On Multiple Point Coverage in Wireless Sensor Networks

Shuhui Yang†, Fei Dai‡, Mihaela Cardei†, and Jie Wu†
†Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
‡Department of Electrical and Computer Engineering
North Dakota State University
Fargo, ND 58105

***Abstract*— We consider a wireless sensor network consisting of a set of sensors deployed randomly. A point in the monitored area is covered if it is within the sensing range of a sensor. In some applications, when the network is sufficiently dense, area coverage can be approximated by guaranteeing point coverage. In this case, all the points of wireless devices could be used to represent the whole area, and the working sensors are supposed to cover all the sensors. Many applications related to security and reliability require guaranteed k-coverage of the area at all times. In this paper, we formalize the k-(Connected) Coverage Set (k-CCS/k-CS) problems, develop a linear programming algorithm, and design two non-global solutions for them. Some theoretical analysis is also provided followed by simulation results.**

I. Introduction

In wireless sensor networks (WSNs), one design challenge is to save limited energy resources to prolong the lifetime of the network. A duty cycle is therefore introduced to allow each sensor to switch between active and sleep modes to save energy. On the other hand, a certain amount of active nodes should be present to ensure a desired level of coverage at all times. The way to select active nodes is called *coverage* and the method to rotate the role of each sensor to meet certain objectives is called *scheduling*, where nodes alternate between active and sleeping modes.

In a WSN, a sensor covers a target if the target is in the sensing range of the sensor. There exist three coverage models depending on how targets are defined:

1) Targets form a contiguous region and the objective is to select a subset of sensors to cover the region [18]. Typical solutions involve geometry properties based on the positions of sensor nodes.
2) Targets form a contiguous region and the objective is to select a subset of sensors to cover the rest of sensors [4]. This model assumes the network is sufficiently dense so that point coverage can approximate area coverage. Typical solutions involve constructing dominating sets or connected dominating sets [20] based on traditional graph theory.
3) Targets are discrete points and the objective is to select a subset of sensors to cover all of the targets. Typical solutions [3] use the traditional set coverage or bipartite graph models.

The desired level of coverage can be defined as a multiple coverage for the purpose of reliability in case of failure or for other applications related to security (e.g., localized intrusion detection) or localization (e.g. triangulation-based positioning).

In this paper, we focus on the second coverage problem. We first formalize the k-(Connected) Coverage Set (k-CCS/k-CS) problems in terms of linear programming, and an approximation algorithm based on integer programming is developed for the k-CS problem. We then propose two non-global k-coverage solutions. One is quasi-local cluster-based with a deterministic bound, the other is localized with a proven probabilistic bound. Two versions of each solution will be considered, one with connectivity for k-CCS and the other without connectivity for k-CS. Using a custom simulator, we compare the effectiveness of the proposed approaches with other local solutions to the same problem. Our contributions in this paper are the following:

1) Define and formalize the k-(Connected) Coverage Set problems (k-CCS/k-CS).
2) Develop a global algorithm for the k-CS problem

The work was supported in part by NSF grants ANI 0083836, CCR 9900646, CNS 0422762, CNS 0434533, and EIA 0130806.
Corresponding authors: syang1@fau.edu, jie@cse.fau.edu.

using linear programming.
3) Design two non-global solutions for k-CS/k-CCS.
4) Conduct performance analysis, through analytical and simulation studies on all the proposed solutions.

The remainder of the paper is organized as follows: Section II reviews the related work in the field. Section III gives the problem definition of the k-CS/k-CCS problems. Section IV presents the linear programming algorithm for k-CS. Section V proposes the quasi-local solution and the local solution for the problems. The theoretical bounds of them are also given in this section. The performance study through simulation is conducted in Section VI. The paper concludes in Section VII.

II. Related Work

Several local solutions exist to maintain 1-coverage in a wireless sensor network. Most of them rely on location information. A pruning method was proposed in [18], where a sensor can switch to sleep mode, if its sensing area is covered by sensing areas of its neighbors. As the calculation of sensing area coverage becomes tedious, some simplifications have been used. One method is to use a grid system [23], where the sensing area is represented by the grid points within this area, and area coverage is approximated by point coverage. In another method [22], the deployment area is divided into small squares. After one sensor is elected to be active in each square, other nodes can switch to sleep mode. The following probing-based solution [24] does not rely on location information. Basically, each sensor tries to detect activities of its neighbors. It switches to sleep mode if some active neighbors are detected; otherwise, it switches to active mode.

For k-coverage, a global method was proposed in [15] to construct k separate sets, each set achieving 1-coverage. Together, these sets provide k-coverage. A local solution was provided in [1] to solve the same problem.

When the objective is to cover individual targets, dominating set algorithms [3] [5] that achieve point coverage should be considered. The problems of double point coverage and k-point coverage in general have been studied in [8]. In [11], three heuristic algorithms are provided to achieve double point coverage. Localized 2-coverage algorithms were discussed in [14]. Dai and Wu [6] has proposed several local algorithms to construct a k-connected k-dominating set. In this paper, we propose to maintain 1-connectivity rather than k-connectivity, to reduce the size of the k-coverage set.

To operate successfully, a sensor network must also provide satisfactory connectivity so that nodes can communicate for data fusion and reporting to base stations. A straightforward solution is to use a communication range (R) that is at least twice the sensing range (r), such that area coverage implies connectivity of active sensors [26]. This conclusion was generalized in [19]: When $R \geq 2r$, a sensor network that achieves k-coverage is k-connected. More analysis can be found in [17].

Jiang et al [10] considered a local solution for k-coverage and extended point coverage to area coverage using a notion of *biggest vacant square territory* (BVST). We will discuss this scheme under the zero BVST, since the area coverage is not an issue here. The basic idea is to apply a local solution to put as many sensors to sleep as possible while ensuring a full 1-coverage assuming the sensor range r is the same as the transmission range R. Then, r is enlarged to ensure k-coverage. Specifically, to ensure k-coverage, r should be set to be at least $[\sqrt{2}+1+(\sqrt{2}/2+2)i]\ R$, where integer i is a minimum value satisfying $\sum_{j=1}^{j=i} 4j \geq k-1$.

III. Problem Definitions

We consider a wireless sensor network consisting of n homogeneous wireless devices (sensors) $s_1, s_2, \ldots, s_n$. To reduce energy consumption while increasing security and reliability, we want to select a minimum subset of sensors with the property that each sensor is monitored by at least k sensors in the selected subset.

We model the network as an undirected graph $G = (V, E)$, with the set of vertices (or nodes) being the set of sensors. An edge exists between two nodes if the two corresponding sensors are each within the other's communication range. We assume that the network is sufficiently dense, such that the network is connected, and each node has at least k neighbors for a given constant k. Let us now introduce the problem definitions.

k-Coverage Set (k-CS) Problem: Given a constant $k > 0$ and an undirected graph $G = (V, E)$ find a subset of nodes $C \subseteq V$ such that (1) each node in V is dominated (covered) by at least k different nodes in C, and (2) the number of nodes in C is minimized.

k-Connected Coverage Set (k-CCS) Problem: Given a constant $k > 0$ and an undirected graph $G = (V, E)$ find a subset of nodes $C \subseteq V$ such that (1) each node in V is dominated (covered) by at least k different nodes in C, (2) the number of nodes in C is minimized, and (3) the nodes in C are connected.

k-CS and k-CCS are extensions of the Dominating Set (DS) and Connected Dominated Set (CDS) problems

[20]. A set is dominating if every node in the network is either in the set or a neighbor of a node in the set. When a DS is connected, it is denoted as a CDS; that is, any two nodes in the DS can be connected through intermediate nodes from the DS. CDS as a connected virtual backbone has been widely used for broadcast process [16], searching in a reduced space, and point coverage in wireless sensor networks [4]. When $k = 1$, k-CS (k-CCS) problem reduces to the DS (CDS) problem. Therefore, for $k = 1$, both k-CS and k-CCS are NP-complete [9].

IV. A Global Solution for the k-CS Problem

In this section, we first formulate the k-CS problem using Integer Programming (IP) and then present the LP-based approximation algorithm.

Given:

- n nodes s_1, ...,s_n
- a_{ij}, the coefficients showing the coverage relationship between nodes. These coefficients are defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if node } s_i \text{ is covered by node } s_j \\ 0 & \text{otherwise} \end{cases}$$

Variables: x_j, boolean variable, for $j = 1 \dots n$:

$$x_j = \begin{cases} 1 & \text{if node } s_j \text{ is selected in the subset } C \\ 0 & \text{otherwise} \end{cases}$$

Integer Programming:

$$\begin{array}{lll} \text{Minimize} & x_1 + x_2 + \ldots + x_n & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \geq k & \text{for all } i = 1, \ldots, n \\ & x_j \in \{0, 1\} & \text{for } i = 1, \ldots, n \end{array} \quad (1)$$

The constraint $\sum_{j=1}^{n} a_{ij} x_j \geq k$, for all $i = 1, \ldots, n$, guarantees that each node in V is covered by at least k nodes in C. Let us note with Δ the maximum node degree in G. We extend the results presented in [7], [9], to our problem and design a ρ-approximation algorithm, where $\rho = \Delta + 1$. Since IP is NP-hard, we first *relax* the IP to Linear Programming (LP), solve the LP in linear time, and then *round* the solution in order to get a feasible solution for the IP.

Relaxed Linear Programming:

$$\begin{array}{lll} \text{Minimize} & x_1 + x_2 + \ldots + x_n & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \geq k & \text{for all } i = 1, \ldots, n \\ & 0 \leq x_j \leq 1 & \text{for } i = 1, \ldots, n \end{array} \quad (2)$$

Next, we present our ρ-approximation algorithm, where $\rho = \Delta + 1$. Based on the optimal solution x^* of the relaxed LP, we compute a solution $\bar{x}$ for the IP. When the algorithm terminates, the set C contains the k-coverage set.

Algorithm 1: LP-based Algorithm (LPA)

1) $C = \phi$
2) Let x^* be an optimal solution of the Relaxed Linear Programming
3) For each $j = 1, \ldots, n$ do:
   a) If $x_j^* \geq 1/\rho$, then $\bar{x}_j = 1$ and $C = C \cup \{s_j\}$
   b) If $x_j^* < 1/\rho$, then $\bar{x}_j = 0$
4) Return C

The complexity of this algorithm is dominated by the linear programming solver. The best performance is $O(n^3)$ using Ye's algorithm [25], where n is the number of variables.

Theorem 1: The LP-based algorithm is an ρ-approximation algorithm for the k-CS problems, where $\rho = \Delta + 1$ and Δ is the maximum node degree in G.

Proof: We first note that $\rho = \Delta + 1 = \max_{1 \leq i \leq n} \sum_{j=1}^{n} a_{ij}$. Next, we show that our algorithm is an ρ-approximation of the optimal solution. Based on the way we set $\bar{x}$, it is clear that $\bar{x}_j \leq \rho \cdot x_j^*$, for any $j = 1, \ldots n$. Therefore, $\sum_{j=1}^{n} \bar{x} \leq \rho \sum_{j=1}^{n} x_j^*$.

Next, we claim that by rounding the fractional values of the variables x^*, we obtain $\bar{x}$, a feasible solution for the initial IP. For this we need to show that $\sum_{j=1}^{n} a_{ij} \bar{x}_j \geq k$ for any $i = 1, \ldots, n$. This guarantees that the subset C output by our algorithm k-covers all the nodes.

Let us divide $\bar{x}$ into two subsets, $I_1 = \{j | x_j^* < \frac{1}{\rho}\}$ and $I_2 = \{j | x_j^* \geq \frac{1}{\rho}\}$. Then for any $i = 1, \ldots, n$ we have $\sum_{j \in I_1} a_{ij} x_j^* < \frac{1}{\rho} \sum_{j \in I_1} a_{ij} \leq 1$; therefore, $\sum_{j \in I_1} a_{ij} x_j^* < 1$. Also, $\sum_{j=1}^{n} a_{ij} \bar{x}_j \geq \sum_{j \in I_2} a_{ij} x_j^* \geq k - \sum_{j \in I_1} a_{ij} x_j^* > k - 1$. Since both $\sum_{j=1}^{n} a_{ij} \bar{x}_j$ and k are integers, it follows that $\sum_{j=1}^{n} a_{ij} \bar{x}_j \geq k$. ■

V. Non-Global Solutions for the k-CS/k-CCS Problems

This section starts with a cluster-based solution which is quasi-local, followed by a local solution for the k-CS/k-CCS problems. Some examples and bounds of the solutions are shown.

A. A Cluster-based Solution

In [21], the cluster-based CDS protocol is classified as a quasi-local solution, since it is based on mainly

local state information and occasional partial global state information. In this subsection, we propose a scheme for the k-CS/k-CCS problems, which is based on the traditional clustering algorithm:

- Sequentially apply a traditional clustering algorithm k times, whereby the clusterheads selected each time are marked and removed immediately from the network.
- Find gateways to connect the first set of the clusterheads and also mark them.
- For each marked node (clusterhead or gateway), if it does not have k marked neighbors, it designates some unmarked neighbors to be marked.

The clustering algorithm divides the network into several clusters, and each has a clusterhead and several neighbors of this clusterhead as members. Any two clusterheads are not neighbors, and the clusterhead set is a maximum independent set (MIS) of the network in addition to a DS. The marked k sets of clusterheads together with one set of gateways form the k-CCS. For coverage without connectivity, the second step, gateway selection, can be removed. Note that gateway selection can be tree-based, whereby gateways are selected globally to make the CDS a tree, or mesh-based, whereby each clusterhead is connected to all of its neighboring clusterheads, and thus the CDS is a mesh structure. The implementation follows. Initially, all the nodes are unmarked. When the algorithm terminates, all the marked nodes (clusterheads or gateways) form the k-CCS/k-CS.

Algorithm 2: Cluster-based k-CS/k-CCS Algorithm (CKA)

1) Using a clustering algorithm to select clusterheads, set C_1, and the selected clusterheads are marked and removed from the network.
2) Repeat step 1 k times, mark and remove C_i, $i = 2, \ldots, k$.
3) Use a gateway selection approach to select gateways, set D, to connect clusterheads in the first set, C_1, and mark nodes in D. (This step is removed for a solution without connectivity.)
4) For each node in $C_1 \cup C_2 \cup \ldots\ C_k \cup D$ (clusterhead or gateway), if the number of its marked neighbors, t, is smaller than k, it designates k-t unmarked neighbors to be marked.

Theorem 2: All the clusterheads (and gateways) marked in CKA form a k-CS (k-CCS) of the networks.

Proof: Let us assume that the network is $G = (V, E)$, and the clusterheads selected in round i are sets C_i, $i = 1, \ldots, k$. We first prove that all the unmarked nodes can be covered k times. If a node u is not marked, it must be the neighbor of a node in C_i in round i. Therefore, there are k nodes from each of the sets C_1, C_2, ..., C_k, that are neighbors of u, and u is covered k times by the set $C = \sum C_i, i = 1, \ldots, k$.

Then we prove the connectivity. Let us assume D is the gateway set of clusterhead set C_1. For any two nodes u, v, ($u, v \in C$), we now prove there is a path which contains only nodes in C to connect them. When $u, v \in C_1$, they are connected since $C_1 \cup D$ is a CDS. When u is not in C_1, u must have a neighbor u', $u' \in C_i$. This is because C_1 is a DS of G. The same is with v. Therefore, u and v are connected through C_1.

Finally we prove that all the marked nodes themselves can also be covered k times by other marked nodes. This is obvious, because step 4 of CKA guarantees it. ■

A traditional algorithm [12] takes $O(n)$ rounds in the worst case, in a network with n nodes. A randomized clustering algorithm [13] has been proposed to achieve 1-coverage in $O(\log^3 n)$ time with high probability. This algorithm can be easily extended to achieve k-coverage in $O(k \log^3 n)$ time with high probability.

B. A Local Solution

In this subsection, a local solution, PKA, for k-CS/k-CCS is developed that is based on only local neighborhood information. A node u is "k-covered" by a subset of C of its neighbors if and only if three conditions hold:

- The subset C is connected by nodes with higher priorities than u.
- Any neighbor of u is a neighbor of at least k nodes from C.
- Each node in C has a higher priority than u.

For coverage without connectivity, the first constraint can be removed. The following algorithm provides an implementation where each node determines its status (marked or unmarked) based on its 2-hop neighborhood information. Initially, it is assumed that all nodes are marked. After the algorithm terminates, all the marked nodes form the k-CCS/k-CS.

Theorem 3: The marked nodes from PKA form a k-CS/k-CCS of the network.

Proof: Let us assume a node u is unmarked. Then according to PKA, there exists a set C, $C = \{s | s \in N(u), L(s) > L(u)\}$, and every node v in $N(u)$ has at least k neighbors in C. That is to say, u is not in the highest k rank (based on priority) nodes of v; thus u is safe to be unmarked. Therefore, for each node v in

Algorithm 3: Pruning-based k-CS/k-CCS Algorithm (PKA)

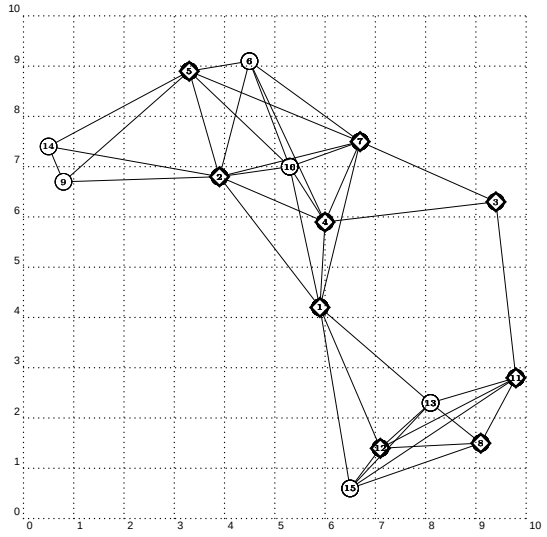
1) Each node u is given a unique priority, $L(u)$, and each node u is represented by tuple ($L(u)$, $ID(u)$).
2) Each node broadcasts its neighbor set $N(u)$, where $N(u) = \{v | v$ is a neighbor of $u\ \}$.
3) At node u, build a subset: $C(u) = \{v | v \in N(u), L(v) > L(u)\}$. Node u is unmarked if:
   a) subset $C(u)$ is connected by nodes with higher priorities than u (this constraint is removed for a solution without connectivity), and
   b) for any node $w \in N(u)$, there are k distinct nodes in $C(u)$, say v_1, v_2, ..., v_k, such that $w \in N(v_i)$.

G, its k highest rank neighbors do not have chances to unmark. Every node in the network is covered k times by the marked nodes.

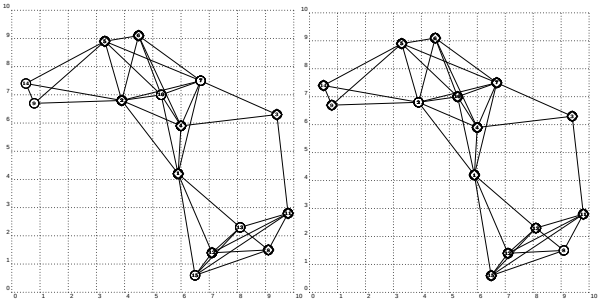
As to the connectivity, when condition 3 holds, the node set marked by PKA is a superset of the node set marked by pruning Rule K algorithm [5] on the network, which takes k as 1. Thus the connectivity is guaranteed. ■

C. Examples

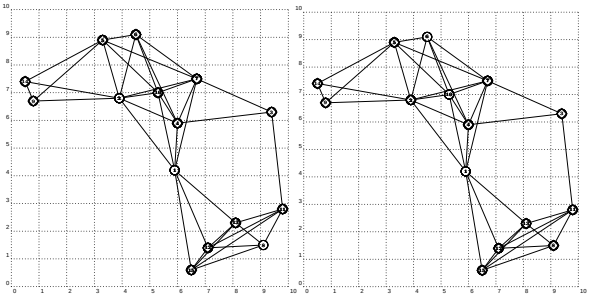
Figure 1 is the small scale example. There are 15 nodes in the network. The transmission range is 4 and k is 2. The minimum node degree in the network is not less than 2. The nodes marked with diamonds form the resultant k-CS or k-CCS in the figures. In (a), there are 9 nodes in the resultant k-CCS using CKA. We can see that all the marked nodes are connected, and every node in the network is covered at least twice by the marked nodes. (b) shows the k-CS of size 9 after the CKA (without connectivity). Although the size is the same as that of the k-CCS in (a), the marked nodes are different. Generally speaking, the size of resultant k-CS by CKA is smaller than that of k-CCS, but this is not necessary. This is because, according to CKA, the gateway selection and the k times of clusterhead selection are independent, and when the last step checks all the marked nodes, additional marked gateways may help to prevent adding more nodes in the set. (c) is the resultant k-CCS with the size of 13 by the PKA. (d) shows the k-CS by the PKA (without connectivity). There are 12 nodes in the set. Compared with (c), node 1 is not marked. This is because, neighbors of 1 form two connected components. One is nodes 4, 7, and 10, and the other is nodes 12, 13, and 15. Neither of these two components can satisfy the three constrains for k-CCS. But if they combine together, they are qualified. Therefore, node 1 unmarks itself in k-CS constructing. (e) is k-CS by LPA, and there are 12 nodes in it.



(a) k-CCS by CKA of size 9



(b) k-CS by CKA of size 9 (c) k-CCS by PKA of size 13



(d) k-CS by PKA of size 12 (e) k-CS by LPA of size 12

Fig. 1. A small scale example ($n = 15, k = 2, r = 40$).

D. Theoretical bounds

Let CKA_k be the backbone constructed by the cluster-based algorithm CKA. Similarly, let PKA_k denote the backbone constructed by the pruning algorithm PKA that

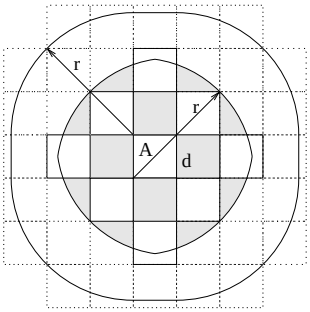


Fig. 2. For any node in region A, placing k nodes in each gray region is sufficient to k-cover its neighbors.

k	2	3	4
Jiang's	325.50	325.50	325.50
CKA	22.20	27.45	33.75
PKA	23.35	32.10	40.55

TABLE I
COMPARISON OF JIANG'S, CKA, AND PKA
($r = 40, n = 1000, r' = 7$).

achieves k-coverage, and OPT_k be the minimal node set that achieves k-coverage. We prove that the size of CKA_k is $O(k^2)$ times the size of OPT_k in the worst case, and the average size of PKA_k is $O(k)$ times the size of OPT_k in random wireless sensor networks.

Theorem 4: In a unit disk graph, $|CKA_k| = O(k^2) \cdot |OPT_k|$ for all $k \geq 1$.

Proof: From the cluster-based algorithm, $CKA_k = C_1 \cup C_2 \cup \ldots \cup C_k \cup D \cup C'_k$, where C_i ($1 \leq i \leq k$) is the set of clusterheads selected in round i, D is the set of gateways to connect C_1, and C'_k is the set of nodes added in the last step to ensure k-coverage of marked nodes. It has been proved in [2] that

$$|C_i| = O(1) \cdot |OPT_1|$$

for $1 \leq i \leq k$ and

$$|D| = O(1) \cdot |OPT_1|$$

Therefore, the number of marked nodes before the last step is

$$|C_1| + |C_2| + \ldots + |C_k| + |D| = O(k) \cdot |OPT_1|$$

Note that in the last step, at most k neighbors of each marked node are added in C'_k. That is,

$$|C'_k| \leq k(|C_1| + |C_2| + \ldots + |C_k| + |D|)$$

Combine the above equations, we have $|CKP_k| \leq (k+1)(|C_1| + |C_2| + \ldots + |C_k| + |D|) = (k+1)O(k) \cdot |OPT_1| = O(k^2) \cdot |OPT_1|$ When $k \geq 1$, $|OPT_1| \leq |OPT_k|$, and $|CLS_k| = O(k^2) \cdot |OPT_k|$. ■

Theorem 5: In random unit disk graphs, $E(|PKA_k|) = O(k) \cdot |OPT_k|$ for all $k \geq 1$.

Proof: Consider a square region A with side $d = r/2\sqrt{2}$ (diagonal line $r/2$). As shown in Figure 2, if A is not empty, neighbors of nodes in A are within a $7\times7-4 = 45$ square region surrounding A. These square regions can be k-covered by putting k nodes in each of the 12 gray regions. Note that these $12k$ nodes are all neighbors of an arbitrary node in A. In addition, these nodes connected via themselves. Suppose these nodes do exist, and among them node v has the lowest priority, then all nodes in A with a lower priority than v can be unmarked. That is, $PKA_A \subseteq \{u | L(u) \geq L(v)\}$, where PKA_A is the set of marked nodes in A.

Let V_A be the set of all nodes within these 45 squares. We sort V_A in the descending order of node priority, and denote them by their *ranks* $1, 2, \ldots, |V_A|$ in the sorted list. The node with the highest priority has the lowest rank 1. Let $AV_A \subseteq V_A$ be the set of $12k$ nodes with minimum ranks that k-covers all neighbors of A, and R_A be the maximal rank of nodes in AV_A. From the above discussion, $|PKA_A| \leq R_A$. In the following discussion, we prove that $E(R_A) = O(k^2)$.

Partition V_A into k subsets $V_A^1, V_A^2, \ldots, V_A^k$, where V_A^i consists of nodes with ranks $i, k+i, 2k+i, \ldots$ for $1 \leq i \leq k$. Let $AV_A^i \subseteq V_A^i$ be the set of 12 nodes with minimum ranks that 1-covers all neighbors of A, and R_A^i be the maximal rank of nodes in AV_A^i. Obviously,

$$R_A \leq \max_{1 \leq i \leq k} R_A^i \leq \sum_{i=1}^{k} R_A^i$$

It has been proved in [5] that when nodes and node priorities are randomly distributed, $E(R_A^i/k) = O(1)$ for $1 \leq i \leq k$. Therefore

$$E(R_A) \leq \sum_{i=1}^{k} E(R_A^i) = \sum_{i=1}^{k} kE(R_A^i/k) = O(k^2)$$

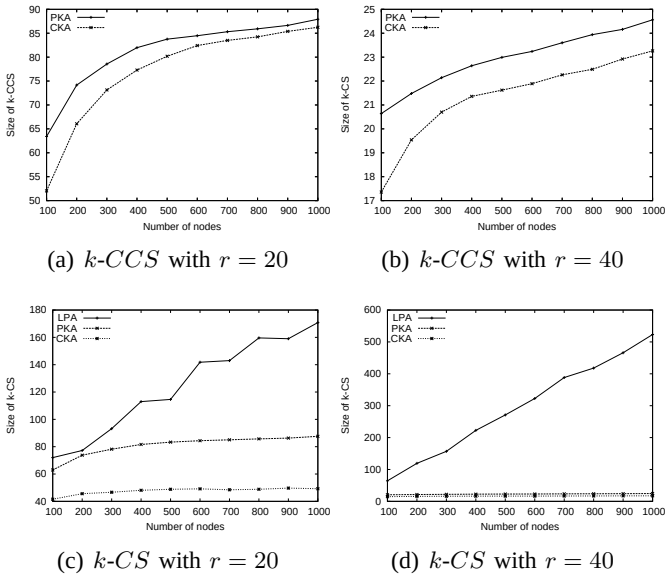
Since each non-empty region A_i is covered by at least k nodes from OPT_k, and each nodes in OPT_k can cover at most $O(1)$ such regions, the total number of non-empty regions in the network is

$$N = O(1/k) \cdot |OPT_k|$$

From the above discussion, the average number of marked nodes in each non-empty region A_i is $O(k^2)$. Therefore,

$$E(|PKA_k|) \leq \sum_{i=1}^{N} E(R_A) = N \cdot O(k^2) = O(k) \cdot |OPT_k|$$

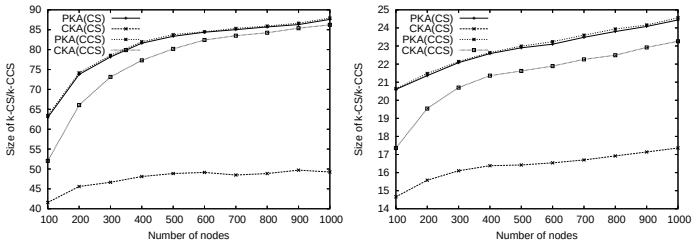
■



(a) k-CCS with $r = 20$ (b) k-CCS with $r = 40$

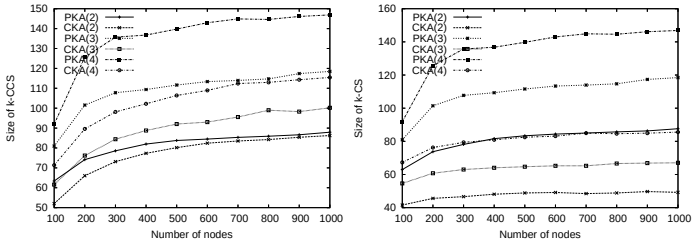
(c) k-CS with $r = 20$ (d) k-CS with $r = 40$

Fig. 3. Comparison of k-CCS & k-CS by different algorithms ($k = 2$).



(a) $r = 20$ (b) $r = 40$

Fig. 4. Algorithms with and without connectivity ($k = 2$).



(a) k-CCS (b) k-CS

Fig. 5. k-CCS & k-CS by PKA & CKA with $k = 2, 3, 4$ ($r = 20$).

VI. Simulation

This section presents results from our simulation. The linear programming approach (LPA) for k-CS, the k-coverage approach by Jiang et al (Jiang's) for k-CCS, the cluster-based algorithm with and without connectivity (CKA), and the pruning algorithm with and without connectivity (PKA) for k-CCS and k-CS are all evaluated and compared in the simulation. A mesh-based gateway selection algorithm is used in the CKA.

Linear programming is implemented using Matlab. All other approaches are implemented on a custom simulator. To generate a random network, n nodes are randomly placed in a restricted 100×100 area. We assume all nodes have the same transmission range. The tunable parameters in our simulation are as follows: (1) The node number n. We change the number of deployed nodes from 100 to 1000 to check the scalability of the algorithms. (2) The transmission range r. We use 20 and 40 as transmission ranges to produce the effect of link density on the algorithms. (3) The coverage parameter k. We use 2, 3, and 4 as its values. The performance metric is the number of nodes in the resultant k-CCS/k-CS. For each tunable parameter, the simulation is repeated 1000 times or until the confidence interval is sufficiently small ($\pm 1\%$, for the confidence level of 90%).

Table I compares the sizes of resultant k-CCS by Jiang's, CKA and PKA. Since in Jiang's algorithm [10], the biggest vacant square territory (BVST) is assumed to small enough, the network is quite dense. We use 1000 as the number of node and 40 as the sensing range in Jiang's. Thus the adjusted transmission range, r', is 7. We can see that CKA and PKA have better performance than Jiang's. This is because Jiang's is designed for the worst case bound, while CKA and PKA are based on average cases, and Jiang's does not generate a k-CCS set corresponding to every single value of k; Jiang's has smaller transmission range than CKA or PKA.

Figure 3 shows the comparison of the proposed LPA, CKA, PKA algorithms. (a) shows the resultant k-CCSs by CKA and PKA when transmission range is 20. (b) is when transmission range is 40. CKA has better performance than PKA because CKA is quasi-local while PKA is local algorithm. More information leads to a more precise precess. (c) shows the k-CSs by CKA, PKA and LPA when range is 20. Both CKA and PKA have better performance than LPA, especially when the node number is large. (d) is when range is 40. LPA has worse performance than CKA and PKA, especially when the network is dense. A dense network has a negative impact on the performance of LPA. This is because a dense network increases the maximum node degree, and thus the LPA's performance ratio. Additionally, a large maximum node degree decreases the k-set cover selection threshold ($1/\rho$), and therefore more nodes are added to the set C. As the theoretical results indicate, LPA performs better for sparse topologies.

Figure 4 shows the comparison of k-CS and k-

CCS by different algorithms using different transmission ranges, 20 in (a) and 40 in (b). We can see that k-CS by CKA has the smallest size, and next is the k-CCS by CKA. k-CS by PKA has almost the same size as k-CCS by PKA. Figure 5 shows the size of k-CCS in (a), and k-CS in (b) as parameter k varies. We can see that with larger k, the size of k-CCS or k-CS from CKA or PKA is larger. But when the number of node is great, this increase is less significant.

The simulation results can be summarized as follows: (1) CKA has better performance than PKA, especially in generating k-CS. (2) CKA and PKA have better performance than LPA, especially when network is relatively dense. (3) Greater k leads to larger sized k-CS/k-CCS. (4) CKA and PKA have better scalability than LPA, especially when the network is relatively dense. (5) LPA performs better in sparse topologies; a dense topology, with a large maximum node degree, negatively affects both LPA's performance ratio and the k-cover set selection threshold.

VII. Conclusion

In this paper, we have addressed the k-(Connected) Coverage Set (k-CS/k-CCS) problems in wireless sensor networks with the objective of minimizing the total energy consumption while obtaining k coverage for reliability. We have proposed one global solution for k-CS and two non-global algorithms. The first one uses a cluster-based approach to select backbone nodes to form the set. The second uses the pruning algorithm based on only 2-hop neighborhood information. We have also analyzed the performance of our algorithms through theoretical analysis and simulations.

References

[1] Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proc. of Information Processing in Sensor Networks*, 2004.

[2] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, and O. Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Trans. of Parallel and Distributed Systems*, 14(5):408–421, 2003.

[3] M. Cardei, J. Wu, M. Lu, and M. O. Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005.

[4] J. Carle and D. Simplot-Ryl. Energy efficient area monitoring by sensor networks. *IEEE Computer*, 37(2):40–46, 2004.

[5] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(10):908–920, 2004.

[6] F. Dai and J. Wu. On constructing k-connected k-dominating set in wireless networks. In *Proc. of IPDPS*, Apr. 2005.

[7] D. Z. Du. Design and analysis of approximation algorithms. Lecture Notes, Department of Computer Science, University of Minnesota.

[8] F. Harary and T. Haynes. Double domination in graphs. *ARS Combinatoria*, 55:201–213, 2000.

[9] D.S. Hochbaum. *Approximation Algorithms for NP-hard Problems*. PWS Publishing. Boston, MA, 1997.

[10] Z. Jiang, R. Kline, J. Wu, and F. Dai. Energy efficient k-coverage in wireless sensor networks with connected dominating set. *Submitted for publication*, 2005.

[11] H. Koubaa and E. Fleury. On the performance of double domination in ad hoc networks. In *Proc. of IFIP Medhoc*, 2000.

[12] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, 1996.

[13] T. Moscibroda and R. Wattenhofer. Efficient computation of maximal independent sets in unstructured multi-hop radio networks. In *Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[14] J. Shaikh, J. Solano, I. Stojmenovic, and J. Wu. New metrics for dominating set based energy efficient activity scheduling in ad hoc networks. In *Proc. of the International Workshop on Wireless Local Networks (WLN)*, 2003.

[15] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proc. of ICC*, 2001.

[16] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[17] D. Tian and N. Georganas. Connectivity maintenance and coverage preservation in wireless sensor networks. *Ad Hoc Networks Journal*. accepted to appear.

[18] D. Tian and N. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications*, 2002.

[19] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems*, 2003.

[20] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial M)*, 1999.

[21] J. Wu and W. Lou. Forward node set based broadcast in clustered mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 3(2):141–154, 2003.

[22] Y. Xu and J. Heidemann. Geography-informed energy conservation for ad hoc routing. In *Proc. of MOBICOM*, 2003.

[23] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In *Proc. of the First International Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[24] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Proc. of ICDCS*, 2003.

[25] Y. Ye. An $o(n^3l)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50, 1991.

[26] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Technical Report UIUC*. UIUCDCS-R-2003-2351, 2003.