shown in Supplementary File, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.124.

6 CONCLUSION

In this work, we propose a framework with strategies that can transform almost any existing complete coverage algorithm with any coverage ratio α to an algorithm that can α-cover the area to trade for network lifetime. Theoretical analysis and solid proof show the efficiency and the many advantages of our proposed framework. The simulation results further validate the efficiency of the four proposed strategies. As future work, we may conduct more simulations to characterize the pattern for "good" values of β.

ACKNOWLEDGMENTS

This work is supported by the US National Science Foundation (NSF) under grant No. CCF-0545667, the National Grand Fundamental Research 973 Program of China under grant No. 2006CB303000, and NSFC-RGC of China under grant No. 60831160525.

REFERENCES

[1] E.W. Weisstein, "Area, MathWorld—A Wolfram Web Resource," http://mathworld.wolfram.com/Area.html, 2010.

[2] C.T. Vu, S. Gao, W.P. Deshmukh, and Y. Li, "Distributed Energy-Efficient Scheduling Approach for k-Coverage in Wireless Sensor Networks," *Proc. 25th Military Comm. Conf. 2006 (MILCOM '06),* Oct. 2006.

[3] C.T. Vu, Z. Cai, and Y. Li, "Distributed Energy-Efficient Algorithms for Coverage Problem in Adjustable Sensing Ranges Wireless Sensor Networks," *Discrete Mathematics, Algorithms and Applications,* vol. 1, no. 3, pp. 299-317, 2009.

[4] C.T. Vu and Y. Li, "Delaunay-Triangulation Based Complete Coverage in Wireless Sensor Networks," *Proc. Int'l Workshop Information Quality and Quality of Service for Pervasive Computing (IQ2S '09) in Conjunction with PERCOM '09,* Mar. 2009.

[5] C.F. Huang, L.C. Lo, Y.C. Tseng, and W.T. Chen, "Decentralized Energy-Conserving and Coverage-Preserving Protocols for Wireless Sensor Networks," *Circuits and Systems,* vol. 1, pp. 640-643, May 2005.

[6] H. Zhang and J.C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Proc. 2004 NSF Int'l Workshop Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks,* 2004.

[7] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized Sensor Area Coverage with Low Communication Overhead," *IEEE Trans. Mobile Computing,* vol. 7, no. 5, pp. 661-672, May 2008.

[8] H. Zhang and J. Hou, "On Deriving the Upper Bound of α-Lifetime for Large Sensor Networks," *Proc. Fifth ACM Int'l Symp. Mobile Ad Hoc Networking and Computing,* 2004.

[9] H. Zhang and J. Hou, "Maximizing α-Lifetime for Wireless Sensor Networks," *Proc. Third Int'l Workshop Measurement, Modeling, and Performance Analysis of Wireless Sensor Networks (SenMetrics '05),* July 2005.

[10] S. Gao, X. Wang, and Y. Li, "p-Percent Coverage Schedule in Wireless Sensor Networks," *Proc. 17th Int'l Conf. Computer Comm. and Networks (ICCCN '08),* Aug. 2008.

[11] Y. Liu and W. Liang, "Approximate Coverage in Wireless Sensor Networks," *Proc. IEEE Conf. Local Computer Networks 30th Anniversary (LCN '05),* pp. 68-75, 2005.

[12] H. Bai, X. Chen, Y. Ho, and X. Guan, "Percentage Coverage Configuration in Wireless Sensor Networks," *Parallel and Distributed Processing and Applications,* vol. 3758/2005, pp. 780-791, 2005.

[13] B. Son, Y.-S. Her, and J.-G. Kim, "A Design and Implementation of Forest-Fires Surveillance System Based on Wireless Sensor Networks for South Korea Mountains," *IJCSNS Int'l J. Computer Science and Network Security,* vol. 6, no. 9B, pp. 124-130, Sept. 2006.

[14] M. Hefeeda, "Forest Fire Modeling and Early Detection Using Wireless Sensor Networks," Technical Report TR 2007-08, School of Computing Science, Simon Fraser Univ., Aug. 2007.

[15] L. Yu, N. Wang, and X. Meng, "Real-Time Forest Fire Detection with Wireless Sensor Networks," *Wireless Comm., Networking and Mobile Computing,* vol. 2, nos. 23-26, pp. 1214-1217, Sept. 2005.

[16] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Efficient Energy Management in Sensor Networks," *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing,* Y. Xiao and Y. Pan, eds., vol. 2, Nova Science Publishers, 2005.

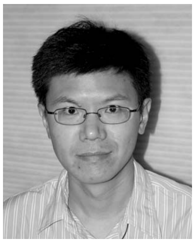
[17] N. Garg and J. Könemann, "Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems," *Proc. 39th Ann. Symp. Foundations of Computer Science,* pp. 300-309, 1998.

[18] Y. Wu, C. Ai, S. Gao, and Y. Li, "p-Percent Coverage in Wireless Sensor Networks," *Proc. Int'l Conf. Wireless Algorithms, Systems and Applications (WASA '08),* Oct. 2008.

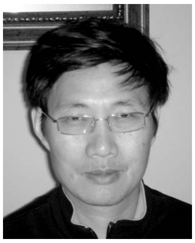
Yingshu Li received the BS degree in computer science from Beijing Institute of Technology, China, and the MS and PhD degrees in computer science from the University of Minnesota at Twin Cities. She is currently an assistant professor in the Department of Computer Science at Georgia State University. Her research interests include wireless networking, wireless sensor networks, and data management in sensor networks. She is the recipient of the US National Science Foundation (NSF) CAREER Award. She is a member of the ACM and a senior member of the IEEE.

Chinh Vu received the BS degrees in electronics and information technology from Hanoi University of Technology, Vietnam, and Hanoi Open University, Vietnam, in 2000, and the MS and PhD degrees from the Department of Computer Science, Georgia State University, in 2007 and 2009, respectively. His current research interests include wireless sensor networks, approximation algorithms, and distributed systems. He is a student member of the IEEE.

Chunyu Ai received the bachelor's and master's degrees in computer science from Heilongjiang University in 2001 and 2004, respectively. She is currently working toward the PhD degree at the Department of Computer Science, Georgia State University. Her research interests include wireless sensor networks, database, and data security. She is a member of the IEEE.



Guantao Chen received the BS degree in mathematics and the MS degree in operations search from the Central China Normal University in 1982 and 1984, respectively, and the PhD degree from the University of Memphis in 1991. He is currently a professor in the Department of Mathematics and Statistics with the joint appointment in the Department of Computer Science at Georgia State University. His research areas include graph theory, combinatorial algorithms, and bioinformatics. He is a member of the IEEE.



Yi Zhao received the PhD degree in mathematics from Rutgers University in 2001. He is an assistant professor of mathematics at Georgia State University. His research interests include graph theory, combinatorics, and their applications to computer science.