CRNP: A Cover-based Relay Node Placement Algorithm to Delay-constrained Wireless Sensor Networks

Chaofan Ma[1], Wei Liang[2,3,*], and Meng Zheng[2,3]

1. Software College, Zhongyuan University of Technology, Zhengzhou 450007, China
2. Key Laboratory of Networked Control Systems, Chinese Academy of Sciences Shenyang 110016, China
3. Shenyang Institute of Automation, Chinese Academy of Sciences Shenyang 110016, China

chaofanma@hotmail.com, {weiliang, zhengmeng_6}@sia.cn

***Abstract*—Wireless Sensor Networks (WSNs) are gradually employed in many applications requiring real-time data transmission. As hop count is an important factor affecting end-to-end delay, in this paper, we investigate the Hop Constrained Relay Node Placement (HCRNP) problem where at least one path fulfilling the hop constraint is built between each Sensor Node (SN) and the sink. To address this problem, we present a Cover-based Relay Node Placement (CRNP) algorithm which places Relay Nodes (RNs) from SNs to the sink. Through formulating the deployment of RNs in each iteration as a cover problem (the set cover problem for arbitrary settings or the discrete unit disk cover problem for special settings) with respect to hop constraint, the CRNP algorithm iteratively deploys RNs adjacent to the SNs or the previously placed RNs so as to gradually connect SNs to the sink. Through rigorous analysis, we show that the CRNP algorithm has an approximation ratio better than existing algorithms for the HCRNP problem (i.e., O(1) for special settings and O($\ln n$) for arbitrary settings, where n is the number of SNs). Finally, we conduct extensive simulations to verify the effectiveness of the proposed algorithm.**

I. Introduction

Due to the benifits of low-cost, convenient-installation and easy-maintenance, Wireless Sensor Networks (WSNs) are gradually used in many applications that have constraints on end-to-end delay, e.g., factory automation and smart grid [1] [2]. In these applications, delay is a critical indicator to evaluate the network performance. For instance, in factory automation, the data gathered by Sensor Nodes (SNs) may be used for alarm notification or feedback control, and thus should be sent to the sink timely. This highlights the importance of the delay constrained Relay Node Placement (RNP) problem. As hop count can represent the end-to-end delay [3], this paper studies the Hop Constrained RNP (HCRNP) problem. Bhattacharya and Kumar [4] prove the NP-hardness of the HCRNP problem. Although this problem is closely related to the edge-weighted Steiner tree problem that has been well studied, the literature on the HCRNP problem are very limited [4]-[7].

*Corresponding author.

Furthermore, algorithms [4]-[7] to the HCRNP suffer different limitations. The Shortest Path Tree based Iterative Relay Pruning (SPTiRP) proposed in [4]-[5] are prone to fall into local optimum. The algorithm presented in [6] can only solve a special case of the HCRNP problem and is not applicable to large-scale problem. The Set-Covering-based Algorithm (S-CA) described in [7] employs a from-SNs-to-sink deployment method which limits its performance in approximation ratio and running time.

To solve the NP-hard HCRNP problem more effectively, this paper proposes a Cover-based Relay Node Placement (CRNP) algorithm, which deploys RNs iteratively from SNs towards the sink. To be specific, the CRNP algorithm decomposes the HCRNP problem into several cover subproblems subject to hop constraints, and in each iteration, the CRNP algorithm deploys a set of RNs to solve a cover subproblem. With the repetition of the CRNP algorithm, SNs are gradually connected to the sink in obedience to their hop constraints. In summary, the contributions of this paper are listed as follows:

- We decompose the HCRNP problem into several cover subproblems without violating the hop constraints imposed on SNs.
- We present a polynomial-time algorithm-CRNP whose approximation ratio is O($\ln n$) in general settings and O(1) in special settings for the HCRNP problem. To the best of our knowledge, there have been no algorithms achieving an approximation ratio better than that of the CRNP algorithm for the HCRNP problem.
- We conduct extensive simulations to evaluate the performance of the CRNP algorithm, and the simulation results show that the CRNP algorithm can save up to 25.11% deployed RNs in comparison to existing algorithms.

II. Problem Formulation

For ease of analysis and explanation, some rational assumptions are given as follows:

978-1-5386-1734-2/18/$31.00 ©2018 IEEE

- WSNs considered in this paper work on the widely-used many-to-one communication pattern [2], in which SNs transmit their sensed data to the sink via multi-hop paths.
- As in [2]-[7], this paper assumes that there exists a perfect schedule scheme such that there is no delay or packet drop resulting from collisions, queuing, congestions, etc.. End-to-end delay is measured based on hop count.
- Due to the existence of obstacles and forbidden regions in practical deployment environment, RNs cannot be placed at will. Therefore, as in [4]-[7], this paper assumes that RNs can only be placed at some predetermined Candidate Deployment Locations (CDLs).

Let X, Y and z denote a set of distributed SNs, a set of predetermined CDLs and the sink, respectively. The communication radii of SNs and RNs are r and R, respectively. Typically, $R \geq r$. For $\forall u, v \in X \bigcup Y \bigcup \{z\}$ $(u \neq v)$, if u and v are neighbors (i.e., u and v can communicate with each other directly), u and v should fulfill the following conditions:

- If $u \in X$ or $v \in X$, then u and v should meet $\|u - v\| \leq r$,
- if $u \notin X$ and $v \notin X$, then u and v should meet $\|u - v\| \leq R$,

where $\|u - v\|$ denotes the Euclidean distance between u and v. If u and v are neighbors, we also say u and v cover each other. All neighbors of node u is denoted by $\mathcal{N}(u)$. In this paper, a path between u and v is denoted by $p(u, v)$, and the hop count of this path is denoted by $\mathcal{H}(p(u, v))$. The hop constraint imposed on node u indicates the maximal hop count from u to the sink z, and is denoted by $\Delta(u)$. A path $p(z, u)$ is called a feasible path if $\mathcal{H}(p(z, u)) \leq \Delta(u)$. Let $\mathcal{S}(u, v)$ denote a shortest (i.e., least hop count) path between nodes u and v.

Definition 1 (HCRNP Problem): Given X, Y and z, the HCRNP problem seeks a minimum subset of CDLs Y to place RNs such that at least one feasible path can be built between each SN in X and the sink z.

III. ALGORITHM TO THE HCRNP PROBLEM

A. *Algorithm Description*

As a feasible solution for the HCRNP problem should be connected and there exists at least one spanning tree in each feasible solution, we thus plan to design a network topology that is a tree taking the sink as root and connecting all SNs to the HCRNP problem. Given a tree T and two different nodes of T, u and v, we denote $p_T(u, v)$ as a path of T between u and v. If each path between the sink and an SN is a feasible path, i.e., $\forall x \in X$, $\mathcal{H}(p_T(z, x)) \leq \Delta(x)$, we call T a feasible tree. In the beginning of CRNP, each SN x has a predetermined hop constraint $\Delta(x)$, and each CDL has an infinite hop constraint, i.e., $\forall y \in Y,\ \Delta(y) = +\infty$.

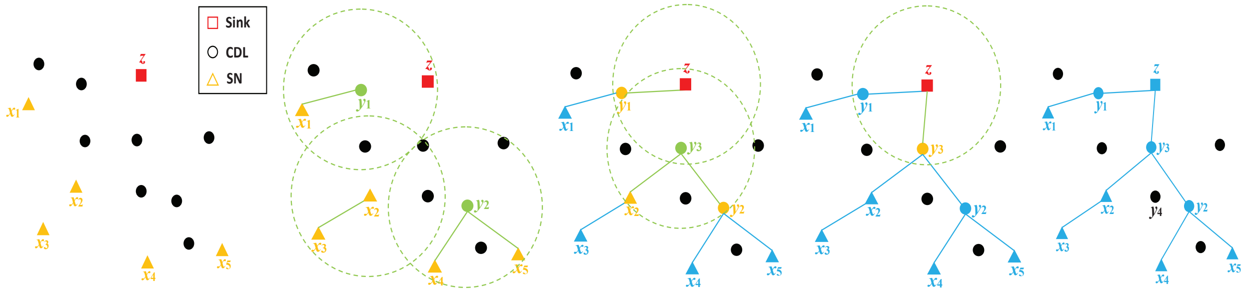
CRNP is composed of three steps. At the first step, CRNP checks whether the given problem is infeasible or can be solved without the help of RNs. In either case, CRNP terminates. Otherwise, CRNP carries out the second step. The objective of the second step of CRNP is to place some RNs such that the network connectivity is preliminarily built. As some redundant RNs may be introduced at the second step, the third step of CRNP is to reduce the number of deployed RNs. At the end of the third step, a feasible tree is returned.

The framework of CRNP is shown in Algorithm 1, in which lines 1-2 represent the implementation of the first step of CRNP. The second step and the third step of CRNP are explicitly described in Algorithm 2 and Algorithm 3, respectively. As the first step of CRNP is quite simple, the following description begins directly from the second step of CRNP.

Algorithm 1: Cover-based connected Node Placement (CRNP) algorithm.

Input: A set X of SNs, a set Y of CDLs, and a sink z.
Output: A tree T: if $T = \varnothing$, this implies that no feasible solutions exist for the connecting phase; otherwise, T is a feasible tree.
begin
1 $t1 \longleftarrow$ a shortest path tree taking z as root and including all SNs in X with the help of RNs placed at a subset of Y;
2 **if** $\forall x \in X,\ p_{t1}(x, z) \leq \Delta(x)$ **then**
3 $t2 \longleftarrow$ a shortest path tree rooted at z and including all SNs in X without the help of RNs;
4 **if** $\exists x \in X,\ p_{t2}(x, z) > \Delta(x)$ **then**
5 input X, Y and z into the second step of CRNP;
6 $\hat{Y} \longleftarrow$ a subset of Y returned by the second step of CRNP;
7 input X, $\hat{Y}$ and z into the third step of CRNP;
8 $T \longleftarrow$ a feasible tree returned by the third step of CRNP;
9 **else**
10 $T \longleftarrow t2$;
11 **else**
12 $T \longleftarrow \varnothing$;
13 **return** T;

As a CDL is selected to place RN, the terms RN and CDL are used interchangeably in the following paper. The second step of CRNP is achieved by placing RNs iteratively. In each iteration of the second step (except for the first iteration, in which a set of nodes are selected to fully cover the distributed SNs), a set of nodes (one node can be either an SN or a RN) are selected to fully cover the nodes selected in the previous iteration, and this iteration repeats until each SN can be connected to the sink via a feasible path. The second step is illustrated in Fig. 1, where the dashed circle denotes the communication range of a node, the blue point (triangle) denotes a selected CDL (SN) and the line denotes a link between two neighboring nodes. In the first iteration, a set of



(a) input nodes. (b) iteration 1. (c) iteration 2. (d) iteration 3. (e) the final topology.

Fig. 1. An illustration to the second step of CRNP.

nodes $\{y_1, y_2, x_2\}$ are selected such that each SN is covered by at least one selected node. In the next iteration, only a CDL y_3 is selected to cover y_2 and x_2 since y_1 is already covered by the sink. This iteration repeats until a feasible path is built between each SN and the sink. The final topology is shown in Fig. 1(e).

However, the concise illustration on Fig. 1 to the CRNP algorithm skips an important problem, i.e., how to check whether a node can be used to cover some other nodes without violating the hop constraints imposed on these nodes. This problem is termed as the feasibility problem in this paper. Taking Fig. 1(b) for example, the problem is whether feasible paths from the sink to x_4 and x_5 will pass through y_2, i.e.,whether the RN placed at y_2 can cover x_4 and x_5 in obedience to the hop constraints imposed on x_4 and x_5.

To address the feasibility problem, some definitions and notations are introduced. Suppose $(k-1)$ iterations have been carried out, and the kth iteration is about to start. The set of nodes selected in the $(k-1)$th iteration is denoted by I_{k-1} ($I_0 = X$ in the first iteration).

Definition 2: Node u is said to be **feasibly covered** by node v ($u \in X \bigcup Y$, $v \in X \bigcup Y \bigcup \{z\}$) if the following conditions are satisfied:

$$\mathcal{H}\left(\mathcal{S}(z,v)\right) \leq \Delta(u) - 1, \tag{1a}$$

$$u \in \mathcal{N}(v). \tag{1b}$$

Clearly, a node can feasibly cover itself. Let $\mathcal{C}(v)$ denote the set of all the nodes (in I_{k-1}) that can be feasibly covered by v. In addition, we stipulate that each node can feasibly cover itself, i.e., $v \in \mathcal{C}(v)$. Node v is called a feasible node for I_{k-1} when $|\mathcal{C}(v)| > 1$. If v is selected to cover a subset $\mathbb{C}(v)$ of $\mathcal{C}(v)$, the hop constraint on v is updated to

$$\Delta(v) = \min\left(\min_{u \in \mathbb{C}(v)} \left(\Delta(u) - 1\right), \Delta(v)\right). \tag{2}$$

Lemma 1: Let u be a node in I_{k-1}. In the kth iteration, we can only select a feasible node v meeting $u \in \mathcal{C}(v)$ to cover u while meeting the hop constraint $\Delta(u)$ imposed on u.

Proof: We prove this by applying the method of reduction ad absurdum. Let q be a node that cannot feasibly cover u, i.e., $u \notin \mathcal{C}(q)$. Then, we assume that q can cover u in obedience to $\Delta(u)$. This not only means u can be covered by q (i.e., $u \in \mathcal{N}(q)$), but also indicates there exists a path $p(z,q)$ satisfying $\mathcal{H}(p(z,q)) + 1 \leq \Delta(u)$, which is closely followed by $\mathcal{H}(\mathcal{S}(z,q)) \leq \Delta(u) - 1$. According to Definition 2, we can conclude that $u \in \mathcal{C}(q)$, which contradicts our assumption that q cannot feasibly cover u. ■

Algorithm 2: The second step of CRNP.

Input: A set X of SNs, a set Y of CDLs, and a sink z.
Output: A subset $\hat{Y}$ of Y.
begin
  $k \longleftarrow 0$, $I_k \longleftarrow X$, $U \longleftarrow Y \bigcup X$, $\hat{Y} \longleftarrow \varnothing$;
  foreach $u \in U$ **do**
    calculate $\mathcal{H}(\mathcal{S}(z,u))$ by applying the shortest path tree algorithm;
  foreach $u \in I_k$ **do**
    if $u \in \mathcal{N}(z)$ **then**
      $I_k \longleftarrow I_k - u$;
  while $I_k \neq \varnothing$ **do**
    foreach $u \in U$ **do**
      find the nodes (in I_k) that can be feasibly covered by u, i.e., $\mathcal{C}(u)$;
    $\hat{Y}_t \longleftarrow$a subset of U searched by an algorithm for the set cover problem or the DUDC problem to fully cover the nodes in I_k;
    $\hat{Y} \longleftarrow \hat{Y} \bigcup \hat{Y}_t$, $k \longleftarrow k+1$, $I_k \longleftarrow \hat{Y}_t$;
    foreach $u \in I_k$ **do**
      if $u \in \mathcal{N}(z)$ **then**
        $I_k \longleftarrow I_k - u$;
      else
        calculate $\Delta(u)$;
  $\hat{Y} \longleftarrow \hat{Y} - X$;
  return $\hat{Y}$;

The answer to the feasibility problem is given by Lemma 1, which implies that to cover a node u in I_{k-1} in obedience to $\Delta(u)$, we can only select an SN x or place an RN at a CDL y, where both x and y can feasibly cover u. Thus, in the kth iteration, we first search a set $\mathcal{E}$ of all the feasible nodes for I_{k-1} from $X \bigcup Y$. Then, we seek a minimum subset $\bar{\mathcal{E}}$

Algorithm 3: The third step of CRNP.

Input: A set X of SNs, a set $\hat{Y}$ of CDLs, and a sink z.
Output: A feasible tree T.
begin
  $T \longleftarrow$ a shortest path tree taking the sink as root and including all SNs in X and a subset of $\hat{Y}$;
  $u \longleftarrow$ the node with the least weight in $\hat{Y}$, $\hat{Y} \longleftarrow \hat{Y} - u$;
  while $u \neq \varnothing$ && $\exists y \in \hat{Y},\ y$ *is not marked as checked* **do**
    $tmpT \longleftarrow$ a shortest path tree taking the sink as root and including all the SNs by using the CDLs in $\hat{Y}$;
    if $tmpT$ *is a feasible tree* **then**
      $T \longleftarrow tmpT$;
      $u \longleftarrow$ the unchecked node with the least weight in $\hat{Y}$;
      $\hat{Y} \longleftarrow \hat{Y} - u$;
    else
      mark u as checked, $\hat{Y} \longleftarrow \hat{Y} \bigcup \{u\}$;
      $u \longleftarrow$ the least weighted unchecked CDL in $\bar{Y}$;
      $\hat{Y} \longleftarrow \hat{Y} - u$;
  return T;

of $\mathcal{E}$ to fully cover nodes in I_{k-1}, which is termed as the cover subproblem in the kth iteration. Obviously, in the general setting where SNs may have different hop constraints and the communication radii of SNs and RNs may be different, the cover subproblem in the kth iteration is mathematically a set cover problem, and typical solution algorithms to the set cover problem are rich in the literature. In this paper, the Greedy Set Cover (GSC) [8] is employed to find a set cover.

Next, we prove that in the special settings where SNs have the same hop constraint and the communication radii of SNs and RNs are identical (i.e., $\forall x \in X,\ \Delta(x) = \delta$, and $r = R$, where δ is a constant.), the cover problem in the kth iteration is a Discrete Unit Disk Cover (DUDC) problem.

Lemma 2: In the special settings, nodes selected to cover I_{k-1} in the kth iteration have the same hop constraint $(\delta - k)$.

Proof: We prove this statement by using the method of mathematical induction.

Basis: As all SNs have the same hop constraint δ, in the first iteration we can get that

$$\begin{aligned}\forall u \in I_1,\ \Delta(u) &= \min\left(\min_{v \in \mathbb{C}(u)} (\Delta(v) - 1), \Delta(u)\right) \\ &= \delta - 1,\end{aligned} \tag{3}$$

which shows that the statement holds for the first iteration.

Inductive step: Assume the statement holds for the $(k-1)$th iteration, i.e., $\forall u \in I_{k-1},\ \Delta(u) = \delta - (k-1)$. Then, we show that the statement holds for the kth iteration. Since $\mathbb{C}(u) \subseteq \mathcal{C}(u) \subseteq I_{k-1}$ and $\forall v \in I_{k-1},\ \Delta(v) = \delta - (k-1)$, $\forall u \in I_k$ we have that

$$\begin{aligned}\Delta(u) &= \min\left(\min_{v \in \mathbb{C}(u)} (\Delta(v) - 1), \Delta(u)\right) \\ &= \delta - k.\end{aligned} \tag{4}$$

Thus, Lemma 2 holds for the kth iteration.

As we have demonstrated that both basis and inductive step hold, by mathematical induction, we can conclude that the statement holds. ■

Theorem 1: In the special settings, the cover problem in the kth iteration is a DUDC problem.

Proof: First of all, we have $R = r$ in the special settings. Thus, the communication range of a node v, either an RN or an SN, can be represented by a disk whose center is v and radius is r. We denote this disk as $D(v)$. Then, we prove that if a node v can feasibly cover an arbitrary node in I_{k-1}, then v can feasibly cover all the nodes (in I_{k-1}) covered by $D(v)$, i.e., the nodes in $\mathcal{N}(v) \bigcap I_{k-1}$. Otherwise, v cannot feasibly cover any node in I_{k-1} except itself.

Let $u \in \mathcal{N}(v) \bigcap I_{k-1}$. When u is feasibly covered by v, we have that

$$\mathcal{H}(\mathcal{S}(z, v)) \le \Delta(u) - 1. \tag{5}$$

According to Lemma 2, we know that

$$\forall q \in \mathcal{N}(v) \bigcap I_{k-1},\ \Delta(q) = \delta - (k-1). \tag{6}$$

Combining (5) and (6), we get that

$$\begin{aligned}\mathcal{H}(\mathcal{S}(z, v)) &\le \Delta(u) - 1 = (\delta - (k-1)) - 1 \\ &= \delta - k,\end{aligned} \tag{7}$$

which implies that if v can feasibly cover an arbitrary node (in I_{k-1}) covered by $D(v)$, then v can feasibly cover all the nodes (in I_{k-1}) covered by $D(v)$.

In the same way, when u cannot be feasibly covered by v, we have that

$$\begin{aligned}\mathcal{H}(\mathcal{S}(z, v)) &> \Delta(u) - 1 = (\delta - (k-1)) - 1 \\ &= \delta - k,\end{aligned} \tag{8}$$

which indicates that if v cannot feasibly cover an arbitrary node (in I_{k-1}) covered by $D(v)$, then v cannot feasibly cover any nodes in I_{k-1}.

Therefore, in the kth iteration, as $\mathcal{E}$ has been searched, each node v in $\mathcal{E}$ can be viewed as a disk $D(v)$ and all disks have the same radius since $r = R$. Then the cover problem in the kth iteration is to seek a minimum subset $\bar{\mathcal{E}}$ of $\mathcal{E}$ such that each node in I_{k-1} is covered by at least one unit disk, which is mathematically the DUDC problem. ■

Therefore, algorithms for the DUDC problem can be employed in the special settings, and these algorithms are rich in the literature. In this paper, a polynomial-time algorithm [9] with an approximation ratio of 22 is adopted.

Another problem may arise when we solve the deployment problem in each iteration. This problem is how to select an appropriate node when different nodes feasibly cover the same set of nodes. To deal with this problem, each node u is given a weight as

$$\omega(u) = \mathcal{H}(\mathcal{S}(z, u)). \tag{9}$$

The smaller the weight of a node is, the fewer RNs may be placed to connect this node to the sink. As a result, when the above problem arises, the node with the least weight is selected.

Furthermore, to save RNs deployed in the second step, the third step of CRNP is designed as shown in Algorithm 3, where the weight of an deployed RN u is defined as $|\mathcal{N}(u)|$.

B. Algorithm Analysis

1) Time Complexity: Let $N = |X| + |Y| + 1$. The time complexity of the first step is O($N \lg N$) since two shortest path trees are formed in this step and the shortest path tree algorithm has a time complexity of O($N \lg N$) [8].

Next, we analyze the time complexity of the second step. Obvoiusly, the time complexities of the first loop (lines 2-3) and the second loop (lines 4-6) are O($N \lg N$) and O(N) [5], respectively. In the main loop of the second step, the time complexities of the two inner loops are O(N^2) and O(N), respectively. The time complexity of line 14 is O(N). The complexity of line 10 is O(N^3) [8] in the general settings, and O(N^6) [9] in the special settings, since different algorithms are selected to deal with the set cover problem and the DUDC problem. The maximal number of iterations is a constant $\Delta_{\max}$ ($\Delta_{\max} = \max_{x \in X} \Delta(x)$) since the hop count of the longest feasible path cannot be larger than $\Delta_{\max}$. Therefore, the second step has a time complexity of O(N^3) in the general case, and O(N^6) in the special case.

At the third step, each iteration will apply the shortest path tree algorithm to check whether the RN deployed in the second step can be deleted. Thus, the time complexity of the third step is O($N^2 \lg N$) since the number of RNs deployed at the second step cannot be larger than N.

As the time complexity of CRNP is the sum of its three steps, we know that the time complexity of CRNP is O(N^3) in the general settings, and O(N^6) in the special settings.

2) Approximation Ratio: We first analyze the approximation ratio of CRNP in the general settings. Let OPT be the set of nodes on an optimal feasible tree and APT be the set of nodes on the feasible tree returned by CRNP. Then, the approximation ratio is given by

$$\begin{aligned}\mathcal{R}_{CRNP} &= \frac{|APT - X - \{z\}|}{|OPT - X - \{z\}|} = \frac{\left|\bigcup_{k=0}^{l} I_k - X - \{z\}\right|}{|OPT - X - \{z\}|} \\ &\leq \frac{\left|\bigcup_{k=0}^{l} I_k\right|}{|OPT - X - \{z\}|},\end{aligned} \tag{10}$$

where I_k is the set of RNs deployed in the kth iteration and l is the number of iterations.

As X, Y and $\{z\}$ are pairwise disjoint, we can further get that

$$\begin{aligned}\mathcal{R}_{CRNP} &\leq \frac{\left|\bigcup_{k=0}^{l} I_k\right|}{|OPT| - |X| - |\{z\}|} = \frac{\left|\bigcup_{k=0}^{l} I_k\right|}{|OPT| - |X| - 1} \\ &< \frac{\left|\bigcup_{k=0}^{l} I_k\right| + |X| + 1}{|OPT|} = \frac{\left|\bigcup_{k=0}^{l} I_k\right|}{|OPT|} + \frac{|X| + 1}{|OPT|}.\end{aligned} \tag{11}$$

Because $|OPT - X - \{z\}| \geq 0$ (which is straightforwardly followed by $|OPT| \geq |X| + 1$) and $I_0 = X$, inequality (11) can be further bounded by

$$\begin{aligned}\mathcal{R}_{CRNP} &< \frac{\left|\bigcup_{k=0}^{l} I_k\right|}{|OPT|} + 1 \leq \frac{\sum_{k=0}^{l} |I_k|}{|OPT|} + 1 \\ &= \frac{|I_0|}{|OPT|} + \sum_{k=1}^{l} \left(\frac{|I_k|}{|OPT|}\right) + 1 \\ &\leq \sum_{k=1}^{l} \left(\frac{|I_k|}{|OPT_k|} \frac{|OPT_k|}{|OPT|}\right) + 2,\end{aligned} \tag{12}$$

where OPT_k is a minimum set cover for the kth iteration. As GSC is employed to solve the set covering problem in each iteration in the general settings, the approximation ratio of GSC is given by

$$\begin{aligned}\forall k \in \{1, 2, ..., l\},\ \frac{|I_k|}{|OPT_k|} &\leq \ln |I_{k-1}| + 1 \\ &\leq \ln |I_0| + 1 = \ln |X| + 1.\end{aligned} \tag{13}$$

Due to the fact that $|OPT_k| \leq |X|$ and $|OPT| \geq |X| + 1$, combining inequalities (12)-(13), we can conclude that

$$\begin{aligned}\mathcal{R}_{CRNP} &< \sum_{k=1}^{l} \left(\frac{|I_k|}{|OPT_k|} \frac{|OPT_k|}{|OPT|}\right) + 2 \\ &\leq \sum_{k=1}^{l} \left((\ln |X| + 1) \left(\frac{|X|}{|X| + 1}\right)\right) + 2 \\ &\leq l (\ln |X| + 1) + 2 \\ &\leq \Delta_{\max} \ln |X| + \Delta_{\max} + 2.\end{aligned} \tag{14}$$

Because the hop constraints are predetermined constants, we can conclude that CRNP is an algorithm with an approximation ratio of O($\ln n$), where $n = |X|$.

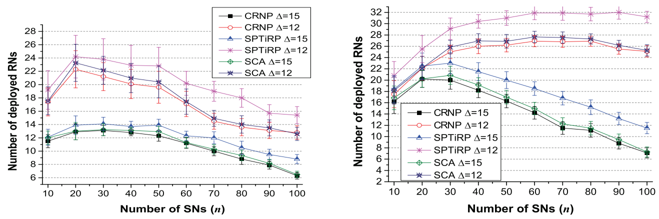
Next, we prove that CRNP is an O(1)-approximation algorithm in the special settings. As shown in Theorem 1, the cover problem in each iteration is actually the DUDC problem in the special settings. As a result, the algorithm [9] with a constant approximation ratio of 22 can be employed to solve the DUDC problem in each iteration. Therefore, we have that

$$\forall k \in \{1, 2, ..., l\}, \frac{|I_k|}{|\overline{OPT}_k|} \leq 22, \tag{15}$$

where $\overline{OPT}_k$ denotes a minimum disk cover for the kth iteration.

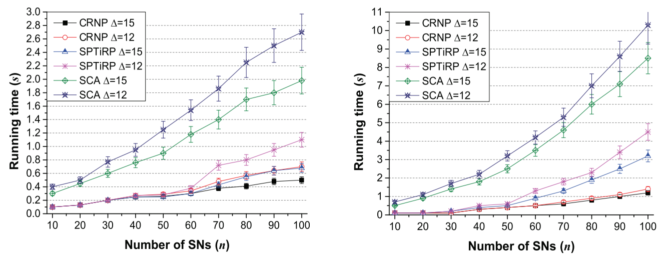
Similarly, as $|\overline{OPT}_k| \leq |X|$ and $|OPT| \geq |X| + 2$, according to inequalities (12) and (15), we can get that

$$\begin{aligned}\mathcal{R}_{CRNP} &< \sum_{k=1}^{l} \left(\frac{|I_k|}{|\overline{OPT}_k|} \frac{|\overline{OPT}_k|}{|OPT|}\right) + 2 \\ &\leq \sum_{k=1}^{l} \left(22 \left(\frac{|X|}{|X| + 2}\right)\right) + 1 \leq 22l + 2 \\ &\leq 22\delta + 2.\end{aligned} \tag{16}$$



(a) $r = 65$, $R = 115$. (b) $r = R = 65$.

Fig. 2. The comparisons between SPTiRP and CRNP on the deployed RNs.



(a) $r = 65$, $R = 115$. (b) $r = R = 65$.

Fig. 3. The simulation results of running time.

Correspondingly, we can conclude that we have an algorithm with an approximation ratio of O(1) in the special settings since δ is a constant.

IV. Simulations

In this section, extensive simulations are performed on a computer equipped with a 2.4 GHz Intel Core i5-2430M CPU and a 2 GB RAM. In these simulations, SNs are randomly placed on a square field with the side length of 600 meters, and the number n of SNs varies from 10 to 100. To ensure that the HCRNP problem has a feasible solution, in each simulation run, we place 400 randomly distributed CDLs in the deployment field. Simulations are carried out under two scenarios: $r = R = 65$ m; $r = 65$ m and $R = 115$ m. Without loss of generality, all SNs have the same hop constraint Δ, which is set as 15 or 12 in simulations. In addition, to perform a fair comparison, 50 simulations are implemented to obtain each piece of data in the simulation figures based on the method of batch means for the confidence level of 95%.

A. Deployment Cost and Running Time

Deployment cost is measured in terms of the number of deployed RNs. The latest algorithms-SPTiRP [5] and SCA [7] for the HCRNP problem are used as the baseline to evaluate the performance of CRNP, and the comparison results are given in Fig. 2. We can observe that the CRNP algorithm significantly outperforms SPTiRP in each simulation figure, and the largest number of RNs saved by CRNP in comparison to SPTiRP (as shown in Fig. 2(a)-2(b)) is 4.52 $(4.52/18.00 \approx 25.11\%)$, and 6.51 $(6.51/32.13 \approx 20.28\%)$, respectively. CRNP also has a obvious performance gain over SCA especially when the number of SNs is small (i.e., [10, 60]).

The simulation results of running time are shown in Fig. 3, which shows that CRNP has the shortest running time among these three algorithms. This is mainly due to the fact that the from-SNs-to-sink deployment method significantly reduces the time for finding the SNs that can be feasibly covered by a CDL. Specifically, in each iteration of CRNP, to search the SNs can be feasibly covered by a CDL, we only need to know the smallest hop count from this CDL to the sink. Therefore, a shortest path tree rooted at the sink and connecting all CDLs is enough to know the smallest hop count from each CDL to the sink. In contrast, SCA has the longest running time since SCA employs a from-sink-to-SNs deployment method. To find the SNs can be feasibly covered by a CDL, SCA needs to build a shortest path tree rooted at this CDL and connecting all SNs. Therefore, SCA has to build such a shortest path tree for each CDL, which is much more time-consuming than CRNP.

V. Conclusion

In this paper, we have studied the HCRNP problem in WSNs. To address this problem, we have presented a polynomial-time algorithm-CRNP, whose approximation ratio is $O(\ln n)$ in arbitrary settings and O(1) in special settings. Moreover, extensive simulations have been carried out to verify the effectiveness and evaluate the performance of proposed the algorithms.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (2017YFE0101300), the Natural Science Foundation of China under grant 61673371 and 71661147005, the Youth Innovation Promotion Association, CAS (2015157), and the key scientific research of universities in Henan China (15B520041, 18B520045).

References

[1] A. A. Kumer S., K. Øvsthus, and L. M. Kristensen, “An Industrial Perspective on Wireless Sensor Networks-A Survey of Requirements, Protocols, and Challeges,” *IEEE Commun. Survey Tuts.,* vol. 16, no. 3, pp. 1391-1412, Jan. 2014.

[2] S. Misra, S. Hong, G. Xue, and J. Tang, “Constrained Relay Node Placement in Wireless Sesnor Networks: Formulation and Approximations,” *IEEE/ACM Trans. Netw.,* vol. 18, no. 2, pp. 434-447, Apr. 2010.

[3] L. Gouveia, “Multicommodity Flow Models for Spanning Trees with Hop Constraints,” *European J. of Oper. Res.,* vol. 95, no. 1, pp. 178-190, Nov. 1996.

[4] A. Bhattacharya, and A. Kumar, “Delay Constrained Optimal Relay Placement for Planned Wireless Sensor Networks,” *Proc. IEEE I-WQoS’10,* pp. 1-9, 2010.

[5] A. Bhattacharya and A. Kumar, “A Shortest Path Tree Based Algorithm for Relay Placement in A Wireless Sensor Network and Its Performance Analysis,” *Comput. Netw.,* vol. 71, no. 4, pp. 48-62, 2014.

[6] A. Nigam, and Y.K. Agarwal, “Optimal Relay Node placement in Delay Constrained Wireless Sensor Network Design,” *European J. of Oper. Res.,* vol. 233, no. 1, pp-220-233, 2014.

[7] C. Ma, W. Liang, and M. Zheng, “Delay constrained relay node placement in two-tiered wireless sensor networks: A set-covering-based algorithm,” *Journal of Network and Computer Applications,* vol. 93, no. 1, pp. 76-90, 2017.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to Algorithm,” *The MIT Press and McGraw-Hill,* pp. 1117-1122, 2009.

[9] F. Claude, R. Dorrigiv, S. Durocher, and R. Fraser, “Practical Discrete Unit Disk Cover Using an Exact Line-Seperable Algorithm,” *in Proc. 20th Int’l Symp. Algorithms and Computation (ISAAC’09),* pp. 45-54, 2009.