

Barrier Coverage by Sensors with Adjustable Ranges

HAOSHENG FAN, MINMING LI, and XIANWEI SUN, City University of Hong Kong
PENG-JUN WAN, Illinois Institute of Technology
YINGCHAO ZHAO, Caritas Institute of Higher Education

One of the most fundamental tasks of wireless sensor networks is to provide coverage of the deployment region. We study the coverage of a line interval with a set of wireless sensors with adjustable coverage ranges. Each coverage range of a sensor is an interval centered at that sensor whose length is decided by the power the sensor chooses. The objective is to find a range assignment with the minimum cost. There are two variants of the optimization problem. In the *discrete* variant, each sensor can only choose from a finite set of powers, whereas in the *continuous* variant, each sensor can choose power from a given interval. For the discrete variant of the problem, a polynomial-time exact algorithm is designed. For the continuous variant of the problem, NP-hardness of the problem is proved and followed by an ILP formulation. Then, constant-approximation algorithms are designed when the cost for all sensors is proportional to r^κ for some constant $\kappa \geq 1$, where r is the covering radius corresponding to the chosen power. Specifically, if $\kappa = 1$, we give a 1.25-approximation algorithm and a fully polynomial-time approximation scheme; if $\kappa > 1$, we give a 2-approximation algorithm. We also show that the approximation analyses are tight.

Categories and Subject Descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Wireless sensor networks, barrier coverage, approximation algorithms

ACM Reference Format:

H. Fan, M. Li, X. Sun, P.-J. Wan, and Y. Zhao. 2014. Barrier coverage by sensors with adjustable ranges. *ACM Trans. Sensor Netw.* 11, 1, Article 14 (July 2014), 20 pages.

DOI: <http://dx.doi.org/10.1145/2629518>

1. INTRODUCTION

One of the most fundamental tasks of wireless sensor networks is to provide coverage of the deployment region. In applications like highway monitoring, tunnel monitoring, and seaway monitoring, the deployment region can be abstracted as a line interval. In these monitoring tasks, although the sensors can be predeployed according to some pattern to optimize certain performance, some sensors may malfunction due to various reasons. Hence, at a certain moment, the distances between sensors do not follow any pattern anymore, but the positions of workable sensors are still known to us. We aim

This work was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 117913).

A preliminary version of this article appeared in Li et al. 2011.

Authors' addresses: H. Fan and M. Li, Department of Computer Science, City University of Hong Kong; emails: haoshefan2-c@my.cityu.edu.hk, minming.li@cityu.edu.hk; X. Sun, Wuhan University of Technology, China; email: xianweisun0528@gmail.com; P.-J. Wan, Department of Computer Science, Illinois Institute of Technology; email: wan@cs.iit.edu; Y. Zhao, Department of Computer Science, Caritas Institute of Higher Education, Hong Kong; email: yczhao@cihe.edu.hk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2014 ACM 1550-4859/2014/07-ART14 \$15.00

DOI: <http://dx.doi.org/10.1145/2629518>

to design coverage solutions that only use those workable sensors. Furthermore, the unanticipated malfunction of sensors makes the solutions for instant energy minimization more useful than maximizing coverage lifetime.

In this article, we study the coverage of a line interval I by a set V of wireless sensors with adjustable coverage ranges. Each sensor v has a set $R(v)$ of possible coverage radii. For each $r \in R(v)$, the coverage range of v is the interval $C(v, r)$ of length $2r$ centered at v , and its associated cost is $c(v, r)$. The cost is proportional to r^κ , where κ is the *path-loss exponent* (a constant between 2 and 5 depending on the wireless environment). According to different targeted monitoring quality, the cost can be adjusted further. For example, the mean power assignment used by Wan et al. [2012] can cut the exponent κ in half, resulting in a range of [1, 2.5]. Therefore, it is worthwhile exploring the coverage problem with $\kappa \in [1, 2.5]$. For simplicity of treatment, we assume that for each sensor v , $0 \in R(v)$ and $c(v, 0) = 0$. The objective is to choose a coverage radius $r(v) \in R(v)$ for each sensor $v \in V$ such that

$$I \subseteq \bigcup_{v \in V} C(v, r(v))$$

and $\sum_{v \in V} c(v, r(v))$ is minimized. This optimization problem is referred to as min-cost linear coverage (MCLC).

We consider two variants of MCLC. In the *discrete* variant, $R(v)$ is finite for each sensor $v \in V$; in the *continuous* variant, $R(v)$ is a finite interval for each sensor $v \in V$. For the discrete variant of MCLC, we present a polynomial-time exact algorithm to compute the optimal solution. For the continuous variant of MCLC, we develop constant-approximation algorithms when the cost $c(v, r)$ for all sensors $v \in V$ is proportional to r^κ for some constant $\kappa \geq 1$. Specifically, if $\kappa = 1$, we give a 1.25-approximation algorithm and a fully polynomial-time approximation scheme (FPTAS); if $\kappa > 1$, we give a simple 2-approximation algorithm.

Related works. Most prior works on wireless sensor coverage assume that each sensor has a fixed circular coverage. For this setting, the coverage of the deployment region can be reduced to the coverage of a finite number of representative points referred to as targets. The problem min-weight disk cover seeks a subset of given disks with the smallest cost to cover the targets. For the special case in which all disks have the same radii normalized to one, the problem min-weight disk cover is known as min-weight unit-disk cover. A series of successively improved approximation algorithms for the unweighted variant of min-weight unit-disk cover were developed in Brönnimann and Goodrich [1995], Călinescu et al. [2004], Narayanappa and Vojtechovský [2006], and Carmi et al. [2007], and the best one among them has an approximation ratio of 38 [Carmi et al. 2007], where the points to cover are given as input. Subsequently, a series of successively improved constant-approximation algorithms for min-weight unit-disk cover were developed in Ambuhl et al. [2006], Huang et al. [2009], Dai and Yu [2009], Zou et al. [2009], and Erlebach and Mihalák [2010]. Among them, the best one is a $(4 + \epsilon)$ -approximation algorithm (here, there is a unit disk centered at every input point), which was first presented in Zou et al. [2009] and rediscovered in Erlebach and Mihalák [2010]. Recently, Wan et al. [2011] developed a PTAS for the unweighted variant of min-weight disk cover, a randomized $2^{O(\log^* n)}$ -approximation for the weighted variant for the general min-weight disk cover, where $\log^* n$ is the iterated logarithm of n with base 2. Other variants of disk coverage problems have been studied in Abrams et al. [2004], Gonzalez [1991], Hochbaum and Maass [1985], Wang and Zhong [2006], and Yun et al. [2010]. In addition to these algorithmic studies, the probabilistic aspects of wireless coverage have been studied in Hou et al. [2009] and Wan and Yi [2006].

When the coverage target is a line interval, it is usually called *barrier coverage*. Along this direction, various works have been done. If sensors are allowed to move, Czyzowicz

et al. [2009] studied the objective of minimizing the maximum sensor movement to achieve the coverage. They proposed an $O(n^2)$ algorithm to compute the optimal movement of sensors when all the sensors have the same sensing range. Recently, Chen et al. [2013] improved the complexity to $O(n \log n)$ and also designed an $O(n^2 \log n)$ algorithm to compute the optimal movement for sensors with arbitrary ranges. Czyzowicz et al. [2010] considered covering a line interval with the goal of minimizing the sum of distances traveled by all the sensors. Mehrandish et al. [2011] considered the objective of minimizing the number of sensors that must move to cover the barrier. Bar-Noy and Baumer [2011] studied the lifetime maximization problem when sensors can adjust their ranges. Then, Bar-Noy et al. [2012] studied another lifetime maximization problem when all sensors can be activated only once. In the most recent paper by Bar-Noy et al. [2013], they introduced a parameter a to beautifully connect sensors' ability to move and adjust ranges, and obtained profound theoretical results to maximize the network lifetime.

In the following discussion, we study the problem of covering a line interval using immobile sensors. Suppose that n sensors are distributed on a line at positions $\{x_1, x_2, \dots, x_n\}$ ($x_i \leq x_{i+1}$ for $1 \leq i \leq n-1$). We want to assign each sensor a power so that each point in the whole line interval can be covered by at least one sensor; the objective is to minimize the total energy consumption of the sensors, which is the dual problem of the lifetime maximization problem studied by Bar-Noy and Baumer [2011]. We divide our discussion into two parts based on whether the different powers that a sensor can choose are either finite (as input) or arbitrary.

2. FINITE CHOICES OF POWERS

A sensor usually is equipped with an eight-bit register to memorize the power at which it is working. Therefore, it is reasonable to assume that each sensor can choose to work at one power selected from a set $\{P^1, P^2, \dots, P^k\}$. If a sensor chooses power P^j and locates at x_i , then it can cover the interval $[x_i - r^j, x_i + r^j]$. Denote the power chosen by sensor at x_i as P^{j_i} . We want to minimize $\sum_{i=1}^n P^{j_i}$ while guaranteeing that any point in the interval $[0, m]$ can be covered by at least one sensor. Here, m is an integer greater than 1.

To solve this problem, we construct a corresponding directed weighted graph $G = (V, E, c)$ as follows. The node set $V = \{v_{ij} | 1 \leq i \leq n, 1 \leq j \leq k\} \cup \{s, t\}$, where s represents position 0, t represents position m , and v_{ij} means that the sensor at position x_i chooses power P^j . Initially, let $E = \emptyset$. Add a directed edge $e = (s, v_{ij})$ into E if and only if $r^j \geq x_i$, and assign $c(e) = 0$; add a directed edge $e = (v_{ij}, t)$ into E if and only if $x_i + r^j \geq m$, and assign $c(e) = P^j$; add a directed edge $e = (v_{ij}, v_{ab})$ into E if and only if $x_a > x_i$ and $r^j + r^b \geq x_a - x_i$, and assign $c(e) = P^j$. We set $c(e)$ to be 0 for edges that are connected to s to avoid double counting the power of the left-most sensor. Now the power minimization problem is equivalent to finding a shortest path from s to t in graph G , because a path in G corresponds to one coverage solution for the interval $[0, m]$ and the weight summation along that path is the power consumed by that solution.

Since the graph G has $kn + 2$ nodes in total, which is polynomial in k and n , the shortest path algorithm can be done in polynomial time.

3. ARBITRARY CHOICE OF POWERS

When the sensors can choose to work at any power, then the energy minimization problem to maintain the coverage of the line becomes difficult. First, we introduce the problem statement as follows.

We put n sensors $\mu_1, \mu_2, \dots, \mu_n$ in the interval $[0, m]$ and suppose that x_1, x_2, \dots, x_n are their coordinates, respectively. Additionally, let $x_0 = 0$ and $x_{n+1} = m$. Each sensor

Table I. Notations and Meaning

Notations	Meaning
n	Number of sensors
μ_i	Sensor i
m	Right-most position of the interval I
x_i	Coordinate of sensor i
r_i	Radius of sensor i
\mathbf{r}_i	Range of sensor i
\mathbf{R}	Range assignment
$C(\mathbf{R}), R$	Cost of \mathbf{R}
κ	Power ratio (a positive constant)
\mathbf{R}^*	Optimal range assignment
R^*	Cost of the optimal assignment
\mathbf{r}_i^*	Range of sensor i in optimal solution (\mathbf{R}^*)
\mathbf{d}_i	i -th segment
d_i	Length of i -th segment (\mathbf{d}_i)
S_k	$\sum_{i=1}^{n+1} d_i^\kappa$
x_W	Coordinate of point W in the interval
S_1	The total length of all segments when $\kappa = 1$ ($\sum_{i=1}^{n+1} d_i$)
P	The left-most point of the line interval I .
Q	The right-most point of the line interval I .

μ_i has a transmission radius r_i . We define $\mathbf{r}_i = [x_i - r_i, x_i + r_i]$ as μ_i 's range. If $r_i > 0$, we say that μ_i is chosen and range \mathbf{r}_i is nontrivial. In the following, whenever we say "range," we mean nontrivial ranges. For any point x in $[0, m]$, we say that \mathbf{r}_i covers x if and only if $|x - x_i| \leq r_i$.

Define $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$ as a range assignment of $[0, m]$ if $\forall x \in [0, m]$, there exists an \mathbf{r}_i that covers x . The cost of \mathbf{R} is defined to be $C(\mathbf{R}) = \sum_{i=1}^n r_i^\kappa$, where κ is a positive constant. We also use R to represent $C(\mathbf{R})$. If a range assignment has the minimal cost among all possible assignments, then we say that this assignment is the optimal assignment and denote it by $\mathbf{R}^* = (\mathbf{r}_1^*, \mathbf{r}_2^*, \dots, \mathbf{r}_n^*)$.

We say that $\mathbf{d}_i = (x_{i-1}, x_i)$ is the i -th segment for $i = 1, 2, \dots, n+1$ and use d_i to denote $x_i - x_{i-1}$. If some \mathbf{r}_i covers every point in a segment \mathbf{d}_j ($\forall j \in \{1, 2, \dots, n+1\}$), we say that \mathbf{r}_i covers \mathbf{d}_j . We call that \mathbf{r}_i partly covers \mathbf{d}_j if \mathbf{r}_i covers partial points in segment \mathbf{d}_j .

Let $S_\kappa = \sum_{i=1}^{n+1} d_i^\kappa$. In this article, we use x_W to denote the coordinate of any point W in $[0, m]$.

3.1. $\kappa = 1$

In this section, we study the problem when $\kappa = 1$. We first prove the NP-hardness of the problem. Then, we give an ILP formulation of the problem and discuss possible approximations.

3.1.1. NP-Hardness. We make a reduction from the Partition problem to our problem. Define $S(A) = \sum_{a \in A} a$.

Partition problem: Given a set of integers $I = \{a_1, a_2, \dots, a_n\}$, the objective is to find a subset $A \subseteq I$ so that $\sum_{a_i \in A} a_i = \frac{1}{2} \sum_{i=1}^n a_i$.

Given a Partition problem with the preceding specified input, we construct an instance of our problem as follows. Define $\epsilon_i = \frac{a_i}{(n+1)\max a_j}$ and $\theta = 3$. Let $I_2 = \{\theta, \theta^2, \dots, \theta^n, \theta - \epsilon_1, \theta^2 - \epsilon_2, \dots, \theta^n - \epsilon_n\}$. We put $2n$ sensors on the line with $d_i = \theta^i$



Fig. 1. Transformed instance. There are $2n$ sensors on the line with $d_i = \theta^i$ if $i \leq n$ and $d_i = \theta^{2n+1-i} - \epsilon_{2n+1-i}$ if $n < i \leq 2n$.

if $i \leq n$ and $d_i = \theta^{2n+1-i} - \epsilon_{2n+1-i}$ if $n < i \leq 2n$, as shown in Figure 1. Let m be at the position of the right-most sensor, which implies that $m = S(I_2)$.

First, we will show that dividing I_2 into two disjoint sets B and $I_2 - B$ with equal sum directly corresponds to a solution for the Partition problem and vice versa. To show this, it is sufficient to prove that if $S(B) = S(I_2 - B) = \frac{1}{2}S(I_2)$, then exactly one element in the pair $(\theta^i, \theta^i - \epsilon_i)$ is in B and the other element is in $I_2 - B$ (because that will force θ^j to appear exactly once in both $S(B)$ and $S(I_2 - B)$, which will then imply that the summation of ϵ 's in both B and $I_2 - B$ be the same, leading to a corresponding solution for the Partition problem). Suppose on the contrary that it is not true, then define i_{max} to be the largest i , where both θ^i and $\theta^i - \epsilon_i$ are in B . Without loss of generality, assume that for every $i_{max} < j \leq n$, exactly one element in the pair $(\theta^j, \theta^j - \epsilon_j)$ is in B . Then,

$$S(B) \geq \sum_{j=i_{max}}^n (\theta^j - \epsilon_j) + \theta^{i_{max}}.$$

Since

$$\theta^{i_{max}} > \sum_{j=1}^{i_{max}-1} \theta^j + 1$$

and

$$\sum_{j=1}^n \epsilon_j < 1,$$

we have

$$S(B) > \sum_{j=1}^n \theta^j > \frac{1}{2}S(I_2),$$

which is a contradiction.

Then, we will show that I_2 can be partitioned into two disjoint equal-sum sets if and only if our problem has a solution with cost $m/2$.

For the if direction, since the total cost is $m/2$, we know that there is no overlap between any two chosen ranges. Suppose that $\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_k}$ are sensors chosen from left to right in the optimal assignment. We have

$$\begin{aligned} r_{i_1} &= d_1 + \dots + d_{i_1}; \\ r_{i_1} + r_{i_2} &= d_{i_1+1} + \dots + d_{i_2}; \\ &\dots \\ r_{i_{k-1}} + r_{i_k} &= d_{i_{k-1}+1} + \dots + d_{i_k}; \\ r_{i_k} &= d_{i_k+1} + \dots + d_{2n}. \end{aligned}$$

The d_j 's appearing in alternative lines exactly add up to $\sum_{j=1}^k r_{i_j} = m/2$. This maps to a solution for partitioning I_2 .

For the only if direction, for all the items selected into B , we combine those corresponding segments into sup-segments according to their adjacency relationship as shown in Figure 2.

We also use D_1, D_2, \dots, D_k to denote the lengths of these sup-segments if the context is clear. Without loss of generality, we assume that D_1 does not touch point 0. For items

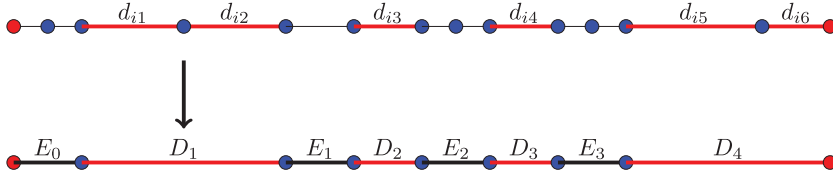


Fig. 2. Sup-segments. We merge successive subsegments that belong to the same set into sup-segments E_i and D_i . E_i and D_i will also be used to represent the lengths of the corresponding sup-segments if the context is clear.

not selected into B , we can do similar merging to get sup-segments E_0, E_1, \dots, E_k , where E_k is optional depending on the configuration. We also use E_0, E_1, \dots, E_k to represent the lengths of the corresponding sup-segments if the context is clear. Two groups of sup-segments alternate on the line as shown in Figure 2. We have

$$\sum_{i=1}^k D_i = \sum_{i=0}^k E_i = m/2.$$

Notice that the end points of D_i are all sensors. Define r_{u_i} and r_{v_i} to be the ranges of left and right end points of D_i for each $i = 1, 2, \dots, k$.

Let $i_0 = \operatorname{argmax}_{1 \leq i \leq k} D_i$, $j_0 = \operatorname{argmax}_{0 \leq j \leq k} E_j$.

In our transformed instance, θ^n and $\theta^n - \epsilon_n$ are two adjacent longest segments that are longer than the summation of all elements on their left or on their right. Further, by the previous discussion that one element in pair $(\theta^n, \theta^n - \epsilon_n)$ is in B and the other one is in $I_2 - B$, we know that to ensure $\sum_{i=1}^k D_i = \sum_{i=0}^k E_i = m/2$, one element in pair $(\theta^n, \theta^n - \epsilon_n)$ must be in D_{i_0} and the other one must be in E_{j_0} . Therefore, we know that either $j_0 = i_0 - 1$ or $j_0 = i_0$.

Case 1: $j_0 = i_0 - 1$

Let $r_{u_1} = E_0$ and $r_{v_1} = D_1 - r_{u_1}$.

For each $2 \leq i < i_0$, let $r_{u_i} = E_{i-1} - r_{v_{i-1}}$ and $r_{v_i} = D_i - r_{u_i}$.

Let $r_{v_k} = E_k$ and $r_{u_k} = D_k - r_{v_k}$.

For each $i_0 \leq i < k$, let $r_{v_i} = E_i - r_{u_{i+1}}$ and $r_{u_i} = D_i - r_{v_i}$.

Notice that according to the length assignment, for each $i < i_0$, we have

$$D_i > \sum_{l=1}^{i-1} (D_l + E_l) + E_0,$$

and for each $i \geq i_0$, we have

$$D_i > \sum_{l=i+1}^k (D_l + E_l).$$

On the other hand, for each $j \leq j_0$, we have

$$E_j > \sum_{l=1}^{j-1} (D_l + E_l),$$

and for each $j > j_0$, we have

$$E_j > \sum_{l=j+1}^k (D_l + E_l).$$

Therefore, for each i , we have $r_{u_i} > 0, r_{v_i} > 0$.

From these equalities, we have

$$r_{v_{i_0-1}} + r_{u_{i_0}} - E_{i_0-1} = \sum_{i=1}^k D_i - \sum_{i=1}^k E_i = 0,$$

which means that the sup-segment E_{j_0} is totally covered like all of the other sup-segments.

Therefore, $\mathbf{R} = (\mathbf{r}_{u_1}, \mathbf{r}_{v_1}, \mathbf{r}_{u_2}, \mathbf{r}_{v_2}, \dots, \mathbf{r}_{u_k}, \mathbf{r}_{v_k})$ covers each point in $[0, m]$, and the cost

$$R = \sum_{i=1}^k (r_{u_i} + r_{v_i}) = m/2.$$

Thus, \mathbf{R} is the optimal solution.

Case 2: $j_0 = i_0$

The proof of this case is similar to Case 1.

3.1.2. ILP Formulation. We put two virtual sensors, μ_0 and μ_{n+1} , on 0 and m , which are chosen by default and have $r_0 = r_{n+1} = 0$ in any assignment. We use y_i to represent whether sensor μ_i is chosen in the assignment ($y_i = 1$ if μ_i is chosen and $y_i = 0$ otherwise). Let $y_0 = y_{n+1} = 1$. We use f_{ij} to represent the adjacency relationship of the chosen sensors: $f_{ij} = 1$ if μ_j is chosen, and μ_i is the closest sensor chosen in the assignment on the left of μ_j and $f_{ij} = 0$ otherwise.

Suppose that in the optimal assignment, we choose k sensors from $\mu_1, \mu_2, \dots, \mu_n$, then we have $k + 1$ of the f_{ij} 's being 1, which gives the following constraint:

$$\sum_{0 \leq i < j \leq n+1} f_{ij} = \sum_{i=1}^n y_i + 1. \quad (1)$$

For any μ_i ($1 \leq i \leq n$), on its left and right, there is one closest chosen sensor, respectively, which gives us the following constraints:

$$\begin{aligned} \sum_{0 \leq t < i} f_{ti} &= y_i \\ \sum_{i < s \leq n+1} f_{is} &= y_i. \end{aligned} \quad (2)$$

To cover the whole interval $[0, m]$, we should have $r_i + r_j \geq x_j - x_i$ if $f_{ij} = 1$, which gives us the following constraint:

$$r_i + r_j \geq x_j - x_i + (f_{ij} - 1)m. \quad (3)$$

Finally, we have additional constraints as follows:

$$\begin{cases} y_0 = 1 \\ y_{n+1} = 1 \\ y_i = 0 \text{ or } 1 \text{ for } 1 \leq i \leq n \\ r_0 = 0 \\ r_{n+1} = 0 \\ r_i \geq 0 \text{ for } 1 \leq i \leq n. \end{cases} \quad (4)$$

The objective of the ILP is to minimize $\sum_{1 \leq i \leq n} r_i$.

Next we prove some lemmas about the relation between \mathbf{R}^* and segments.

LEMMA 3.1. *In the optimal range assignment $\mathbf{R}^* = (\mathbf{r}_1^*, \mathbf{r}_2^*, \dots, \mathbf{r}_n^*)$, there are at most two ranges that partly cover $\mathbf{d}_i \forall i = 1, 2, \dots, n + 1$.*

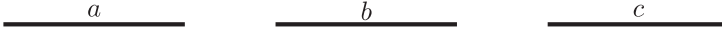


Fig. 3. The relative position of a, b, and c.

PROOF. We prove this lemma by contradiction. Suppose that there are more than two ranges that partly cover \mathbf{d}_i in the optimal assignment. Then, among those sensors whose ranges partly cover \mathbf{d}_i , there are at least two sensors on the same side of this segment. Without loss of generality, we assume that μ_A and μ_B are on the left side of \mathbf{d}_i and $x_A < x_B$.

If $x_A + r_A \geq x_B + r_B$, then we can remove sensor μ_B , which contradicts the assumption that this assignment is optimal.

If $x_A + r_A < x_B + r_B$, then we can get a new assignment with $r'_A = r_B + (x_B - x_A)$ and $r'_B = 0$. It is easy to see that sensor μ_A with new range r'_A can cover what the original μ_A and μ_B cover. Since $r_A > x_B - x_A$, we get $r'_A + r'_B = r_B + (x_B - x_A) < r_B + r_A$, which contradicts our assumption that this assignment is optimal.

Therefore, for each segment \mathbf{d}_i , there are at most two ranges that partially cover it. \square

LEMMA 3.2. $\forall i \in \{1, 2, \dots, n+1\}$, $R^* \geq d_i$, and especially, if there exists some $d_i \geq S_1/2$, then we have $R^* = d_i$.

PROOF. By Lemma 3.1, there are at most two ranges that partly cover \mathbf{d}_i in \mathbf{R}^* . Suppose that \mathbf{r}_{i1} and \mathbf{r}_{i2} partly cover \mathbf{d}_i . Let \mathbf{r}_{i1} cover a length L_1 in \mathbf{d}_i ($r_{i1} \geq L_1$), and \mathbf{r}_{i2} must cover a length $L_2 = d_i - L_1$ in \mathbf{d}_i ($r_{i2} \geq L_2$). Therefore,

$$R^* \geq r_{i1} + r_{i2} \geq L_1 + L_2 = d_i.$$

If there exists some $d_i \geq S_1/2$, let $r_{i-1} = x_{i-1}$, $r_i = d_i - x_{i-1}$, $r_1 = r_2 = \dots = r_{i-2} = r_{i+1} = \dots = r_n = 0$, then $(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$ is an assignment with total cost d_i . Hence, $R^* \leq d_i$. On the other hand, we already proved that $R^* \geq d_i$. Therefore, we have $R^* = d_i$. \square

LEMMA 3.3. Let $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \{\mathbf{d}_i, i = 1, 2, \dots, n+1\}$ (their locations are shown in Figure 3), we use a, b, c to denote their lengths. Sort a, b, c to an ordered sequence $X_3 \leq X_2 \leq X_1$, then we have $R^* \geq \max(X_1, X_2 + X_3)$.

PROOF. If $X_1 \geq X_2 + X_3$, then we have $R \geq X_1 = \max(X_2 + X_3, X_1)$ by Lemma 3.2. If $X_1 < X_3 + X_2$, since there are at most two ranges that cover \mathbf{a} in \mathbf{R}^* by Lemma 3.1, we suppose that \mathbf{r}_{i1} and \mathbf{r}_{i2} cover \mathbf{a} from left and right, respectively. By symmetry, we assume that $r_{i1} > 0$. Then, we need to discuss two cases as follows:

Case 1: $r_{i2} = 0$.

Case 2: $r_{i2} > 0$.

To prove the lemma, it is sufficient to prove that $R^* \geq a+b$ or $R^* \geq b+c$ or $R^* \geq a+c$.

For Case 1, if \mathbf{r}_{i1} also partly covers \mathbf{c} , then \mathbf{r}_{i1} covers \mathbf{a} and \mathbf{b} , so

$$R^* \geq r_{i1} \geq a + b.$$

Otherwise (\mathbf{r}_{i1} does not cover any point in \mathbf{c}), we need $r_{i3} \geq c$ to cover \mathbf{c} (\mathbf{r}_{i3} is different from \mathbf{r}_{i1} and \mathbf{r}_{i2}) or two other ranges with the summation of radius larger than c to cover \mathbf{c} , then

$$R^* \geq r_{i1} + c \geq a + c.$$

For Case 2, $r_{i1} + r_{i2} \geq a$, if \mathbf{r}_{i2} does not cover any point in \mathbf{c} , we need $r_{i3} \geq c$ to cover \mathbf{c} (\mathbf{r}_{i3} is different from \mathbf{r}_{i1} and \mathbf{r}_{i2}), then

$$R^* \geq r_{i1} + r_{i2} + r_{i3} \geq a + c.$$

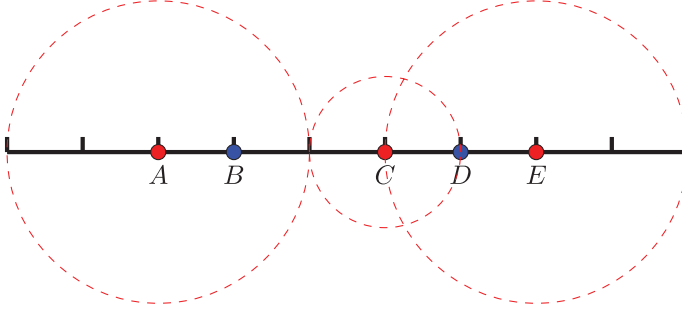


Fig. 4. R_l process illustration. Sensors are chosen from left to right. First, we choose the left-most sensor, which is A . The dashed circle centered at A is the area covered by A . Then, we choose the left-most sensor that is out of A 's range, which is C . Next, we choose sensor E following the same process. After choosing E , we extend the default radius that covers the remaining part on the left to a radius that can cover the right-most point of the interval and then finish the whole process.

Otherwise (r_{i_2} also partly covers c) r_{i_2} cover b , if r_{i_2} covers b from left, let r_{i_2} and r_{i_3} cover b and c (r_{i_2} is different from r_{i_3}), then

$$R^* \geq r_{i_2} + r_{i_3} \geq b + c.$$

Otherwise, if r_{i_2} covers b from right, then r_{i_1} and r_{i_2} cover a and b , which implies

$$R^* \geq r_{i_1} + r_{i_2} \geq a + b. \quad \square$$

We now design an algorithm to approximate R^* .

First, we define three candidate range assignments. Recall that $S_1 = \sum_{i=1}^{n+1} d_i$. For the first range assignment, we search from the left of the axis and let $r_1 = d_1$. Then, the first range covers the region $[0, 2r_1]$. If $2r_1$ is larger than or equal to S_1 , we stop the search. Otherwise, we can find the smallest integer i_2 , $2 \leq i_2 \leq n$, satisfying $x_{i_2-1} \leq 2r_1 < x_{i_2}$ (in other words, it is the first sensor on the right of the already covered part). The next range will start from $2r_1$, using sensor μ_{i_2} , and let $r_{i_2} = x_{i_2} - 2r_1$ so that the new coverage does not overlap with the previous one. The search will stop if $x_{i_2} + r_{i_2} \geq S_1$. Otherwise, we can find the smallest integer i_3 , satisfying $x_{i_3-1} \leq x_{i_2} + r_{i_2} < x_{i_3} \dots$. We continue the process until there is no sensor on the right of the covered interval. If at this moment the whole interval $[0, m]$ is covered, we stop the algorithm. Otherwise, suppose that μ_{i_j} is the sensor we found in the last round, then let $r_{i_j} = S_1 - x_{i_j}$. We have a range assignment \mathbf{R}_l with cost $R_l = \sum_{k=1}^j r_{i_k}$. An intuitive example is shown in Figure 4.

If we search from right to left using the same method, we get a range assignment \mathbf{R}_r with cost R_r .

We design the third range assignment as follows. Let $k = \operatorname{argmin}_i (|x_i - S_1/2|)$. If $x_k \leq S_1/2$, let $r_k = S_1 - x_k$; otherwise, let $r_k = x_k$. We denote this range assignment as \mathbf{R}_c with cost $R_c = r_k$.

Among the three range assignments, we choose the one with the minimum cost as the output of our algorithm \mathbf{R} .

We prove the following theorem.

THEOREM 3.4. *The range assignment \mathbf{R}' is a $\frac{5}{4}$ -approximation.*

PROOF. By Lemma 3.2, if there exists one $d_i \geq S_1/2$, then we can easily find the optimal solution. Therefore, we assume that $d_i < S_1/2$ for all i in the following discussion. Denote P and Q as the left-most point and right-most point of the line interval I .

If there exists a sensor in the region $[\frac{3}{8}S_1, \frac{5}{8}S_1]$, then $R_c \leq \frac{5}{8}S_1$, which implies

$$R' \leq \frac{5}{4}S_1/2 \leq \frac{5}{4}R^*.$$

Otherwise, there is no sensor in $[\frac{3}{8}S_1, \frac{5}{8}S_1]$, which means that there exists a segment that is larger than $\frac{S_1}{4}$ in the “center” of PQ . Suppose that B_1, B_2 are the left and right end points of the segment. We divide the proof into three cases.

Case 1: There is no other segment whose length is at least $\frac{S_1}{4}$.

Define $b = |B_1B_2|$, $a = |PB_1|$, and $c = |B_2Q|$. Since $b < S_1/2$, we have

$$a + c = S_1 - b > S_1/2.$$

Without loss of generality, suppose that $c > \frac{S_1}{4}$. We consider the range assignment \mathbf{R}_l . Since there is no other segment with length at least $\frac{S_1}{4}$, the sensor μ_j at B_2 must be chosen to cover part of segment B_1B_2 because the previous choice of active sensors in assignment R_l have radii smaller than $\frac{S_1}{4}$. Furthermore, we have $r_j \leq b$. Considering the last range that may go beyond Q or cause some overlap of coverage, we have

$$2R_l \leq S_1 + \max(r_j - c, c - r'_j, r_i) \leq S_1 + \max(b - c, r_i),$$

where the term $r_j - c$ corresponds to the case when μ_j is the last sensor chosen in R_l and the coverage extends to the right of point Q , $c - r'_j$ corresponds to the case when μ_j is the last sensor chosen in R_l but covering the remaining part on the left of B_2 is not sufficient to reach point Q (in this case, we use r'_j to represent the radius that is sufficient to cover the remaining part on the left of B_2 and therefore sensor μ_j needs to use a radius r_j larger than r'_j by $c - r'_j$, causing an overlap of coverage with length $c - r'_j$, and since there is no other segment larger than $S_1/4$ in c , we know that $c - r'_j \leq S_1/4$), and r_i corresponds to the case when some sensor μ_i instead of μ_j is the last sensor chosen in R_l .

Since $b - c < S_1/2 - S_1/4 = S_1/4$ and $r_i < S_1/4$ because no other segment on the right of B_2 is larger than $S_1/4$, we have

$$2R_l \leq \frac{5}{4}S_1.$$

Therefore, we have

$$R' \leq \frac{5}{4}R^*.$$

Case 2: There is only one other segment that is at least $\frac{S_1}{4}$.

Let A_1, A_2 be the left and right end points of this segment. Because we consider both \mathbf{R}_l and \mathbf{R}_r , without loss of generality, we suppose that $[A_1, A_2]$ locates in the region $[0, \frac{3S_1}{8}]$. Since $x_{A_2} < \frac{3S_1}{8}$, we have

$$x_{A_1} < \frac{S_1}{8}.$$

If we use \mathbf{R}_l , there exists a range \mathbf{r}_j that partly covers A_1A_2 from the left, and $r_j < \frac{S_1}{8}$. It is easy to see that

$$x_{A_2} + r_{A_2} \geq \frac{S_1}{2}.$$

Therefore, the range \mathbf{r}_{A_2} at least partly covers $[B_1, B_2]$.

If the range \mathbf{r}_{A_2} totally covers $[B_1, B_2]$ or the range \mathbf{r}_{A_2} partly covers $[B_1, B_2]$ but $x_{B_2} + r_{B_2} \leq S_1$, more sensors are needed to cover the whole interval. We assume that μ_i is the last sensor chosen. Since no more segment is at least $\frac{S_1}{4}$ on the right of B_2 , we have $r_i < \frac{S_1}{4}$ no matter whether μ_i uses the radius sufficient to cover the remaining uncovered part on its left or extending that to cover point Q . Then, $2R_l \leq S_1 + r_i < S_1 + S_1/4 = \frac{5}{4}S_1$. Otherwise, the sensor at B_2 is chosen and $x_{B_2} + r_{B_2} > S_1$. Since

$$\frac{S_1}{4} < |B_1B_2| < \frac{S_1}{2},$$

we have

$$x_{B_2} < \frac{7S_1}{8}.$$

Furthermore, because

$$x_{B_2} > \frac{5S_1}{8},$$

we have

$$r_{B_2} < S_1 - \frac{5S_1}{8} = \frac{3S_1}{8}.$$

Therefore, we have

$$2R_l \leq x_{B_2} + r_{B_2} < \frac{7S_1}{8} + \frac{3S_1}{8} = \frac{5S_1}{4}.$$

Hence, we have

$$R' \leq \frac{5}{4}S_1/2 \leq \frac{5}{4}R^*.$$

Case 3: There are two other segments that are at least $\frac{S_1}{4}$.

We assume that these two segments are $[A_1, A_2]$ and $[C_1, C_2]$ with $[A_1, A_2]$ on the left. Because we consider both \mathbf{R}_l and \mathbf{R}_r , without loss of generality, we assume that $|A_1A_2| \geq |C_1C_2|$, then we have

$$x_{B_1} \geq x_{A_2} > \frac{2S_1}{8}$$

and

$$x_{B_2} \leq x_{C_1} < \frac{6S_1}{8}.$$

We still use \mathbf{R}_l in this case. If the sensor at x_{B_2} is chosen and $x_{B_2} + r_{B_2} > S_1$, then by similar argument, we have

$$R' \leq R_l \leq (x_{B_2} + r_{B_2})/2 \leq \frac{9}{8}R^*.$$

If B_2 is chosen and $x_{B_2} + r_{B_2} < S_1$ or \mathbf{r}_{A_2} totally covers $[B_1, B_2]$, let $a = |A_1A_2|$, $b = |B_1B_2|$ and $c = |C_1C_2|$ and sort a, b, c to an ascending sequence $X_3 \leq X_2 \leq X_1$. By Lemma 3.3., we have

$$R^* \geq \max(X_2 + X_3, X_1).$$

Since for each $i = 1, 2, 3$,

$$\frac{S_1}{4} \leq X_i < \frac{S_1}{2},$$



Fig. 5. The sensors' locations: $x_P = 0; x_{A_1} = \frac{1}{4}S_1; x_{A_2} = \frac{3}{8}S_1; x_{B_1} = \frac{5}{8}S_1; x_{B_2} = \frac{3}{4}S_1; x_Q = S_1$.

we have

$$R^* \geq X_2 + X_3.$$

On the other hand, we have

$$2R' \leq S_1 + \max(d_i),$$

where d_i can be the distance of any segment in the interval $[x_{B_2} + r_{B_2}, S_1]$. Hence, we have

$$2R' \leq S_1 + \max(d_i) \leq S_1 + c.$$

Since $A_1A_2 \geq C_1C_2$, we have $c \leq X_2$, which implies

$$2R' \leq S_1 + X_2.$$

Then,

$$\frac{R'}{R^*} - \frac{5}{4} \leq \frac{S_1 + X_2}{2(X_2 + X_3)} - \frac{5}{4} = \frac{2S_1 - (3X_2 + 5X_3)}{4(X_2 + X_3)}.$$

Since

$$X_2 \geq X_3 \geq \frac{S_1}{4},$$

we have

$$3X_2 + 5X_3 \geq 2S_1.$$

Then, we have

$$\frac{R'}{R^*} - \frac{5}{4} \leq 0.$$

Finally, we have

$$R' \leq \frac{5}{4}R^*. \quad \square$$

Next, we show that R' can do no better than $\frac{5}{4}$ -approximation for general inputs by providing one example.

As shown in Figure 5, We assume that there are six sensors in the range with length S_1 . We name these sensors P, A_1, A_2, B_1, B_2 , and Q , from left to right. Their locations are

$$\begin{aligned} x_P = 0; x_{A_1} = \frac{1}{4}S_1; x_{A_2} = \frac{3}{8}S_1; \\ x_{B_1} = \frac{5}{8}S_1; x_{B_2} = \frac{3}{4}S_1; x_Q = S_1. \end{aligned}$$

First of all, we consider the range assignment \mathbf{R}' , as shown in Figure 6.

In this case, we choose sensors A_1, B_1 , and Q . The first sensor A_1 covers the region $[0, \frac{1}{2}S_1]$ with its radius $r_{A_1} = \frac{1}{4}S_1$. The second sensor B_1 has a radius of $r_{B_1} = \frac{1}{8}S_1$. The last sensor Q covers the rest of this range with a radius of $r_Q = \frac{1}{4}S_1$.

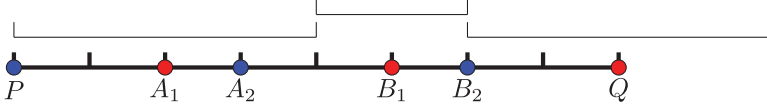


Fig. 6. The range assignment \mathbf{R}_l . Sensors A_1, B_1 , and Q are chosen. $R_l = \frac{1}{4}S_1 + \frac{1}{8}S_1 + \frac{1}{4}S_1 = \frac{5}{8}S_1$.

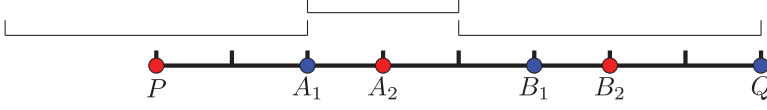


Fig. 7. The range assignment \mathbf{R}_r . Sensors P, A_2 , and B_2 are chosen. $R_r = \frac{1}{4}S_1 + \frac{1}{8}S_1 + \frac{1}{4}S_1 = \frac{5}{8}S_1$.

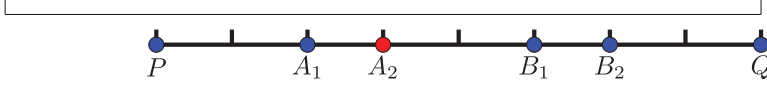


Fig. 8. The range assignment \mathbf{R}_c . Only sensor A_2 is chosen. $R_c = \frac{5}{8}S_1$.

Therefore, the cost of this range assignment is

$$R_l = \frac{1}{4}S_1 + \frac{1}{8}S_1 + \frac{1}{4}S_1 = \frac{5}{8}S_1.$$

Then, we consider the range assignment \mathbf{R}_r , as shown in Figure 7.

This time we choose three sensors: B_2, A_2 , and P . Their radii are $r_{B_2} = \frac{1}{4}S_1$, $r_{A_2} = \frac{1}{8}S_1$, and $r_P = \frac{1}{4}S_1$ correspondingly.

The cost of this assignment is

$$R_r = \frac{1}{4}S_1 + \frac{1}{8}S_1 + \frac{1}{4}S_1 = \frac{5}{8}S_1.$$

For the third assignment shown in Figure 8, only one sensor A_2 is chosen. Because $x_{A_2} < \frac{1}{2}S_1$, the radius of A_2 is $r_{A_2} = \frac{5}{8}S_1$. Obviously, the cost is

$$R_c = r_{A_2} = \frac{5}{8}S_1.$$

The optimal solution can be achieved by selecting sensors A_1 and B_2 as shown in Figure 9. Both of them have the radii $r_{A_1} = r_{B_2} = \frac{1}{4}S_1$. The cost of the optimal assignment is $R^* = \frac{1}{4}S_1 + \frac{1}{4}S_1 = \frac{1}{2}S_1$.

Since

$$R' = \min(R_l, R_r, R_c) = \frac{5}{8}S_1,$$

we have

$$R' = \frac{5}{4}R^*.$$

3.1.3. Dynamic Programming. If all of the x_i 's are integers, we can use dynamic programming to find an exact optimal solution for the $\kappa = 1$ case in pseudo-polynomial time.

Let function $Cost(i, l)$ denote the minimum cost of the assignment that covers the range $[0, l]$ using the first i sensors only. Let $A(i, l)$ represent the corresponding assignment of $Cost(i, l)$. Note that we allow l to take integer values only.

If we want to find the optimal assignment that covers the range $[0, l]$ with at most i sensors from $\{\mu_1, \dots, \mu_i\}$, then we can either use sensor μ_i or do not use μ_i . In the first

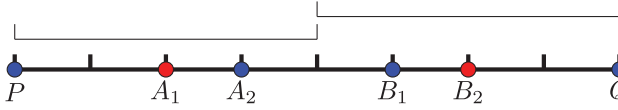


Fig. 9. The optimal solution. Sensor A_1 and B_2 are chosen. $R^* = \frac{1}{4}S_1 + \frac{1}{4}S_1 = \frac{1}{2}S_1$. Therefore, $\frac{R}{R^*} = \frac{\frac{5}{8}S_1 / \frac{1}{2}S_1}{\frac{1}{2}S_1} = \frac{5}{4}$.

case, we have two choices: using μ_i alone or using μ_i and some other sensors in the first $i - 1$ sensors. We separate these two cases because the right-most point covered by μ_i may be greater than l if we use μ_i alone. However, it is also possible that the right-most point covered by the optimal solution is greater than l if we use μ_i and some other sensors. But, if that happens, we can force μ_i to decrease its radius and cover l exactly and adjust the first $i - 1$ sensors' radii to cover the remaining part. Then, we can obtain an assignment whose cost is the same as the optimal solution, and the right-most point covered by the assignment is l . Hence, if the optimal solution to cover $[0, l]$ covers beyond l , then we can assume that it only uses one sensor. If we use μ_i alone, the total cost is $\max(x_i, l - x_i)$. If we use μ_i and some sensors from the first $i - 1$ sensors, since we use μ_i and it is the right-most selected sensor, its range is at least $[x_i - (l - x_i), x_i + (l - x_i)]$ if $l > x_i$. And the remaining range $[0, x_i - (l - x_i)]$ must be covered by the first $i - 1$ sensors. Hence, in this case, the total cost is $Cost(i - 1, 2x_i - l) + (l - x_i)$. Notice that if $l \leq x_i$, then selecting μ_i will cause waste, and we can just use the first $i - 1$ sensors to get better cost. In the second case, where we do not use μ_i , the cost is just $Cost(i - 1, l)$. Therefore, we can give the recurrence of $Cost$ as follows:

$$Cost(i, l) = \begin{cases} \min\{Cost(i - 1, l), \max(x_i, l - x_i), Cost(i - 1, 2x_i - l) + (l - x_i)\} & \text{if } l > x_i, \\ Cost(i - 1, l) & \text{otherwise.} \end{cases} \quad (5)$$

The initial value for $Cost(1, l)$ is defined as follows:

$$Cost(1, l) = \begin{cases} 0 & \text{if } l \leq 0 \\ x_1 & \text{if } 0 < l \leq 2x_1 \\ l - x_1 & \text{if } l > 2x_1 \end{cases} . \quad (6)$$

The optimal assignment for range $[0, m]$ using at most n sensors corresponds to $A(n, m)$, and the minimum cost of this optimal assignment is $Cost(n, m)$. To get the optimal solution, we need to create a table whose (i, l) entry records the value of $Cost(i, l)$ and assigns r_i for sensor μ_i . Since each entry can be computed in constant time, the whole table can be computed in $O(mn)$ time. Because m may not be bounded by a polynomial of n , the preceding dynamic programming runs in pseudo-polynomial time.

3.1.4. FPTAS. If m is bounded by a polynomial of n , then the dynamic programming gives the optimal solution in polynomial time. Hence, we scale down the value of each coordinate in the following way.

Choose a constant number $\epsilon > 0$. Let $K = \frac{\epsilon m}{3n+1}$. We scale each coordinate x_i to $x'_i = \lfloor \frac{x_i}{K} \rfloor$. Because $m' = \lfloor \frac{m}{K} \rfloor = \lfloor \frac{3n+1}{\epsilon} \rfloor$ is bounded by $poly(\frac{n}{\epsilon})$, the scaled instance can be solved by dynamic programming in $O(m'n) = O(\frac{n^2}{\epsilon})$.

Suppose that $\mathbf{S}' = (\mathbf{r}'_1, \mathbf{r}'_2, \dots, \mathbf{r}'_n)$ is the optimal assignment of the scaled instance. Let $r_i = r'_i * K + K/2$ for all r_i that is neither the first nor last nonzero r_i in the sequence, and let $r_i = r'_i * K + K$ otherwise. Then, $\mathbf{S} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$ is an assignment of the original instance as proved in Lemma 3.5. Suppose that $\mathbf{O} = (\mathbf{r}'_1, \mathbf{r}'_2, \dots, \mathbf{r}'_n)$ is the optimal assignment of the original instance. Let $r'_i = \lfloor \frac{r_i}{K} \rfloor + 1$. Then, $\mathbf{O}' = (\mathbf{r}'_1, \mathbf{r}'_2, \dots, \mathbf{r}'_n)$ is an assignment of the scaled instance.

LEMMA 3.5. *To obtain a feasible assignment for the original instance from a feasible solution of its scaled instance, we need to add at most $\frac{n+1}{2}K$ to the summation of all radii.*

PROOF. If the distance between the adjacent chosen sensors in the scaled instance is d , then after we transform the instance back to the original one, the corresponding distance will increase to at most $dK + K$. Therefore, assigning $r_i = Kr'_i + K/2$ can guarantee the coverage of all intermediate segments. However, for the left-most segment and the right-most segment, only one sensor is chosen to cover each of them, so these two sensors' radii need to be $Kr'_i + K$ to cover these two segments.

Hence, we need to add at most $\frac{n+1}{2}K$ to recover the original instance. \square

We have the following deductions:

$$C(\mathbf{S}) \leq K \cdot C(\mathbf{S}') + K \cdot \frac{n+1}{2} \quad (7)$$

$$\leq K \cdot C(\mathbf{O}') + K \cdot \frac{n+1}{2} \quad (8)$$

$$\leq C(\mathbf{O}) + K \cdot n + K \cdot \frac{n+1}{2} \quad (9)$$

$$\leq C(\mathbf{O}) + \frac{\epsilon m}{2} \quad (10)$$

$$\leq C(\mathbf{O}) + \epsilon C(\mathbf{O}). \quad (11)$$

Because \mathbf{S}' is the optimal assignment of the scaled instance, we have $C(\mathbf{S}') \leq C(\mathbf{O}')$. Since the union of the ranges in the assignment has to cover the whole interval $[0, m]$, the summation of these ranges is at least $m/2$, which implies Inequality (11).

Therefore, we can scale down the instance in $O(n)$ time and use dynamic programming to get an optimal solution \mathbf{S} for the scaled instance. In addition, the cost of \mathbf{S} in the original instance is at most $(1 + \epsilon)$ larger than the optimal solution. The running time of the dynamic programming is $O(\frac{n^2}{\epsilon})$, which is polynomial in both n and $\frac{1}{\epsilon}$ and therefore the proposed method is an FPTAS.

3.2. $\kappa > 1$

In this section, we discuss the range assignment problem when the cost of assignment $R = \sum r_i^\kappa$ ($\kappa > 1$).

The following inequality is a special instance of the generalized mean inequality.

LEMMA 3.6. *If $x, y > 0$ and $\kappa \geq 1$, then $(x + y)^\kappa \leq 2^{\kappa-1}(x^\kappa + y^\kappa)$.*

Since $S_\kappa = \sum_{i=1}^{n+1} d_i^\kappa$, if $\mathbf{R}^* = (\mathbf{r}_1^*, \mathbf{r}_2^*, \dots, \mathbf{r}_n^*)$ is an optimal assignment, then we have the following lemma.

LEMMA 3.7. $S_\kappa \leq 2^\kappa R^* - (2^{\kappa-1} - 1)(d_1^\kappa + d_{n+1}^\kappa)$.

PROOF. In the optimal assignment \mathbf{R}^* , suppose that $\mathbf{r}_{i_1}^*, \mathbf{r}_{i_2}^*, \dots, \mathbf{r}_{i_j}^*$ are nontrivial ranges with $i_1 < i_2 < \dots < i_j$. Then, we have the following inequalities:

$$\begin{aligned} d_1 + \dots + d_{i_1} &\leq r_{i_1}^*, \\ d_{i_1+1} + \dots + d_{i_2} &\leq r_{i_1}^* + r_{i_2}^*, \\ &\dots \\ d_{i_{j-1}+1} + \dots + d_{i_j} &\leq r_{i_{j-1}}^* + r_{i_j}^*, \\ d_{i_j+1} + \dots + d_{n+1} &\leq r_{i_j}^*. \end{aligned}$$



Fig. 10. The sensors' locations: $x_A = x_B = \frac{1}{4}; x_C = x_D = \frac{3}{4}$.

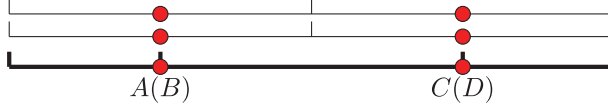


Fig. 11. The solution output by Theorem 3.8. All sensors are chosen. $R = (\frac{1}{4})^2 + (\frac{1}{4})^2 + (\frac{1}{4})^2 + (\frac{1}{4})^2 = \frac{1}{4}$.

Hence, by Lemma 3.6, we have

$$\begin{aligned}
 S_\kappa &\leq (d_1 + \dots + d_{i_1})^\kappa + (d_{i_1+1} + \dots + d_{i_2+1})^\kappa + \dots + (d_{i_{j-1}+1} + \dots + d_{i_j})^\kappa \\
 &\quad + (d_{i_j+1} + \dots + d_{n+1})^\kappa \\
 &\leq (r_{i_1}^*)^\kappa + (r_{i_1}^* + r_{i_2}^*)^\kappa + \dots + (r_{i_{j-1}}^* + r_{i_j}^*)^\kappa + (r_{i_j}^*)^\kappa \\
 &\leq (r_{i_1}^*)^\kappa + 2^{\kappa-1}((r_{i_1}^*)^\kappa + (r_{i_2}^*)^\kappa) + 2^{\kappa-1}((r_{i_2}^*)^\kappa + (r_{i_3}^*)^\kappa) + \dots + 2^{\kappa-1}((r_{i_{j-1}}^*)^\kappa + (r_{i_j}^*)^\kappa) + (r_{i_j}^*)^\kappa \\
 &= 2^\kappa R^* - (2^{\kappa-1} - 1)((r_{i_1}^*)^\kappa + (r_{i_j}^*)^\kappa) \\
 &\leq 2^\kappa R^* - (2^{\kappa-1} - 1)((d_1 + \dots + d_{i_1})^\kappa + (d_{i_j+1} + \dots + d_{n+1})^\kappa) \\
 &\leq 2^\kappa R^* - (2^{\kappa-1} - 1)(d_1^\kappa + d_{n+1}^\kappa).
 \end{aligned}$$

The lemma can also be easily verified if there is only one active sensor in the optimal assignment. \square

THEOREM 3.8. *Let $r_1 = \max(d_1, \frac{d_2}{2})$, $r_2 = \max(\frac{d_2}{2}, \frac{d_3}{2})$, \dots , $r_{n-1} = \max(\frac{d_{n-1}}{2}, \frac{d_n}{2})$, and $r_n = \max(\frac{d_n}{2}, d_{n+1})$, then the range assignment $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$ is a 2-approximation.*

PROOF. When $n = 1$, \mathbf{R} is the optimal assignment. When $n \geq 2$, we have

$$\begin{aligned}
 R &\leq \max(d_1^\kappa, 2^{-\kappa} d_2^\kappa) + \max(2^{-\kappa} d_2^\kappa, 2^{-\kappa} d_3^\kappa) + \dots + \max(2^{-\kappa} d_{n-1}^\kappa, 2^{-\kappa} d_n^\kappa) \\
 &\quad + \max(2^{-\kappa} d_n^\kappa, d_{n+1}^\kappa) < d_1^\kappa + 2^{1-\kappa} * (d_2^\kappa + d_3^\kappa + \dots + d_n^\kappa) + d_{n+1}^\kappa \\
 &= 2^{1-\kappa} S_\kappa + (1 - 2^{1-\kappa})(d_1^\kappa + d_{n+1}^\kappa).
 \end{aligned}$$

Since $S_\kappa \leq 2^\kappa R^* - (2^{\kappa-1} - 1)(d_1^\kappa + d_{n+1}^\kappa)$, by Lemma 3.7, we have

$$\begin{aligned}
 R &< 2^{1-\kappa} (2^\kappa R^* - (2^{\kappa-1} - 1)(d_1^\kappa + d_{n+1}^\kappa)) + (1 - 2^{1-\kappa})(d_1^\kappa + d_{n+1}^\kappa) \\
 &= 2R^* - (1 - 2^{1-\kappa})(d_1^\kappa + d_{n+1}^\kappa) + (1 - 2^{1-\kappa})(d_1^\kappa + d_{n+1}^\kappa) \\
 &= 2R^*.
 \end{aligned}$$

Therefore, $\frac{R}{R^*} < 2$. \square

In fact, approximation ratio 2 is also the best that we can prove for the preceding range assignment. Consider the example shown in Figure 10, where the whole interval is $[0, 1]$. The first two sensors are at $1/4$, and the last two sensors are at $3/4$. In this example, the cost of the assignment in Theorem 3.8 (Figure 11) is exactly twice the cost of the optimal assignment (Figure 12).

4. EXPERIMENTS

In this section, we evaluate the performance of our approximation algorithms with randomly generated sensor positions (uniformly chosen along the line).

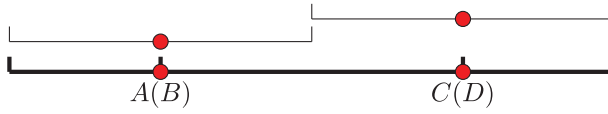


Fig. 12. The optimal solution. Sensors A and C are chosen. $R^* = (\frac{1}{4})^2 + (\frac{1}{4})^2 = \frac{1}{8}$. Therefore, $\frac{R}{R^*} = \frac{1}{4} / \frac{1}{8} = 2$.

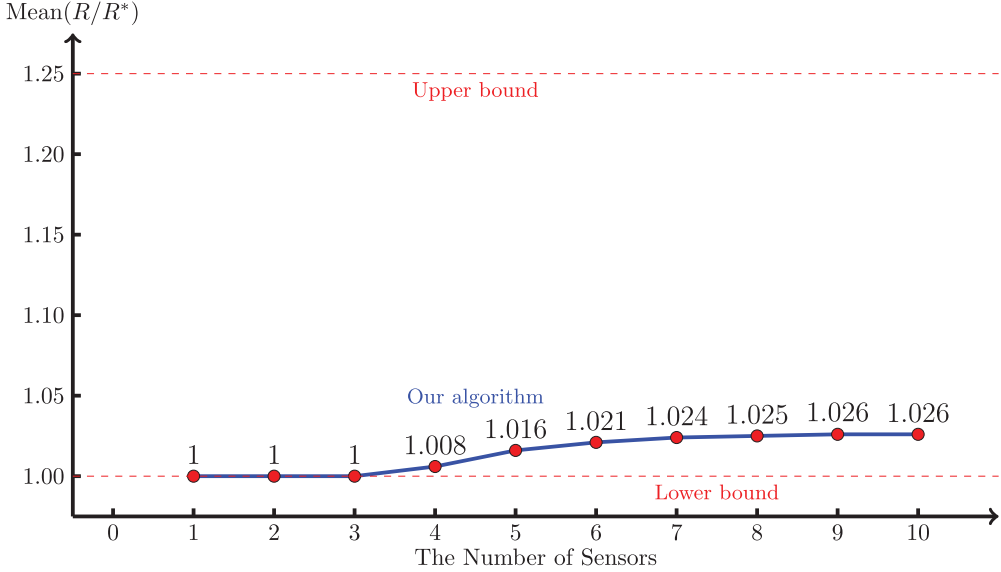


Fig. 13. This graph shows the comparison between our algorithm and the optimal solution when $\kappa = 1$. The x -axis is the number of sensors, and the y -axis is the average value of our solution over the optimal solution (Mean(R/R^*)). The two horizontal dashed lines are the theoretical upper bound and lower bound of our algorithm.

In the experiment, R^* is the cost of optimal range assignment, which is achieved by calculating all possible assignments without any range overlapping on both sides. R is the cost of range assignment obtained by our $\frac{5}{4}$ -approximation algorithm. We uniformly choose the locations of sensors at random and calculate R^* , R , and $\frac{R}{R^*}$.

The purpose of this experiment is to compare the result given by our algorithm with the optimal solution for cases when $\kappa = 1$ and $\kappa = 2$. To the best of our knowledge, very few researchers are doing exactly the same problem. In Bar-Noy and Baumer [2011], the authors proposed an algorithm called *RoundRobin*, where each round uses a single sensor to cover the line interval. Our center assignment algorithm is a special case of their methods with the minimum energy consumption, and because our $\frac{5}{4}$ -approximation algorithm takes the best from three assignments, the performance of our algorithm can only be better, and therefore we do not compare our algorithm experimentally with other algorithms.

4.1. $\kappa = 1$

We vary the number of sensors from 1 to 10. For every fixed number of sensors, 10,000 cases are generated; the average ratio is recorded in Figure 13. In the figure, the x -axis is the number of sensors, and the y -axis shows the mean value of $\frac{R}{R^*}$. The distribution is shown in Figure 14.

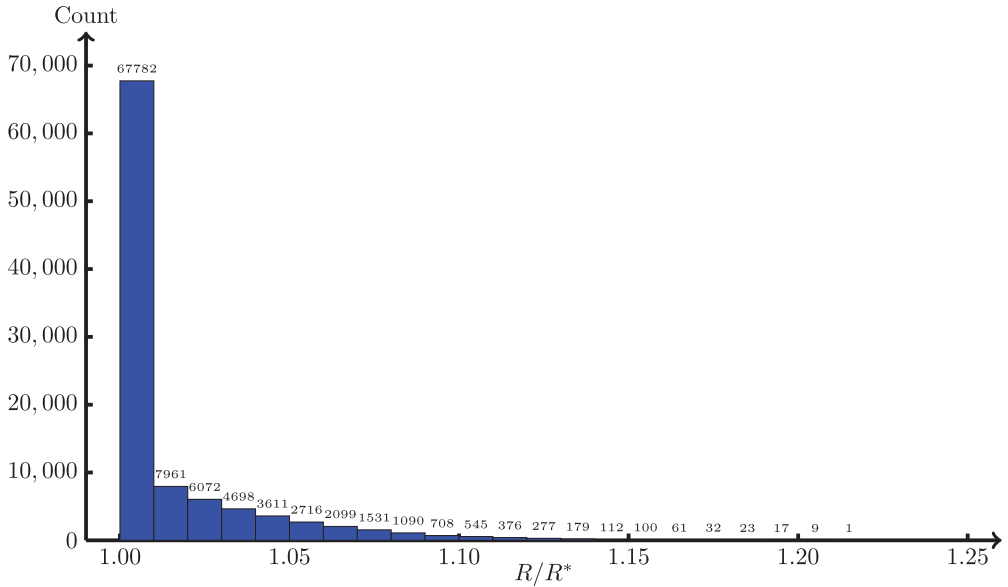


Fig. 14. Distribution graph when $\kappa = 1$. 1.25 is our theoretical upper bound when $\kappa = 1$. From this graph, we find that very few cases go beyond 1.1, and more than 90% of cases are between 1.00 and 1.05. Our algorithm works well for random inputs.

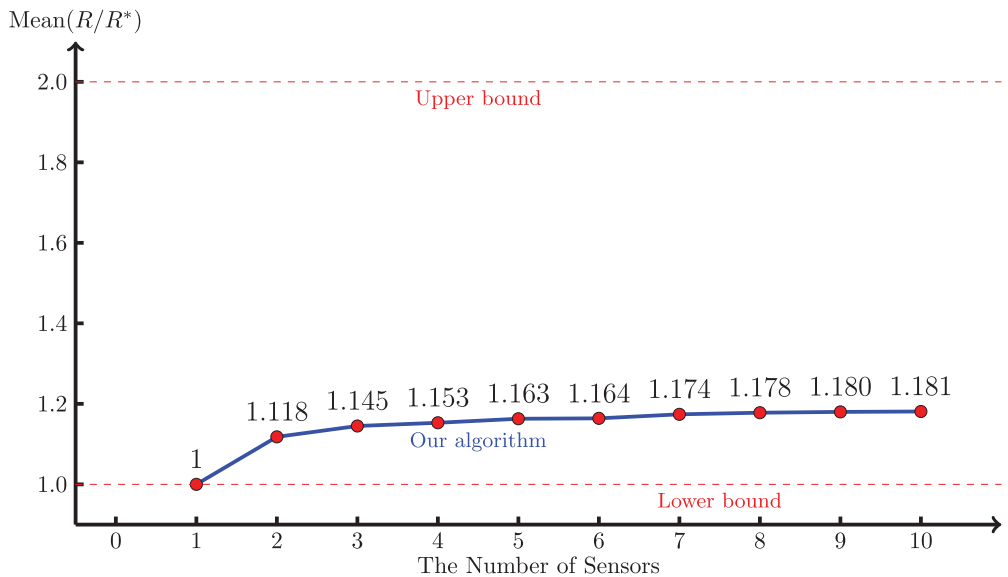


Fig. 15. This graph shows the comparison between our algorithm and the optimal solution when $\kappa = 2$. The x -axis is the number of sensors, and the y -axis is the average value of our solution over the optimal solution ($\text{Mean}(R/R^*)$). The two horizontal dashed lines are the theoretical upper bound and lower bound of our algorithm.

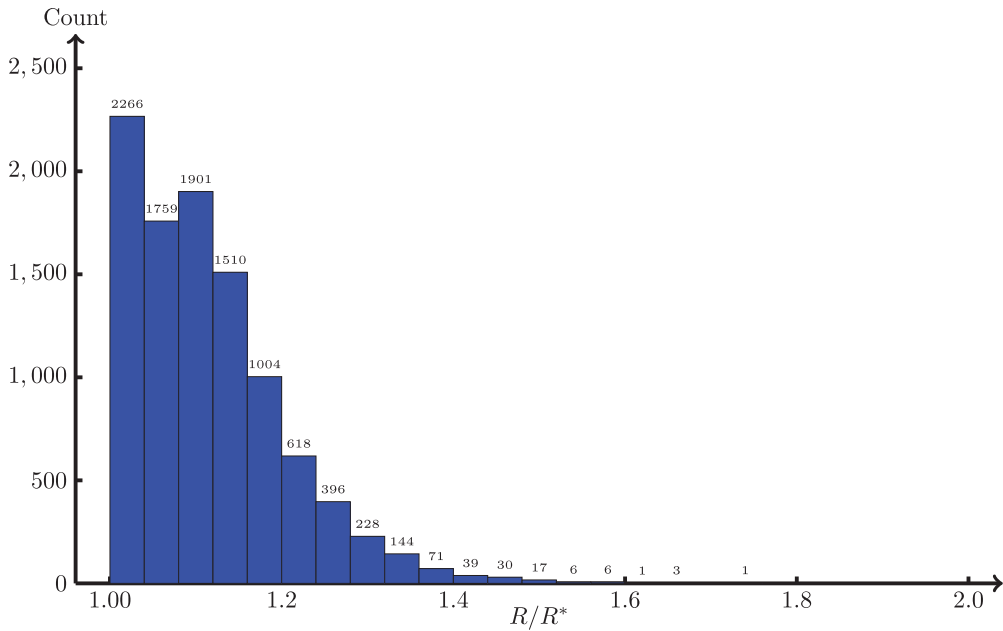


Fig. 16. Distribution graph when $\kappa = 2$. The value 2.0 is the theoretical upper bound of our algorithm when $\kappa = 2$. However, very few cases reach the upper bound. Most cases fall in the range from 1 to 1.2. Our algorithm works well for random inputs.

4.2. $\kappa = 2$

We vary the number of sensors from 1 to 10. For every fixed number of sensors, 1,000 cases are generated; the average ratio is recorded in Figure 15. We can see that the actual average performance of our algorithm is much better than the theoretical approximation ratio 2. The distribution is shown in Figure 16.

5. CONCLUSION

In this article, we study the coverage of a line interval with a set of wireless sensors with adjustable coverage ranges. The objective is to find a range assignment with the minimum cost. For the discrete variant, we present a polynomial-time algorithm to compute the optimal solution. For the continuous variant, we develop constant-approximation algorithms when the cost for all sensors is proportional to r^κ for some constant $\kappa \geq 1$, where r is the covering radius corresponding to the chosen power. Specifically, if $\kappa = 1$, we give a simple 1.25-approximation algorithm and an FPTAS; if $\kappa > 1$, we give a simple 2-approximation algorithm. Possible future directions are improving the approximation ratio for the continuous variant for different values of κ and extend the results to covering a 2D region.

REFERENCES

- Z. Abrams, A. Goel, and S. Plotkin. 2004. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. ACM, New York, NY, 424–432.
- C. Ambuhl, T. Erlebach, M. Mihalák, and M. Nunkesser. 2006. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. *Lecture Notes in Computer Science*, Vol. 4110, 3–14.
- A. Bar-Noy and B. Baumer. 2011. Maximizing network lifetime on the line with adjustable sensing ranges. In *Proceedings of the 7th International Conference on Algorithms for Sensor Systems, Wireless Ad Hoc*

Networks and Autonomous Mobile Entities (ALGOSENSORS'11). Lecture Notes in Computer Science, Vol. 7111, 28–41.

- A. Bar-Noy, B. Baumer, and D. Rawitz. 2012. *Set it and forget it: Approximating the set once strip cover problem*. Technical Report 1204.1082, CoRR.
- A. Bar-Noy, D. Rawitz, and P. Terlecky. 2013. Maximizing barrier coverage lifetime with mobile sensors. In *Proceedings of the 21st Annual ESA Symposium*. 97–108.
- H. Brönnimann and M. Goodrich. 1995. Almost optimal set covers in finite VC-dimension. *Discrete and Computational Geometry* 14, 1, 463–479.
- G. Călinescu, I. Mandoiu, P. Wan, and A. Zelikovsky. 2004. Selecting forwarding neighbors in wireless ad hoc networks. *Mobile Networks and Applications*, 9, 2, 101–111.
- P. Carmi, M. Katz, and N. Lev-Tov. 2007. Covering points by unit disks of fixed location. In *Proceedings of the 18th International Conference on Algorithms and Computation*. 644–655.
- D. Chen, Y. Gu, J. Li, and H. Wang. 2013. Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. *Discrete and Computational Geometry* 50, 2, 374–408.
- J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. Yazdani. 2009. On minimizing the maximum sensor movement for barrier coverage of a line segment. in *ADHOC-NOW 2009*. LNCS, vol. 5793, pp. 194–212.
- J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. Yazdani. 2010. On minimizing the sum of sensor movements for barrier coverage of a line segment. In *Proceedings of the 9th International Conference on Ad-hoc, Mobile and Wireless Networks*. Lecture Notes in Computer Science, vol. 6288, 29–42.
- D. Dai and C. Yu. 2009. A $5 + \epsilon$ -approximation algorithm for minimum weighted dominating set in unit disk graph. *Theoretical Computer Science* 410, 8–10, 756–765.
- T. Erlebach and M. Mihalák. 2010. A $(4 + \epsilon)$ -approximation for the minimum-weight dominating set problem in unit disk graphs. Lecture Notes in Computer Science, vol. 5893, 135–146.
- T. Gonzalez. 1991. Covering a set of points in multidimensional space. *Information Processing Letters* 40, 4, 181–188.
- D. Hochbaum and W. Maass. 1985. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM* 32, 1, 130–136.
- J. Hou, D. Yau, C. Ma, Y. Yang, H. Zhang, I. Hou, N. Rao, and M. Shankar. 2009. Coverage in wireless sensor networks. In *Guide to Wireless Sensor Networks*, 47–49.
- Y. Huang, X. Gao, Z. Zhang, and W. Wu. 2009. A better constant-factor approximation for weighted dominating set in unit disk graph. *Journal of Combinatorial Optimization* 18, 2, 179–194.
- M. Li, X. Sun, and Y. Zhao. 2011. Minimum-cost linear coverage by sensors with adjustable ranges. In *Proceedings of the 6th International Conference on Wireless Algorithms, Systems, and Applications (WASA'11)*. Lecture Notes in Computer Science, vol. 6843, 25–35.
- M. Mehrandish, L. Narayanan, and J. Opatrny. 2011. Minimizing the number of sensors moved on line barriers. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'11)*. 653–658.
- S. Narayanappa and P. Vojtechovský. 2006. An improved approximation factor for the unit disk covering problem. In *Proceedings of the Canadian Conference on Computational Geometry*.
- P.-J. Wan, X.-H. Xu, and Z. Wang. 2011. Wireless coverage with disparate ranges. In *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'11)*. Article No. 11.
- P.-J. Wan, D. Chen, G. Dai, Z. Wang, and F. Yao. 2012. Maximizing capacity with power control under physical interference model in duplex mode. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12)*. 415–423.
- P. Wan and C. Yi. 2006. Coverage by randomly deployed wireless sensor networks. *IEEE Transactions on Information Theory* 52, 2658–2669.
- J. Wang and N. Zhong. 2006. Efficient point coverage in wireless sensor networks. *Journal of Combinatorial Optimization* 11, 3, 291–304.
- Z. Yun, X. Bai, D. Xuan, T. Lai, and W. Jia. 2010. Optimal deployment patterns for full coverage and k -connectivity ($k \leq 6$) wireless sensor networks. *IEEE/ACM Transactions on Networking* 18, 3, 934–947.
- F. Zou, Y. Wang, X. Xu, X. Li, H. Du, P. Wan, and W. Wu. 2009. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science* 412, 3, 198–208.

Received August 2013; revised January 2014; accepted January 2014