Three Approximation Algorithms for Energy-Efficient Query Dissemination in Sensor Database System*

Zhao Zhang[1], Xiaofeng Gao[2], Xuefei Zhang[2], Weili Wu[2], and Hui Xiong[3]

[1] College of Mathematics and System Sciences, Xinjiang University, P.R. China
zhzhao@xju.edu.cn
[2] Department of Computer Science, University of Texas at Dallas, Richardson, USA
{xxg052000,xxz068000,weiliwu}@utdallas.edu
[3] Management Science & Information Systems Department,
The State University of New Jersey, Rutgers, USA
hxiong@rutgers.edu

Abstract. Sensor database is a type of database management system which offers sensor data and stored data in its data model and query languages. In this system, when a user poses a query to this sensor database, the query will be disseminated across the database. During this process, each sensor generates data that match the query from its covered area and then returns the data to the original sensor. In order to achieve an energy-efficient implementation, it will be useful to select a minimally sufficient subset of sensors to keep active at any given time. Thus, how to find a subset efficiently is an important problem for sensor database system. We define this problem as *sensor database coverage* (SDC) problem.

In this paper, we reduce the SDC problem to *connected set cover* problem, then present two approximation algorithms to select a minimum connected set cover for a given sensor database. Moreover, to guarantee robustness and accuracy, we require a fault-tolerant sensor database, which means that each target in a query region will be covered by at least m sensors, and the selected sensors will form a k-connected subgraph. We name this problem as (k,m)-SDC problem and design another approximation algorithm. These three algorithms are the first approximation algorithms with guaranteed approximation ratios to SDC problem. We also provide simulations to evaluate the performance of our algorithms. We compare the results with algorithms in [17]. The comparison proves the efficiency of our approximations. Thus, our algorithms will become a new efficient approach to solve coverage problem in sensor database systems.

Keywords: Sensor Database, Set Cover, Fault Tolerance.

1 Introduction

1.1 Background

Sensors are often deployed widely to monitor continuously changing entities such as temperature, sound, vibration, pressure, locations of moving objects and other interests.

* This work is supported by National Natural Science Foundations of China (10671152), NSFC (60603003), the National Science Foundation under grant CCF-0514796 and CNS-0524429. This work was completed when Dr. Zhao Zhang visiting Department of Computer Science, The University of Texas at Dallas.

S.S. Bhowmick, J. Küng, and R. Wagner (Eds.): DEXA 2009, LNCS 5690, pp. 807–821, 2009.

The sensor readings are reported to a centralized database system, and are subsequently used to answer queries. Modern sensors not only respond to physical signals to produce data, but also embed computing and communication capabilities. They are able to store and process their productions locally, and transfer data through database system. Examples of monitoring applications include supervising items in a factory warehouse, gathering information in a disaster area, or organizing vehicle traffic in a large city [4]. These applications involve a combination of stored data, and we name them as *sensor databases* [2].

Sensor database system is a newly developed DBMS in recent years, which has been discussed in many literatures such as [2, 3, 21]. In a sensor database, users can issue database queries to one or more nodes in this database. Such process is called *sensor query*, which can also be defined as an acyclic graph of relational and sequence operators [2]. For instance, in a sensor database to measure temperature at regular interval, a typical sensor query can be shown like "*Return repeatedly the abnormal temperatures measured by all sensors*" or "*Every five minutes retrieve the maximum temperature measured over the last five minutes*" [2].

Sensor queries are long-running queries. During the span of a long-running query, relations and sensor sequences might be updated. The inputs of a relational operator are base sequences or the output of another sequence. We define R as a relation of a sensor database, and S as a sensor sequence. An update to R can be an insert, a delete, or modifications of record in R. An update to S is the insertion of a new record associated to a position greater than or equal to all undefined positions in S. There is a centralized realizations of a sensor database [6], where all data from each node in the sensor database is sent to a designated node within the database.

When a user (or an application) poses a query to the sensor database, the query is disseminated across the database. In response to this query, each node generates data that match this query, and transmits matching data to the original sensor. Each sensor can only generate data from its own covered area. As data routed through the database, intermediate sensors might apply one or more database operators. Then users can simply query this database. Such requirement means that users can get the result by querying at any sensor in the system. However, such process is impractical in the sensor database if every sensor can implement queries, since it requires significant communication and too many energy. Due to battery limitations, we need a minimally sufficient subset of sensors which can cover the whole query region at any given time. Since we need to transmit the query data outside the sensor network, such subset should also be connected. We define this problem as *sensor database coverage*(SDC) problem.

By Observation, SDC problem can be reduced to a *connected set cover* problem, which is proved to be *NP*-hard in general graph [16]. This problem can also be used in distributed Internet measurement systems for distributed agents to periodically measure the Internet by a tool called *traceroute* [5]. In this paper, we propose two approximation algorithms to select minimum connected set cover for a given sensor database. Moreover, to guarantee robustness and accuracy, we require a fault-tolerant sensor database, which means that each target in a query region will be covered by at least m sensors, and the selected sensors will form a k-connected subgraph. Under such constraints, we design another approximation algorithm for (k,m)-SDC problem. To make the algorithm

practical, we set $k = 2$ specifically. Both of these algorithms are the first approximation algorithms with guaranteed approximation ratios in general sensor database systems. We also provide simulations to evaluate the performance of our algorithms. We compare the results with algorithms in [17]. The comparison proves the efficiency of our approximations.

1.2 Related Works

The COUGAR project at Cornell University [2] is one of the first attempts to model a sensor database system. It focused on the interaction between the sequence data produced in sensor networks and stored data in backend relational databases. It extended both the SEQ [8] sequence data model and the relational data model by introducing new operators between sequence data and relational data. In [3], R. Cheng and S. Prabhakar presented a framework that represents uncertainty of sensor data. They proposed a new kind of probabilistic queries called *Probabilistic Threshold Query*. Also, they studied techniques for evaluating queries under different details of uncertainty, and investigated the tradeoff between data uncertainty, answer accuracy and computation costs. Recently, A lot of techniques have been introduced to solve coverage problems in sensor networks (e.g., [10, 11, 17, 18, 19, 20, 22]). One of the commonly used approach is reduce sensor coverage problem into connected dominating set (CDS) problem [26]. For k-coverage problem, literatures [9, 27] etc. proposed several greedy algorithms, but did not regard connectivity properties. We can use these techniques to solve SDC problem.

In our paper, we use sensor database as our communication model, which is seldom discussed because of the complexity of problem requirements. Actually, it is well known that minimum set cover (SC) problem is *NP*-hard [16], and can not be approximated within a factor of $(1-\varepsilon)\ln n$ for any $\varepsilon > 0$ unless $NP \subseteq DTIME(n^{\log\log n})$ [15], where $n = |V|$. Since SC is a special case of connect set cover (CSC) (taking G to be a complete graph), CSC is also NP-hard and is not $(1-\varepsilon)\ln n$-approximable. Furthermore, Shuai et.al. [23] showed that even when at most one vertex of the graph G has degree greater than two, the CSC problem is still non-$(1-\varepsilon)\ln n$-approximable. In the case that the graph is a path, Shuai et.al. gave two polynomial-time algorithms. In the case that the graph has exactly one vertex of degree greater than two, they proposed a $(1+\ln n)$-approximation algorithm. For the general case, there is no known approximation algorithm with guaranteed performance ratio.

1.3 Our Contribution

In this paper, we provide three efficient approximation algorithms to solve the SDC problem and (k,m)-SDC problem for efficient query dissemination in sensor database systems. Those approximation algorithms are the first ones with approximation ratio analysis. We also provide simulations to evaluate the performance of our algorithms. We compare the results with algorithms in [17]. The comparison proves the efficiency of our approximations. The detailed technologies can be summarized as follows.

We first define a new generalization of the connected set cover (CSC) problem that is equivalent to the SDC problem, and give two approximation algorithms. Assume we have a set collection $\mathbf{S} = \{S_1, S_2, \cdots, S_k\}$. The goal of these two algorithms are finding

a minimum size sub-collection $\mathbf{R} \subseteq \mathbf{S}$, such that all the target region is covered by $\mathbf{R}$, and $\mathbf{R}$ is connected. The approximation ratio is highly depends on a parameter $D_c(G)$, which can be defined as follows.

For any two sets $S_i, S_j \in \mathbf{S}$, $dist_G(S_i, S_j)$ is the length of a minimum (S_i, S_j)-path in G, where length refers to the number of edges on this path. Two sets $S_i, S_j \in \mathbf{S}$ are said to be *cover-adjacent* if $S_i \cap S_j \neq \emptyset$. Define $D_c(G) = \max\{dist_G(S_i, S_j) \mid S_i, S_j \in \mathbf{S}$ and S_i, S_j are cover-adjacent$\}$.

The first algorithm is a two-step algorithm. It finds an SC using an α-approximation algorithm, and then connects them with a Steiner Minimum Tree with Minimum Number of Steiner Points (SMT-MSP) using a β-approximation algorithm. The performance ratio of this algorithm is $\alpha + \beta + \alpha\beta(D_c(G) - 1)$. The second algorithm uses a greedy strategy, and the performance ratio is $1 + D_c(G) \cdot H(\gamma - 1)$, where H is the harmonic function, and $\gamma = \max\{|S| \mid S \in \mathbf{S}\}$. In many cases, $D_c = 1$. For example, if two reserves containing a same species are regarded to be adjacent, then $D_c = 1$. In such cases, the two algorithms given in this paper has performance ratio $\alpha + \beta$ and $1 + H(\gamma - 1)$ respectively.

Then, we consider the (k, m)-SDC problem. For a SDC $\mathbf{R}$, if the subgraph of G induced by $\mathbf{R}$ is k-connected, and every element of V is covered by at least m sets of $\mathbf{R}$, then $\mathbf{R}$ is a *(k, m)-connected set cover* ((k, m)-CSC for short). It is obvious that (k, m)-CSC problem is equivalent to (k, m)-SDC problem. If a reserve system takes the form of a (k, m)-SDC, then every species is represented at at least m reserves, and the connection among the reserves is more fault tolerant in face of disasters.

Specifically, in this paper, we present a greedy algorithm for the minimum $(2, m)$-SDC problem, using a parameter $PD(G)$. Given three vertices u, v, w in a graph G, define the *pair distance between u and* $\{v, w\}$, denoted by $dist(u; v, w)$, to be the shortest length of a pair of disjoint (u, v)-path and (u, w)-path. In another word, it is the length of a shortest (v, w)-path through vertex u. The *pair diameter* of a graph G is $PD(G)$= min{dist(u;v,w), where u, v, w are three distinct vertices in $V(G)$}. Our algorithm has performance ratio $(PD(G) - 1)(1 + H(\gamma - 1))$.

Then, we compared our algorithms to algorithms in [17] in several scenarios. We change the number of sensors in database and the radius of the sensors to exhibit the performance of our algorithms. The result showed that our algorithms are much better than these naive algorithms. The sizes of solutions we obtained are much closer to the corresponding optimum solutions.

The rest of this paper is organized as follows: Section 2 illuminates some basic concepts which may used in algorithm description and performance analysis. Section 3 presents the idea and detailed steps of our approximations for SDC problem. Section 4 provides a greedy algorithm to solve (k, m)-SDC problem. Proofs and performance analysis are also included in these two sections. Section 5 compares our performance with various previous works. Finally, Section 6 gives a brief conclusion of our work.

2 Preliminaries

We consider our communication model under general graphs, which can reflect any type of sensor database in practice, bringing benefits and efficiency to real-life applications.

Actually, we do not need to consider specific geometrical characteristics, since they are too strict to what dimension the models are built, and based on Euclidean formula (e.g., some of the rules are suitable in 2-dimensional space, but incorrect in 3-dimensional space). Therefore, our algorithm can be implemented in a wide range of environments. The following are basic definitions that we need to use in our algorithm descriptions.

Definition 1 (Query Region). *Query Region is the area of the entire sensor database that the end user (or an application) wants to issue a query.*

Definition 2 (Sensor Covering a Point). *A sensor in a sensor database system S is said to cover a point p, if the distance $d(p,S)$ between p and S is less than R_S, which is the sensing radius of the sensor (Here we assume that each sensor has the same R_S).*

Definition 3 (Sensor Database Coverage (SDC)). *Given a sensor database system with sensor set* **S**, *where* $\mathbf{S} = \{S_1, ..., S_k\}$. *We need to find a minimum subset* **R** *of* **S** *to cover all the query region, such that the subgraph induced by* **R** *is connected.*

Definition 4 (Set Cover (SC)). *Let V be a set of elements, and* **S** *be a family of subsets of V such that $\bigcup_{S \in \mathbf{S}} S = V$. A* set cover (SC) with respect to $(V, \mathbf{S})$ *is a sub-family* **R** *of* **S** *such that every element $v \in V$ is in some set $S \in \mathbf{R}$. We say that S covers v.*

Definition 5 (Connected Set Cover (CSC)). *Let G be a connected graph on vertex set* **S**. *A* connected set cover with respect to $(V, \mathbf{S}, G)$ *(abbreviated as CSC) is a set cover* **R** *with respect to* $(V, \mathbf{S})$ *such that the subgraph of G induced by* **R** *is connected.*

Definition 6 ((k,m)-CSC). *A (k,m)-CSC is a set cover* **R** *with respect to* $(V, \mathbf{S})$ *such that the subgraph of G induced by* **R** *is k-connected, and every element of V is covered by at least m sets of* **R**, *then* **R** *is a* (k,m)-connected set cover *((k,m)-CSC for short).*

Note that we use terminology 'set' and 'vertex' interchangeably when talking about elements in **S**. Because the sensor database coverage problem is indeed a generalization of the connected set cover problem.

Now let us discuss how to reduce SDC problem to CSC problem (so that (k,m)-SDC is equivalent to (k,m)-CSC). SDC problem is considering cover a whole region, while CSC problem is considering cover a set of targets. Thus, we need to reduce region coverage into target coverage. Figure 1 is an example to illustrate this reduction.

The square in Fig. 1 is the potential region for sensoring. For each point in this area, let A denote the set of sensors that can cover this point. Partition the area into different parts, each with different sensor coverage set. As a consequence, area coverage problem can be reduced to target coverage problem when we consider each part as a target. This problem can further reduced to a set cover problem. Consider each small division as a target and mark it with a number from 1 to 14, and then insert covered targets into each set of sensors. Say, $S_1 = \{7,9,11,12,13,14\}$, $S_2 = \{6,9,13\}$, $S_3 = \{3,6,7,8,10,13,14\}$, $S_4 = \{3,4,5\}$, $S_5 = \{2,1,3,4,8,12,14\}$ and $S_6 = \{2\}$. The coverage problem can be reduced to the problem of finding a minimum set cover from S_i to cover $T = \{1, \cdots, 14\}$.

Next, we will introduce our approximation algorithm for SDC problem in the next section now.

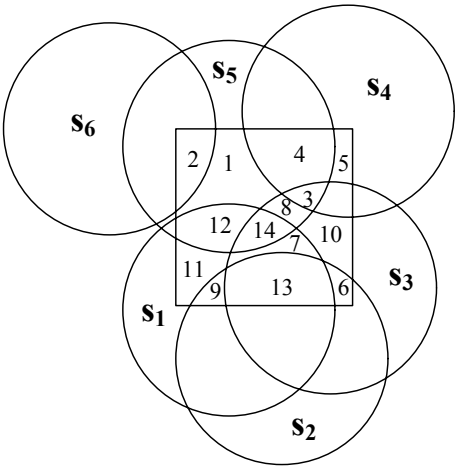

Fig. 1. An example to reduce region coverage to target coverage

3 Algorithm for SDC Problem

In this section, we will firstly exhibit a two-step SDC algorithm, and then give a modified greedy algorithm to deal with more general cases. In the next section, we will show algorithm to solve a special (k,m)-SDC problem where $k = 2$. Performance analysis and theory proofs are provided immediately after algorithm descriptions.

3.1 Two-Step SDC Algorithm

Firstly, let us depict the two-step algorithm as follows.

Algorithm 1. (Two-Step SDC)

Input: $(V,\mathbf{S},G)$; an algorithm **A** computing a minimum set cover; an algorithm **B** computing a Steiner tree with minimum number of Steiner points.
Output: A connected set cover $\mathbf{R}$.

1: Use **A** to compute a set cover $\mathbf{R}_1$ with respect to $(V,\mathbf{S})$.
2: Use **B** to compute a Steiner tree T in G with terminal set $\mathbf{R}_1$. and Steiner points $\mathbf{R}_2$.
3: Output $\mathbf{R} = \mathbf{R}_1 \cup \mathbf{R}_2$.

Theorem 1. *Suppose the approximation ratio of* **A** *and* **B** *are* α *and* β *respectively. Then the approximation ratio for Two-Step SDC is* $\alpha + \beta + \alpha\beta(D_c - 1)$.

Proof. Let $\mathbf{R}^*$ be an optimal solution to SDC, and $\mathbf{R}_2^*$ be a Steiner tree of G connecting terminal set $\mathbf{R}_1$ with minimum number of Steiner points. Since $\mathbf{R}^*$ is also a set cover with respect to $(V,\mathbf{S})$, we have

$$|\mathbf{R}_1| \leq \alpha|\mathbf{R}^*|. \tag{1}$$

Let S be a set in $\mathbf{R}_1$. Suppose v is an element of V covered by S, and S^* is a set in $\mathbf{R}^*$ covering v. Then S, S^* are cover-adjacent, and thus $dist_G(S,S^*) \leq D_c$. By adding at most

$D_c - 1$ vertices of G connects S to S^*. It follows that by adding at most $(D_c - 1)|\mathbf{R}_1|$ vertices, all sets in $\mathbf{R}_1$ are connected to $\mathbf{R}^*$. Since $G[\mathbf{R}^*]$ is connected, we have

$$|\mathbf{R}_2^*| \leq |\mathbf{R}^*| + (D_c - 1)|\mathbf{R}_1|. \tag{2}$$

Combining inequalities (1) and (2) with $|\mathbf{R}_2| \leq \beta|\mathbf{R}_2^*|$, the approximation ratio follows.

In the Relay Node Problem, if $R \geq 2r$, then $D_c = 1$ and thus the approximation ratio is $\alpha + \beta$.

3.2 Best Candidate Path Algorithm (BCP) for SDC Problem

Now let us introduce the best candidate path algorithm (BCP) for SDC problem with better performance.

Definition 7 (Candidate Path). *Let* $\mathbf{R}$ *records the sets which have been chosen and* U *records the set of elements of* V *which have been covered. For* $\mathbf{R} \neq \emptyset$ *and a set* $S \in \mathbf{S} \setminus \mathbf{R}$*, an* $\mathbf{R}$*-*S *candidate path is a path in* G *such that its initial vertex is in* $\mathbf{R}$ *and its end vertex is* S*.*

For a shortest $\mathbf{R}$-S candidate path P_S, it has exactly $|P_S|$ vertices in $\mathbf{S} \setminus \mathbf{R}$, where $|P_S|$ is the number of edges in P_S. We use $C(P_S)$ to denote the set of elements of $V \setminus U$ which are covered by vertices on P_S. Define $e(P_S) = \frac{|P_S|}{|C(P_S)|}$. Then we have Algorithm 2.

Algorithm 2. (BCP Algorithm)

Input: $(V, \mathbf{S}, G)$.
Output: A connected set cover $\mathbf{R}$.

1: Choose $S_1 \in \mathbf{S}$ such that $|S_1|$ is maximum. $\mathbf{R} = \{S_1\}$, $U = S_1$.
2: **while** $V \setminus U \neq \emptyset$ **do**
3: For each $S \in \mathbf{S} \setminus \mathbf{R}$ which is cover-adjacent with a set in $\mathbf{R}$, compute a shortest $\mathbf{R}$-S path P_S. Choose S such that $e(P_S)$ is minimum. Add all sets on P_S except S into $\mathbf{R}$, $U = U \cup C(P_S)$.
4: **end while**
5: Output $\mathbf{R}$.

Clearly, the output $\mathbf{R}$ of Algorithm 2 is a connected set cover for $(V, \mathbf{S}, G)$. Next, we analyze the approximation ratio.

Theorem 2. *The BCP Algorithm has approximation ratio* $1 + D_c(G) \cdot H(\gamma - 1)$*, where* $\gamma = \max\{|S| \mid S \in \mathbf{S}\}$*, and* H *is the harmonic function.*

Proof. Suppose S_i is the set chosen in the i^{th} iteration (S_1 is the initial set chosen in line 1). Let $\mathbf{S}_i$ be the set of sets added to $\mathbf{R}$ in the i^{th} iteration (that is, the sets on P_{S_i} which is not already in $\mathbf{R}$). Then $\mathbf{R}_k = \bigcup_{i=1}^{k} \mathbf{S}_i$ is the set of sets chosen after the k^{th} iteration. Suppose Algorithm 2 runs K rounds. Then $\mathbf{R}_K$ is the output of the algorithm. When S_i is chosen, we assign each element $v \in C(P_{S_i})$ a weight $w(v) = e(P_{S_i})$ for $i \geq 2$ and $w(v) = 1/|S_1|$ for $i = 1$. Then each element $v \in V$ is assigned a weight exactly once, and

$$\sum_{v \in V} w(v) = \sum_{k=1}^{K} \sum_{v \in C(P_{S_k})} w(v) = \sum_{k=1}^{K} \sum_{v \in C(P_{S_k})} \frac{|P_{S_k}|}{|C(P_{S_k})|} = \sum_{k=1}^{K} |P_{S_k}| = |\mathbf{R}_K|. \tag{3}$$

Suppose $\mathbf{R}^* = \{S_1^*, ..., S_{opt}^*\}$ is an optimal solution to the SDC problem. Set $N_1 = S_1^*$, and for $i = 2, ..., k$, set $N_i = S_i^* \setminus (\bigcup_{j=1}^{i-1} N_j)$. Since $\mathbf{R}^*$ covers all elements of V, we see that $N_1, ..., N_{opt}$ is a partition of V. It follows that

$$\sum_{v \in V} w(v) = \sum_{k=1}^{opt} \sum_{v \in N_k} w(v). \tag{4}$$

Next, we show that for each $k \in \{1, ..., opt\}$,

$$\sum_{v \in N_k} w(v) \leq 1 + D_c(G) \cdot H(\gamma - 1). \tag{5}$$

Let $n_0 = |N_k|$, and for $i = 1, ..., k$ let n_i be the number of elements in N_k which are not covered after the i^{th} iteration. For $i = 1, ..., k$, after the i^{th} iteration, $n_{i-1} - n_i$ elements of N_k are covered and each such an element is assigned a weight

$$e(P_{S_i}) \leq e(P_{S_k^*}) = \frac{|P_{S_k^*}|}{|C(P_{S_k^*})|} \leq \frac{D_c(G)}{n_{i-1}} \text{ for } i \geq 2, \tag{6}$$

and at most $1/(n_0 - n_1)$ for $i = 1$. There are something to be explained about (6).

(a) It is possible that $n_{i-1} - n_i > 0$. But only those i's with $n_{i-1} - n_i > 0$ works in the analysis.

(b) As a consequence of the above assumption, S_k^* is not chosen after the $(i-1)^{th}$ iteration since choosing S_k^* covers all the elements in N_k. Furthermore, $n_0 - n_1 > 0$ implies that

$$S_k^* \text{ is cover-adjacent with } S_1. \tag{7}$$

Hence S_k^* is a candidate to be chosen as S in the i^{th} iteration for $i \geq 2$. By the choice of S_i, the first inequality of (6) holds.

(c) Also by observation (7), we have $|P_{S_k^*}| \leq D_c(G)$. Since choosing S_k^* could cover all the remaining elements in N_k, we have $|C(P_{S_k^*})| \geq n_{i-1}$. The second inequality in (6) holds.

Then by a standard analysis as in dealing with set cover problem (see for example [14] §35.3), we have

$$\begin{aligned} \sum_{v \in N_k} w(v) &\leq (n_0 - n_1) \frac{1}{n_0 - n_1} + D_c(G) \sum_{i=2}^{opt} \frac{n_{i-1} - n_i}{n_{i-1}} \\ &\leq 1 + D_c(G)(H(n_1) - H(n_{opt})). \end{aligned}$$

Inequality (5) follows from the observation that $n_{opt} = 0$ and $n_1 < n_0 = |N_k| \leq |S_k^*| \leq \gamma$. Combining inequalities (3) (4) and (5), we have

$$|\mathbf{R}| = \sum_{k=1}^{opt} \sum_{v \in N_k} w(v) \leq (1 + D_c(G) H(\gamma - 1)) \cdot opt.$$

The theorem is proved.

4 Best Efficiency Ear Algorithm (BEE) for $(2,m)$-SDC Problem

In this section we provide another algorithm, best candidate ear algorithm (BEE) for (k,m)-SDC problem with fixed parameter $k=2$. To compute a $(2,m)$-SDC, we make use of the *ear decomposition* of 2-connected graphs.

Definition 8 (Ear). *An* ear *of a graph G is a path P in G such that all internal vertices on P has degree two in G.*

An ear is *open* if its two ends are different, otherwise it is *closed*. A cycle is a closed ear. The ear decomposition theorem says that every 2-connected graph has an open ear P such that the graph obtained by deleting internal vertices of P from G is still 2-connected. In another word, a graph G is 2-connected if and only if G can be constructed in the following way: Starting from a cycle (that is a closed ear); Iteratively adding open ears to the graph.

The BEE Algorithm computes a $(2,m)$-SDC by greedy strategy. It starts from a 'most efficient' cycle, then iteratively adds 'most efficient' open ears to it until all the cover requirements are satisfied.

To compute the open ears, we use the concept of shortest (u,v)-cycle.

Definition 9 (Shortest (u,v)-cycle). *For two distinct vertices u,v in a graph G, a* shortest (u,v)-cycle *is a cycle in G through u and v such that the length of the cycle (that is, the number of edges in the cycle) is minimum.*

A shortest (u,v)-cycle can be computed by any algorithm finding *shortest pair of disjoint paths*. In fact, the union of a pair of disjoint (u,v)-paths is an (u,v)-cycle. There are many algorithms for shortest pair of disjoint paths problem, for example, [24].

For a subgraph H of G, a shortest open ear to H through a given vertex $v \in V(G) \setminus V(H)$ can be computed as follows: Add a new vertex s to G and connect s to every vertex in H; Compute a shortest (v,s)-cycle in the extended graph; Then the path obtained by deleting s from this cycle is as required.

4.1 Algorithm Description

Now let us give the detailed description of $(2,m)$-SDC algorithm in Algorithm 3.

In this algorithm, each element $v \in V$ is assigned a label $m(v)$ which records the remaining number of times element v is to be covered. Initially $m(v)=m$ for all v. When $m(v)$ decreases to zero, we say that the cover requirement on v is satisfied. The total number of remaining cover requirements is recorded by M. Initially $M=m|V|$. Set U is used to record the elements of V whose cover requirements has not been satisfied.

For an ear Q_S computed in the algorithm, we use $c(Q_S)$ to denote the number of cover requirements satisfied by adding Q_S to the currently constructed 2-connected subgraph. To speak it more concretely, for each element $v \in U$, let $m'(v)$ be the number of sets in $V(Q_S) \setminus \mathbf{R}$ which cover v, and set $\widetilde{m}(v) = \min\{m'(v), m(v)\}$. Then $\widetilde{m}(v)$ is the number of requirements newly satisfied at element v by adding Q_S, and $c(Q_S) = \sum_{v \in U} \widetilde{m}(v)$ is the total number of requirements newly satisfied by adding Q_S. Define the *efficiency* of Q_S to be

$$e(Q_S) = \frac{|V(Q_S) \setminus \mathbf{R}|}{c(Q_S)}.$$

Algorithm 3. (BEE Algorithm)

Input: $(V, \mathbf{S}, G)$, where G is 2-connected and every element in V is covered by at least m sets in $\mathbf{S}$.

Output: A $(2,m)$-connected set cover $\mathbf{R}$.

1: Set $M = m|V|$, $U = V$, and $m(v) = m$ for each $v \in V$.
2: Choose $S_1 \in \mathbf{S}$ such that $|S_1|$ is maximum. $\mathbf{R} = \{S_1\}$. For each element $v \in S_1$, set $m(v) = m(v) - 1$. $M = M - |S_1|$. Remove all vertices v in U with $m(v) = 0$.
3: **if** $M = 0$ **then**
4: Output $\mathbf{R}$.
5: **else**
6: For each $S \in \mathbf{S} \setminus \mathbf{R}$, compute a shortest (S_1, S)-cycle Q_S.
7: Choose S such that $e(Q_S)$ is minimum.
8: **for** each set $R \in V(Q_S) \setminus \mathbf{R}$ **do**
9: $\mathbf{R} = \mathbf{R} \cup \{R\}$.
10: For each element $v \in R \cap U$, $m(v) = m(v) - 1$, $M = M - 1$, and remove v from U if $m(v) = 0$.
11: **end for**
12: **end if**
13: **while** $M > 0$ **do**
14: Construct a graph $\widetilde{G}$ by adding a new vertex S_0 and connect S_0 to every vertex in $\mathbf{R}$.
15: For each $S \in \mathbf{S} \setminus \mathbf{R}$, compute a shortest (S_0, S)-cycle in $\widetilde{G}$. Let Q_S be the open ear to $G[\mathbf{R}]$ obtained by deleting S_0 from this cycle.
16: Choose S such that $e(Q_S)$ is minimum.
17: **for** each set $R \in V(Q_S) \setminus \mathbf{R}$ **do**
18: $\mathbf{R} = \mathbf{R} \cup \{R\}$.
19: For each element $v \in R \cap U$, $m(v) = m(v) - 1$, $M = M - 1$, and remove v from U if $m(v) = 0$.
20: **end for**
21: **end while**
22: Output $\mathbf{R}$.

Line 6 to 11 is constructing the initial cycle and line 14-20 is iteratively adding open ears. By the ear decomposition theorem, the output of Algorithm 3 is indeed a $(2,m)$-SDC.

4.2 Performance Analysis

To analyze the performance ratio of the BEE Algorithm, we define the concept of pair diameter. Given three vertices u, v, w in a graph G, define the *pair distance between u and* $\{v,w\}$, denoted by $dist(u;v,w)$, to be the shortest length of a pair of disjoint (u,v)-path and (u,w)-path. In another word, it is the length of a shortest (v,w)-path through vertex u. The *pair diameter* of a graph G is $PD(G) = \min\{dist(u;v,w) \mid u, v, w$ are three distinct vertices in $V(G)\}$.

Theorem 3. *The performance ratio of BEE Algorithm is* $PD(G)(1 + H(\gamma - 1))$*, where* $\gamma = \max\{|S| \mid S \in \mathbf{S}\}$.

Proof. The proof idea is similar to that of Theorem 2. The difference lies in dealing with the multiple covering of each element and estimating the length of added ear.

Suppose $V = \{v_1, ..., v_n\}$ where $n = |V|$. Duplicate each element v_i by m times. Denote by $V_i = \{v_i^{(1)}, ..., v_i^{(m)}\}$, where $v_i^{(1)}, ..., v_i^{(m)}$ are the duplicates of element v_i. Set $\mathbf{V} = \bigcup_{i=1}^{n} V_i$.

Use the notation S_i, $\mathbf{R}_k$ as in the proof of Theorem 2. Suppose Algorithm 3 runs K rounds. For $i \geq 2$, when S_i is chosen, sets in $V(Q_{S_i}) \setminus \mathbf{R}$ are added into $\mathbf{R}$ sequentially in line 8 to line 11. When it is the turn to deal with $R \in V(Q_{S_i}) \setminus \mathbf{R}$, a vertex $v \in R \cap U$ has its copy $v^{(m(v))}$ assigned a weight $e(Q_{S_i})$ (recall that $1 \leq m(v) \leq m$ is the remaining cover requirements on v just before R is added to $\mathbf{R}$). We may regard R to cover $v^{(m(v))}$. When $i = 1$, each element $v \in S_1$ has its copy $v^{(m)}$ assigned a weight $1/|S_1|$. Then each element $v^{(j)} \in \mathbf{V}$ is assigned a weight exactly once.

Suppose $\mathbf{R}^* = \{S_1^*, ..., S_{opt}^*\}$ is an optimal solution to the $(2,m)$-SDC problem. Define a partition $N_1, ..., N_{opt}$ of $\mathbf{V}$ as follows (write $\mathbf{N}_i = \bigcup_{k=1}^{i} N_k$ for simplicity): Set $N_1 = \{v^{(1)} \mid v \in S_1^*\}$, and for $i = 2, ..., opt$, set $N_i = \{v^{(j)} \mid v \in S_i^*,\ v^{(m)} \notin \mathbf{N}_{i-1},\ j$ is the first index such that $v^{(1)}, ..., v^{(j-1)} \in \mathbf{N}_{i-1}$ and $v^{(j)} \notin \mathbf{N}_{i-1}\}$. Figure 2 illustrates the partition.

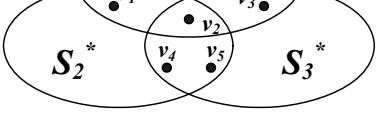

Fig. 2. An illustration of the partition. Here we have that $m = 2$, $N_1 = \{v_1^{(1)}, v_2^{(1)}, v_3^{(1)}\}$, $N_2 = \{v_1^{(2)}, v_2^{(2)}, v_4^{(1)}, v_5^{(1)}\}$, $N_3 = \{v_3^{(2)}, v_4^{(2)}, v_5^{(2)}\}$.

The following proof is similar to that in Theorem 2. The only difference is using $PD(G)$ to upper bound $|V(Q_{S_k^*}) \setminus \mathbf{R}_i|$. Note that for each $k \in \{1, ..., opt\}$, $|N_k| \leq \gamma$ since each element v has at most one copy in N_k.

5 Performance Evaluation

In this section, we present tow simulations to evaluate the performance of our approximation algorithms. Since the performance of BCP is better than two-step SDC, we will build one simulation for BCP algorithm, and another with (k,m)-SDC.

We ran our algorithms on a randomly generated sensor database system where a certain number of sensor nodes are placed randomly in an area of 40×40 unit square. We assume that the query region is the entire sensor database region. Each sensor has a uniform sensing radius of 4 units, which means that it can only detect targets in a cycle of 4 units with itself as the center. We vary the size n of this sensor database from 1000 to 4000 (which provides substantial redundancy), and deploy these sensors randomly. For each fixed topology, we calculate the required subset from our algorithm. Also,

we determine the sensing radius R of sensor nodes from 2 units to 12 units. In both scenarios, we compare our outputs with the results calculated from K Times 1-Greedy algorithm, which is mentioned in [17].

5.1 Result for BCP Algorithm

We plot the size of the solution by different algorithms in Figure 3 for different values of sensor database sizes, and transmission radius. For each parameter, we run each scenario for 1000 times and take average value as our solution to avoid abnormal cases.

Figure 3 (a) shows the solution size delivered by two algorithms with different sensor database sizes. Note that the solution size is much smaller than the sensor database size (10^2 vs. 10^3), which means that selecting a subset of sensors to execute sensor queries for a sensor database do save many energy. Therefore, selecting a connected subset is an energy-efficient method for a sensor database system to improve its performance.

From this figure we can also observe that when the size of database increases, the solution size decreases. This is because when the target region fixed, if the density of sensors increases, it is easier to find a connected subset to cover the whole area. Moreover, the solution size doesn't decrease so much when the sensor database size becomes larger, which means the solution obtained from 1000 sensors is quite close to the OPT. It also means that the sensors database provides substantial redundancy if the size increases. We can see that for a fixed sensing radius, the solution size returned by BCP algorithm is almost the half of the solution returned by K times 1-greedy algorithm, which proves that our algorithm can perform better results.

Figure 3 (b) shows the solution size delivered by various algorithms for different sensing radius. Note that when the sensing radius change from 2 to 4, the solution size has a sharp decrease, but if the sensing radius increase continuously, the solution size stays stable. That means when the sensing radius is bigger than 4, the query region is covered more than once. This property can guarantee robustness and accuracy. Also note that the solution size is quit small and good enough to reach the optimum solution when the sensing radius becomes larger.

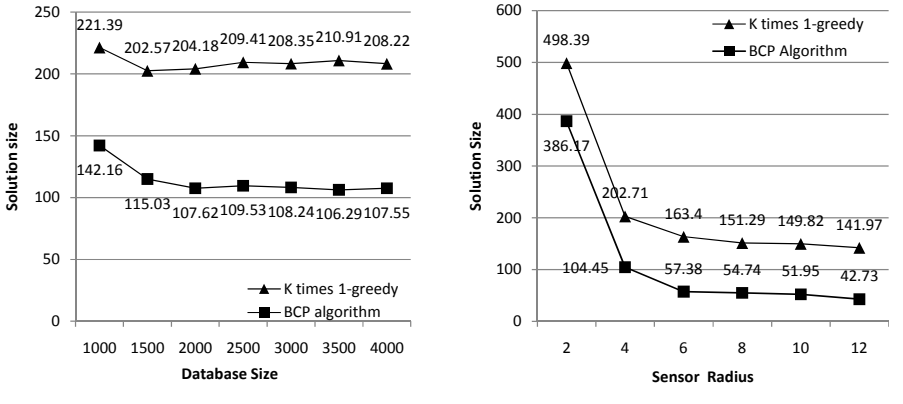

Fig. 3. Solution size of connected sensor cover delivered by various algorithms with different sensor database size and transmission radius

5.2 Result for BEE Algorithm

We plot the size of the solution from different algorithms in Figure 4 for different values of coverage degree. The coverage degree *m* denotes that each target should be covered by at least *m* sensors. Similarly as previous testing, under every parameter we run each algorithms for 1000 times, and choose the average value as our final result, so that we will avoid extreme cases (since the topology is randomly generated).

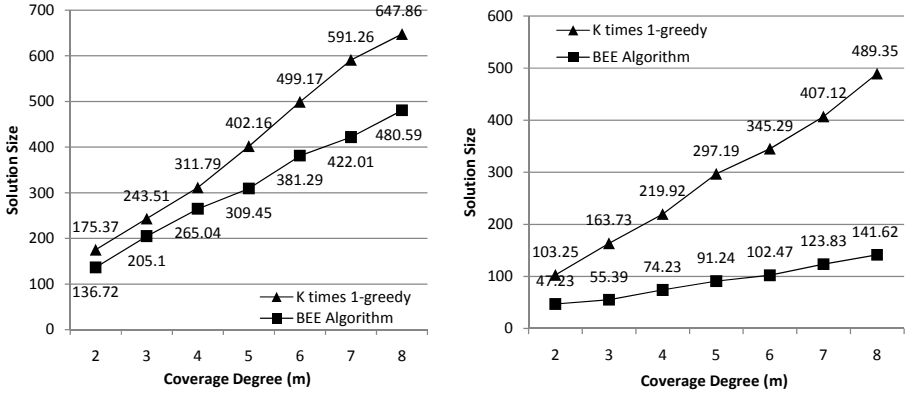

Fig. 4. Solution size of K-cover delivered by two algorithms with different K. Here the size of our sensor database 3000.

Figure 4 (a) shows the solution size delivered by two algorithms with different values of coverage degrees when the sensing radius is 4 units. Note that the results returned by each algorithm almost constitute a line in the 2-D plane. It means that to get larger coverage degree *m*, we need to select more sensors. Also we can see that the slope of BEE algorithm is much smaller than the slope of the K times 1-greedy algorithm, which proves the efficiency of BEE algorithm. This means the BEE algorithm uses less duplicated sensors than K times 1-greedy algorithm when the sensing radius increases.

Figure 4 (b) shows the solution size delivered by two algorithms with different values of coverage degree when the sensing radius is 8 units. We have similar conclusion as in Figure 4 (a). Note that the slope of BEE algorithm in (b)is smaller than in (a). From this, we can observe that the redundancy decreases when the sensing radius increases.

5.3 Summary

From the above two figures, we can see that the BCP algorithm and BEE algorithm did better that the K times 1-greedy algorithm for different size of the sensor database system, different sensing radius and different value of coverage degree. By apply these algorithms, we can have less redundant sensors in the system to guarantee an efficient, energy-saving and robust system. We can conclude that our algorithms are really efficient. Thus, our algorithms become a new approach to solve coverage problem in sensor database.

6 Conclusion

In this paper, to deal with coverage problem in sensor database system, we introduce minimum connected set cover (SDC) problem and k-connected m-set cover problem ((k,m)-SDC) for fault-tolerance. Moreover, we provide two approximation algorithms for SDC problem in general sensor database systems. Logarithm performance guarantee was obtained, incorporating a new parameter D_c which measures the maximum distance between two sets covering a common element. We also give a logarithm approximation algorithm for Minimum (k,m)-SDC problem with fixed $k = 2$, using a new parameter $PD(G)$ which in fact measures the maximum length of an ear. These are the first algorithms for SDC problems in general graphs with guaranteed performance ratio. These two algorithms can become a new approach to deal with coverage problem in sensor database.

To improve the performance ratio is one of our future directions. To study the Minimum (k,m)-SDC problem for $k \geq 3$ is another direction. Weighted version of SDC problem is also an interesting topic. However, the methods used in this paper can not be generalized for that. A lot of deep insights and new ideas are needed.

References

1. Wilschut, A.N., Apers, P.M.G.: Dataflow Query Execution in a Parallel Main-Memory Environment. Distributed and Parallel Databases 1(1), 103–128 (1993)
2. Bonnet, P., Gehrke, J., Seshadri, P.: Towards Sensor Database Systems. Mobile Data Management, 3–14 (2001)
3. Cheng, R., Prabhakar, S.: Managing Uncertainty in Sensor Database. ACM SIGMOD 32(4), 41–46 (2003)
4. Estrin, D., Govindan, R., Heidemann, J.: Embedding the Internet: Introduction. Communications of the ACM Journal 43(5), 38–41 (2000)
5. Gonen, M., Shavitt, Y.: A Θ (log n)-Approximation for the Set Cover Problem with Set Ownership. Information Processing Letters 109, 183–186 (2009)
6. Govindan, R., Hellerstein, J.M., Hong, W., Madden, S., Franklin, M., Shenker, S.: The Sensor Network as a Database, USC Computer Science Department Technical Report (September 2002)
7. Hellerstein, J.M., Avnur, R., Ranman, V.: Informix under CONTROL: Online Query Processing. Data Mining and knowledge Discovery 4(4) (October 2000)
8. Seshadri, P., Livny, M., Ramaakrishman, R.: SEQ: A Model for Sequence Databases. In: Proceedings of the 11th International Conference on Data Engineering (ICDE), pp. 232–239 (1995)
9. Abrams, Z., Goel, A., Plotkin, S.: Set k-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks. In: Proceedings of the 3rd Conference on Information Processing in Sensor Networks, IPSN 2004 (2004)
10. Cardei, M., Wu, J.: Energy-Efficient Coverage Problems in Wireless Ad-Hoc Sensor Networks. Computer Communications 29(4), 413–420 (2006)
11. Cardei, M., Thai, M., Li, Y., Wu, W.: Energy-Efficient Target Coverage in Wireless Sensor Networks. In: Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 2005), Miami, Florida USA, March 13-17, pp. 1976–1984 (2005)

12. Cardei, M., Du, D.Z.: Improving Wireless Sensor Network Lifetime through Power Aware Organization. ACM Wireless Networks 11(3), 333–340 (2005)
13. Cerdeira, J.O., Pinto, L.S.: Requiring Connectivity in the Set Covering Problem. Journal of Combinatorial Optimization 9, 35–47 (2005)
14. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn (2002)
15. Feige, U.: A Threshold of $\ln n$ for Approximating Set Cover. In: Proceedings of the 28th ACM Symposium on Theory of Computing (ACM 1996), pp. 314–318 (1996)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability. W.H. Freeman and Company, New York (1979)
17. Gupta, H., Das, S.R., Gu, Q.: Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. In: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2003 (2003)
18. Huang, C.F., Tseng, Y.C.: The Coverage Problem in a Wireless Sensor Network. In: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, pp. 115–121 (2003)
19. Jaggi, N., Abouzeid, A.A.: Energy-Efficient Connected Coverage in Wireless Sensor Networks. In: Proceedings of 4th Asian International Mobile Computing Conference, Kolkata, India, pp. 77–86 (2006)
20. Li, X.Y., Wan, P.J., Frieder, O.: Coverage in Wireless Ad-Hoc Sensor Networks. IEEE Transactions on Computers 52(6), 753–763 (2003)
21. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In: OSDI (2002)
22. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage Problems in Wireless Ad-Hoc Sensor Networks. In: Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), vol. 3, pp. 1380–1387 (2001)
23. Shuai, T.-P., Hu, X.: Connected Set Cover Problem and its Applications. In: Cheng, S.-W., Poon, C.K. (eds.) AAIM 2006. LNCS, vol. 4041, pp. 243–254. Springer, Heidelberg (2006)
24. Suurballe, J.W., Tarjan, R.E.: A Quick Method for Finding Shortest Pairs of Disjoint Paths. Networks 14, 325–336 (1984)
25. Tague, P., Lee, J., Poovendran, R.: A Set-Covering Approach for Modeling Attacks on Key Predistribution in Wireless Sensor Networks, Technical Report CACR, 41 (2005)
26. Thai, M.T., Wang, F., Du, H., Jia, X.: Coverage Problems in Wireless Sensor Networks: Designs and Analysis. International Journal of Sensor Networks, special issue on Coverage Problems in Sensor Networks 3(3), 191–200 (2008)
27. Zhou, Z.H., Das, S., Gupta, H.: Connected K-Coverage Problem in Sensor Networks. In: Proceedings of the 13th International Conference onComputer Communications and Networks (ICCCN 2004), pp. 373–378 (2004)