

Wireless Coverage with Disparate Ranges

Peng-Jun Wan, Xiaohua Xu, Zhu Wang

Department of Computer Science

Illinois Institute of Technology

Chicago, IL 60616

wan@cs.iit.edu, xxu23@iit.edu, zwang59@iit.edu

ABSTRACT

Coverage has been one of the most fundamental yet challenging issues in wireless networks. Given a set of nodes and a set of disks of disparate radii, the problem Minimum Disk Cover seeks a disk cover of all nodes with minimum cardinality. We present the first polynomial time approximation scheme. We also consider a classical generalization where each input disk is associated with a positive cost, the problem Min-Cost Disk Cover seeks a disk cover of all nodes with minimum total cost. We present a randomized algorithm that can achieve an approximation ratio of $2^{O(\log^* n)}$ with high probability, where n is the number of input disks.

Another line of this work is exploring the relations between disk cover and an important practical problem which seeks a wireless covering schedule of maximum life subject to an energy budget function. We present two algorithms: Ellipsoid Algorithm (EA) and Price-Directive Algorithm (PDA), and prove that by applying our algorithmic results on disk cover, the approximation ratios for EA and PDA are $2^{O(\log^* n)}$ and $(1 + \epsilon) 2^{O(\log^* n)}$ respectively.

Categories and Subject Descriptors

C.2.1 [Computer-Communication networks]: Network Architecture and Design

General Terms

Algorithms, Design, Theory

Keywords

Wireless networks, disk cover, cost, budget, lifetime.

1. INTRODUCTION

In wireless networks, many challenges are intrinsically related to the coverage issue. Moreover, most kinds of activities for a wireless system, such as sending/receiving data, tracking, event monitoring can not be realized without coverage. Thus, coverage plays a critical role for wireless networking performances.

As a classical coverage problem, *disk cover* has drawn a lot of research interest independently. Formally, the problem can be defined as follows: given a set \mathcal{D} of disks of arbitrary

radii and a set \mathcal{P} of nodes in the plane, a node $p \in \mathcal{P}$ is *covered* by a disk $D \in \mathcal{D}$ if p lies within the disk D (i.e., $p \in D$). A subset $\mathcal{D}' \subseteq \mathcal{D}$ of disks is said to be a *disk cover* of \mathcal{P} if each node in \mathcal{P} is covered by at least one disk from \mathcal{D}' . The problem **Minimum Disk Cover (MDC)** seeks a smallest disk cover $\mathcal{D}' \subseteq \mathcal{D}$ of \mathcal{P} . We further consider a classical generalization where each disk is associated with a positive cost, the problem **Min-Cost Disk Cover (MCDC)** seeks a minimum cost disk cover $\mathcal{D}' \subseteq \mathcal{D}$ of \mathcal{P} .

The disk cover problem is a geometric set cover problem where the given sets are defined by disks. It has been proved to be NP-hard even with the restriction that all disks have uniform radii [15]. For the disk cover problem with such a restriction, tremendous work [3–5, 19] is done and it can admit a constant-approximation. However, we will investigate both the MDC and MCDC problems without such a restriction, i.e., all disks may have disparate radii in this work.

Another line of this work is exploring the intrinsic relationship between disk cover and an important practical problem in wireless networks, called Maximum Life Wireless Coverage (MLWC). Assume the wireless nodes collaborate with each other to cover (or watch) all the targets, then a *wireless cover* is defined as a subset of wireless nodes such that each target (which can be devices, sensors *etc.*) is covered by some wireless node(s). Assume each wireless node u has an initial coverage power $p(u)$, then, for a wireless cover denoted by C , the coverage power consumption by the wireless node u is defined as: when $u \in C$, $p_C(u) = p(u)$; and when $u \notin C$, $p_C(u) = 0$, this means that u has no power consumption at this time. From both economic and applicable concerns, we require a wireless network system to sustain and provide the coverage service as long as possible. However, most wireless nodes are powered by batteries and have a stringent energy budget. Thus, we need to design a maximal lifetime wireless covering schedule subject to the energy budget.

We formulate the challenging problem as follows, given a wireless network system consisting of a set of targets, and a set of wireless nodes with disparate covering ranges, a target is *covered* by a wireless node if and only if the target lies within the covering range of the wireless node. For the purpose of unified treatment, a coverage task is given implicitly

by a collection \mathcal{C} of wireless covers. A covering schedule subject to an energy budget function $b \in \mathbb{R}_+^V$ is a set of pairs $(C_i, x_i) \in \mathcal{R} \times \mathbb{R}_+$ for $i = 1, \dots, k$ satisfying that

$$\sum_{i=1}^k p_{C_i}(u) x_i \leq b(u), \forall u \in V.$$

Here $C_i \in \mathcal{C}$ is a wireless cover, and $x_i \in \mathbb{R}^+$ is the length scheduled for C_i with \mathbb{R}^+ denoting the set of all positive real numbers. The life (or length) of this schedule is defined to be $\sum_{i=1}^k x_i$. The problem **Max-Life Wireless Coverage (MLWC)** seeks a covering schedule of maximum life subject to an energy budget function $b \in \mathbb{R}_+^V$.

Our Main Contributions: In this work, we will present algorithm design with theoretical analysis for the problems MDC, MCDC, and MLWC respectively:

1. For the problem **MDC**, we present the first polynomial time approximation scheme (PTAS) based on a recent breakthrough result [18].
2. For the problem **MCDC**, we propose a randomized algorithm that can achieve an approximation ratio of $2^{O(\log^* n)}$, with high probability, where n is the number of input disks and $\log^* n$ is the smallest number of iterated “logarithms” applied to n to yield a constant.
3. For the problem **MLWC**, we present two algorithms: Ellipsoid Algorithm (EA) and Price-Directive Algorithm (PDA). Here PDA is adapted from the algorithm in [8] for fractional packing problems. We prove that by utilizing the algorithmic results for disk cover, the approximation ratio for EA and PDA are $2^{O(\log^* n)}$ and $(1 + \epsilon) 2^{O(\log^* n)}$ respectively.

The rest of the paper is organized as follows: In Section 2, we perform a thorough literature review for disk cover and related problems. In Section 3, we present the first PTAS for the problem MDC. In Section 4, we propose a randomized algorithm for the problem MCDC. In Section 5, we study the problem MLWC, and present two algorithms: Ellipsoid Algorithm (EA) and Price-Directive Algorithm (PDA). We conclude our paper in Section 6.

2. LITERATURE REVIEW

Since there are no existing work touching the maximum life coverage problem, to the best of our knowledge, we will only present literature reviews for disk cover. As we know, disk cover is a classical geometric *set cover* problem. It is NP-hard [15] even for unit disks, while it admits a constant-approximation. Note that the general set cover problem is not approximable within $O(\log n)$, where n is the number of sets [20].

2.1 Minimum Disk Cover

One main branch for disk cover that receives great research interest is called *discrete unit disk cover*, and there have been a series of work done for it [3–5, 19]. Specifically, Brönnimann and Goodrich [3] presented a deterministic ϵ -net based algorithm where the approximation ratio is

Table 1: Algorithmic results for disk cover

Objective	Min-Size		Min-Cost	
	uniform	arbitrary	uniform	arbitrary
Best result	$1 + \epsilon$	none	$4 + \epsilon$	none

not specified. Calinescu *et al.* [4] gave a 102-approximation algorithm. Narayanappa and Vojtechovsky [19] improved the approximation ratio to 72. Carmi *et al.* [5] gave a 38-approximation algorithm by solving a subproblem where the nodes to be covered lie below a line and all disks lie above the line.

For the problem *continuous disk cover* with minimum cardinality, where the disk center locations may be chosen at any point in the plane, it admits a PTAS by using a grid-shifting strategy [11, 12].

2.2 Min-Cost Disk Cover

For a classical generalization of minimum disk cover, which is *min-cost disk cover*, [26] proposed $(4 + \epsilon)$ -approximation algorithm for unit disk graph (UDG), which is the best result so far. Note that, a lot of recent algorithmic results [2, 6, 7, 14] for *min-weight dominating set* imply algorithms for min-cost disk cover with the same approximation ratios.

In summary, we list the algorithmic results for different variants of the disk cover problem in Table 1.

2.3 Related Problems and Useful Techniques

In spite of disk cover, there are other related work for coverage [1, 13, 23–25]. Yun *et al.* [25] studied deployment patterns to achieve full coverage and k -connectivity under different ratios of the sensor communication range to the sensing range for homogeneous wireless networks. Wan *et al.* [23] analyzed the probability of the k -coverage when the sensing radius or the number of sensors changes while taking the boundary effect into account.

Recently, Mustafa and Ray [17, 18] proposed a PTAS for the discrete geometric hitting set problem. Based on their techniques, Gibson *et al.* [9, 10] gave a PTAS for the unweighted case, and $2^{O(\log^* n)}$ -approximation for the weighted case of the problem minimum dominating set in disk intersection graph with arbitrary disk radii.

3. PTAS FOR MINIMUM DISK COVER

3.1 Algorithm Design

Our method will be based on a k -level *local search* meta-heuristic where k will control the approximation factor.

Given an input instance of MDC, let us first remove every disk properly contained in some other disk from \mathcal{D} , and obtain a new instance. For this new instance, we begin with the disk set \mathcal{N} equal to the set of all disks, which is clearly a disk cover of the input node set \mathcal{P} . Let $|\mathcal{B}|$ denote the size of a disk set \mathcal{B} , we replace any subset \mathcal{B} of \mathcal{N} with size at

Local Search Algorithm:
Input: a disk set \mathcal{D} , a node set \mathcal{P} , a parameter k ; $\mathcal{N} \leftarrow \mathcal{D}$; for any disk $D \in \mathcal{N}$ for any disk $D' \in \mathcal{N} \setminus \{D\}$ if D' properly contains D $\mathcal{N} \leftarrow \mathcal{N} \setminus \{D\}$; break; for any disk subset \mathcal{B} of \mathcal{N} with $ \mathcal{B} \leq k$ for any disk subset \mathcal{B}' of \mathcal{D} with $ \mathcal{B}' \leq \mathcal{B} - 1$ if $(\mathcal{N} \cup \mathcal{B}') \setminus \mathcal{B}$ cover \mathcal{P} $\mathcal{N} \leftarrow (\mathcal{N} \cup \mathcal{B}') \setminus \mathcal{B}$ return \mathcal{N} .

Table 2: Local Search Algorithm

most k by a subset of \mathcal{D} with size at most $|\mathcal{B}| - 1$, if the disk set \mathcal{N} after replacement is still a disk cover of \mathcal{P} . We keep replacing until we can not further reduce the size of \mathcal{N} . Finally, we output the disks in \mathcal{N} . The details are shown in Table 2.

3.2 Performance Analysis

In this section, we will show that the proposed algorithm for the problem MDC is a PTAS.

First, it is easy to verify that the algorithm will terminate after at most $\sum_{i=1}^k \binom{n}{i}$ replacements which means that it is a polynomial-time algorithm. Second, we can verify that the optimal solution for the new instance (obtained in Subsection 3.1) where no disk is properly contained in any other disk in \mathcal{D} , has the same cardinality as the optimal solution for the original instance. Thus, we only need to focus on this new instance and prove that the proposed algorithm can achieve an approximation ratio of $1 + \epsilon$ for it. Specifically, for this new instance of MDC, let \mathcal{B} be the solution returned by the proposed algorithm, let \mathcal{R} be an optimal solution ($|\mathcal{R}| \geq 2$), we want to prove that $|\mathcal{B}| \leq (1 + \epsilon) \cdot |\mathcal{R}|$. Our arguments will rely on a concept of *locality condition* for \mathcal{R} and \mathcal{B} .

LEMMA 1. (*Locality condition*) *There is a planar bipartite graph $G = (\mathcal{R}, \mathcal{B}, E)$ on disk sets \mathcal{R} and \mathcal{B} . For any node $p \in \mathcal{P}$, there exists an edge connecting two disks both covering p , one disk from \mathcal{R} and the other from \mathcal{B} .*

Here, we say that an edge connects two disks R and B if this edge connects the centers of R and B , we will keep this convention from now on. The proof of Lemma 1 (construct a graph on \mathcal{R}, \mathcal{B} that satisfy the locality condition) is delayed to Section 3.3. We first explore the property of locality condition and relate the size of \mathcal{R} to the size of \mathcal{B} . Specifically, we show that: $|\mathcal{B}| \leq (1 + \epsilon) \cdot |\mathcal{R}|$.

Let us first assume $\mathcal{R} \cap \mathcal{B} = \emptyset$, later, we will consider the general case that $\mathcal{R} \cap \mathcal{B} \neq \emptyset$. In the planar graph G satisfying

the locality condition, for any disk D , we denote by $N_G(D)$ the set of neighbors of D . Similarly, for any subset \mathcal{D}' of disks from G , denote by $N_G(\mathcal{D}')$ the set of all neighbors of the disks in \mathcal{D}' .

LEMMA 2. $\forall \mathcal{B}' \subseteq \mathcal{B}$ with $|\mathcal{B}'| \leq k$, we have $|N_G(\mathcal{B}')| \geq |\mathcal{B}'|$.

PROOF. Otherwise the proposed local search algorithm can perform further replacement (by replacing \mathcal{B}' with $N_G(\mathcal{B}')$). This leads to a contradiction. \square

We then show that the proposed algorithm can achieve $(1 + \epsilon)$ -approximation.

THEOREM 1. $\forall \epsilon > 0$, the algorithm in Table 2 returns a feasible solution \mathcal{B} with $|\mathcal{B}| \leq (1 + \epsilon) \cdot |\mathcal{R}|$.

PROOF. Let us first introduce a result from [18], which is obtained from a classical property on planar graph partition.

LEMMA 3. ([18]) *Let $G = (\mathcal{R}, \mathcal{B}, E)$ be a planar bipartite graph on vertex sets \mathcal{R} and \mathcal{B} , $|\mathcal{R}| \geq 2$, such that for every subset $\mathcal{B}' \subseteq \mathcal{B}$ of size at most k , where k is a large enough number, $|N_G(\mathcal{B}')| \geq |\mathcal{B}'|$. Then $|\mathcal{B}| \leq (1 + c/\sqrt{k}) \cdot |\mathcal{R}|$, where c is a constant.*

Lemma 2 and Lemma 3 together imply that: given any parameter ϵ , by setting $k = c^2 \epsilon^{-2}$, we can achieve a $(1 + \epsilon)$ -approximation for the algorithm in Table 2. \square

When $\mathcal{R} \cap \mathcal{B} \neq \emptyset$, let $\mathcal{I} = \mathcal{R} \cap \mathcal{B}$, $\mathcal{D}' = \mathcal{D} \setminus \mathcal{I}$, $\mathcal{R}' = \mathcal{R} \setminus \mathcal{I}$, $\mathcal{B}' = \mathcal{B} \setminus \mathcal{I}$. Let \mathcal{P}' be the node set that are not covered by \mathcal{I} . Clearly, \mathcal{B}' and \mathcal{R}' are disjoint; at the same time, \mathcal{R}' is an optimal solution for the instance of MDC with the disk set as \mathcal{D}' and node set as \mathcal{P}' . If we can prove that $|\mathcal{B}'|$ is approximately equal to $|\mathcal{R}'|$, it is straightforward that $|\mathcal{B}|$ is approximately equal to $|\mathcal{R}|$.

3.3 Establish the Locality Condition

We will construct a planar bipartite graph on the disk sets \mathcal{R}, \mathcal{B} that satisfy the locality condition. Remember that \mathcal{B} is the solution returned by the proposed algorithm shown in Table 2, and \mathcal{R} is an optimal solution ($|\mathcal{R}| \geq 2$).

3.3.1 Construct the planar graph

Similar to [10], we will use the dual of a weighted Voronoi diagram as the planar graph.

Let us introduce a concept of *weight function* $\omega(p, D)$ with two parameters: a point p and a disk D :

$$\omega(p, D) = \|p, d\| - r_D$$

where r_D is the radius of D ; and $\|p, d\|$ is the Euclidean distance between p and the center d of disk D . Intuitively, $\omega(p, D)$ is the Euclidean distance from p to the boundary of D . Figure 1 gives an illustration for the weight function.

For a disk $D \in \mathcal{R} \cup \mathcal{B}$, we define a *cell* (denoted as $\text{cell}(D)$) as the set of points p in the plane such that

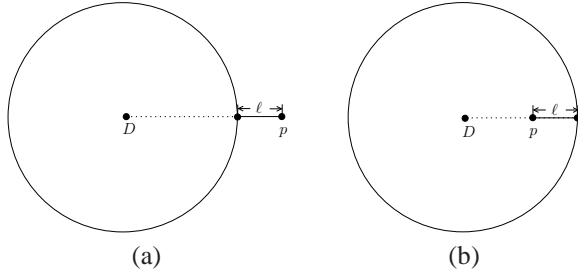


Figure 1: Illustration for weight function: (a) when p is not in D , $\omega(p, D) = \ell$; (b) when p is in D , $\omega(p, D) = -\ell$.

$\omega(p, D) \leq \omega(p, D'), \forall D' \in \mathcal{D}$. Then, the cells of all disks from \mathcal{D} induce a decomposition of the plane, which is the weighted Voronoi diagram. Based on the Voronoi diagram, we can construct a planar bipartite graph, similar to [10].

LEMMA 4. [10] *By using the dual of a weighted Voronoi diagram, we can construct a planar bipartite graph on disk set \mathcal{D} where an edge exists for any pair of disks R and B if $\text{cell}(R)$ and $\text{cell}(B)$ share a common point.*

Note that, to construct the planar graph a precondition exists that no more than three cells share a common boundary point in the weighted Voronoi diagram. If the precondition does not hold, we can not construct a planar graph directly for the Voronoi diagram such that there is an edge connecting every pair of disks when their corresponding cells share a common point. In this case (four or more disks share a common point x in a weighted Voronoi diagram), we call any four among those disks as a *degenerate quadruple*, and the point x as a *degenerate point*. Figure 2 gives two different cases of degenerate quadruples for the weighted Voronoi diagram. To the best of our knowledge, all methods for constructing planar graphs based on (weighted) Voronoi diagram suffer such degenerate issue, we are the first to provide a effective *patch* (in Subsection 3.4).

3.3.2 Verify the locality condition

We next prove that the planar graph constructed on $\mathcal{R} \cup \mathcal{B}$ satisfies the locality condition, *i.e.*, for each node $p \in \mathcal{P}$, there is an edge in the planar graph, that connects a disk $R \in \mathcal{R}$ and a disk $B \in \mathcal{B}$, and both disks cover p . The proof will extensively use a property which is described as follows:

CLAIM 1. *Given a node p and two disks R and B satisfying $\omega(p, R) \leq \omega(p, B)$, if B covers p , then R covers p .*

PROOF. Note that for any point p , and any disk D , if disk D covers the point p , then $\omega(p, D) \leq 0$, else (D does not covers p), $\omega(p, D) > 0$. If B covers p , then $\omega(p, R) \leq \omega(p, B) \leq 0$, which means that R covers p as well. \square

We will call the disks in \mathcal{R} and \mathcal{B} as red and blue disks respectively. Then, the main lemma to establish the locality condition is as follows:

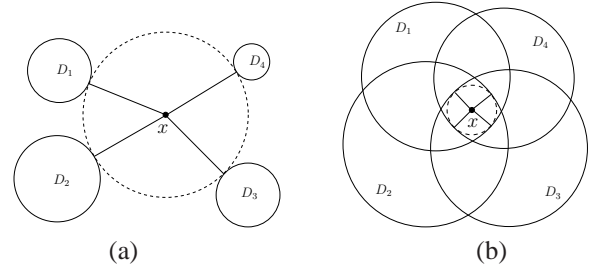


Figure 2: Degenerate quadruples: (a) the four corresponding cells for four disks D_1, D_2, D_3, D_4 share a common point x and x lies outside of these four cells; (b) the four corresponding cells for four disks D_1, D_2, D_3, D_4 share a common point x and x lies inside of these four cells.

LEMMA 5. *There exists a planar bipartite graph constructed on $\mathcal{R} \cup \mathcal{B}$ such that for each node $p \in \mathcal{P}$, there exists an edge connecting a red disk and a blue disk, both of which cover p .*

PROOF. Consider the planar bipartite graph G constructed on $\mathcal{R} \cup \mathcal{B}$ (in Lemma 4), for a node p , without loss of generality, assume $p \in \text{cell}(R)$ for some red disk $R \in \mathcal{R}$. Then, R must cover p . Otherwise, for any disk $D' \in \mathcal{R} \cup \mathcal{B}$, $\omega(p, D') \geq \omega(p, R)$, by Claim 1, D' does not cover p , which contradicts the fact that $\mathcal{R} \cup \mathcal{B}$ cover p .

Let B denote a closest blue disk to p , that is $\omega(p, B) \leq \omega(p, B') : \forall B' \in \mathcal{B}$. Note that B must cover p . Otherwise, by Claim 1, no other blue disks would cover p ; this would contradict the fact that \mathcal{B} cover p . Based on the arithmetic relation between $\omega(p, B)$ and $\omega(p, R)$, we have two cases.

Case (1) If $\omega(p, B) = \omega(p, R)$, then $p \in \text{cell}(B)$ and B covers p . Since $\text{cell}(R)$ and $\text{cell}(B)$ share a common point p , By Lemma 4, there exists an edge connecting R and B in the planar graph G , both of which cover the node p .

Case (2) $\omega(p, B) > \omega(p, R)$. We will walk from p to b (the center of disk B) along the straight line segment \overrightarrow{pb} . During this walk, we may cross some red cells and blue cells, and at some point before reaching b we will enter $\text{cell}(B)$ the first time. Let x be the point at which we first enter

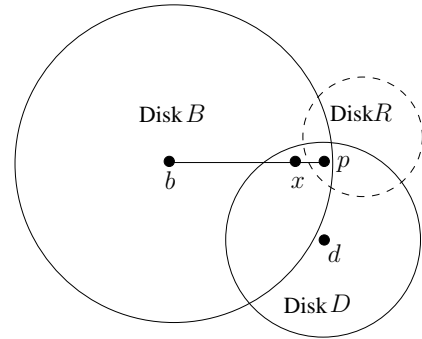


Figure 3: Illustration for Case (2) in the proof of Lemma 5.

$\text{cell}(B)$. We must enter this cell from another cell, we assume the cell is $\text{cell}(D)$ and the center of disk D is d . Note that $D \in \mathcal{R} \cup \mathcal{B}$ may be either blue or red, and $\text{cell}(D)$ shares a common point x with $\text{cell}(B)$.

Next, we will argue that D covers p and D is red.

If $D = R$, we are done. Otherwise, we have $\|p, d\| < \|p, x\| + \|x, d\| \Rightarrow \omega(p, D) < \|p, x\| + \omega(x, D) = \|p, x\| + \omega(x, B) = \omega(p, B)$. Thus, by Claim 1, D covers p , moreover, D is red, otherwise B would not be the closest blue disk to p , which causes contradiction.

To sum up, $\text{cell}(B)$ and $\text{cell}(D)$ share a common point, which implies that an edge between B and D exists in the planar graph. Moreover, B is blue, D is red, and both disks cover p . This completes the proof. \square

3.4 Patch for Degenerate Issue

For a given set of nodes \mathcal{P} , let us define the *threshold* for a disk D as $\beta(D) = \min_{p \in \mathcal{P}} \{\omega(p, D) : \omega(p, D) > 0\}$. Note that, (1) $\beta(D) > 0$ and (2) when we increase the radius of disk D by a value smaller than $\beta(D)$, the set of nodes covered by D remain the same. In other words, if a node $p \in \mathcal{P}$ is covered before this scaling operation, then p is covered by D after the operation, and vice versa. Thus, the scaling will not affect the problem MDC intrinsically.

We then develop a *scaling process* (by scaling the disks in degenerate quadruples) to eliminate degenerate quadruples.

THEOREM 2. *Given a node set \mathcal{P} and a disk set \mathcal{D} , we can remove all degenerate quadruples by scaling disks such that for each disk, the set of nodes covered will not be changed.*

PROOF. We will iteratively remove degenerate quadruples. During each iteration, we arbitrarily select a degenerate quadruple (D_1, D_2, D_3, D_4) , assume their corresponding cells share a common point x . Note that simply increasing the radius of only disk D_1 by any value ℓ for $0 < \ell < \beta(D_1)$ can result in the fact: $x \notin \text{cell}(D_2)$ (at the same time, $x \notin \text{cell}(D_3)$ and $x \notin \text{cell}(D_4)$), thus, we can remove such a degenerate quadruple easily. The challenge is that, we have to guarantee that no new degenerate quadruple is introduced by this scaling operation.

We call a value ℓ as a *critical value* for D_1 if increasing the radius of D_1 by exactly ℓ will introduce a new degenerate quadruple. We then upper-bound the number of critical values for each disk as follows:

CLAIM 2. *There are at most $O(n^3)$ critical values for the disk D_1 (The proof is available in the appendix).*

Based on Claim 2, there exist at most $O(n^3)$ critical values for the disk D_1 . We can select a value that is not equal to any of these critical values, and at the same time, is strictly smaller than the threshold $\beta(D_1)$ for then disk D_1 . We then increase the radius of disk D_1 by exactly this value, clearly, we can guarantee that this scaling operation can eliminate a degenerate quadruple, and will not introduce any new degenerate quadruple, at the same time.

Observe that due to the recursive nature of the scaling process, we must prove that the process will terminate in a finite number of iterations. We define a **Potential Function** as the number of degenerate quadruples. Clearly, the initial cardinality of the set \mathcal{C} is at most $\binom{n}{4} = O(n^4)$. After one iteration, at least one degenerate quadruple is removed by scaling only one disk; at the same time, the scaling will not introduce any new degenerate quadruple. Thus, the potential function will be decreased by at least one after each iteration, which means that the scaling process will terminate within at most $O(n^4)$ iterations. \square

4. MIN-COST DISK COVER

4.1 Algorithm Design

Given an instance of the problem MCDC: a node set \mathcal{P} and a disk set \mathcal{D} that can cover \mathcal{P} , we can formulate the optimization problem as a Linear Programming (LP). For any node $p \in \mathcal{P}$, let $\mathcal{D}_p \subseteq \mathcal{D}$ denote the subset of disks covering p . Let $x_D \in \{0, 1\}$ for any disk $D \in \mathcal{D}$ indicate whether the disk D is selected in the solution or not, we relax the requirement, such that $x_D : D \in \mathcal{D}$ can be any value in $[0, 1]$, instead of only integers. We then consider the following LP relaxation for the problem MCDC. We assume the cost function for \mathcal{D} is $c : \mathcal{D} \mapsto \mathbb{R}^+$:

$$\begin{aligned} \min : & \sum_{D \in \mathcal{D}} c(D) \cdot x_D, & \text{s.t. :} \\ & \sum_{D \in \mathcal{D}_p} x_D \geq 1, & \forall p \in \mathcal{P} \\ & x_D \geq 0, & \forall D \in \mathcal{D} \end{aligned} \quad (1)$$

After solving the LP relaxation, we obtain an optimal solution $\{x_D : D \in \mathcal{D}\}$, and then create a set \mathcal{D}_0 of disks as follows: for each disk D , we add $\lfloor 2n \cdot x_D \rfloor$ copies of D to \mathcal{D}_0 where n is the cardinality of \mathcal{D} (note that for the special case when $x_D < \frac{1}{2n}$, $\lfloor 2n \cdot x_D \rfloor = 0$, we do not add any copy of D to \mathcal{D}_0); each copy of D inherits its original cost. For the disk set \mathcal{D}_0 , we observe two important facts (given in Lemma 6).

Next, we iteratively apply the *uniform sampling process* (in Table 4) to produce a successively sparse disk cover (after probabilistically removing some disks from the disk cover). For the first iteration, we set the input disk cover as \mathcal{D}_0 , and the parameter $L_1 = n$. For the i -th iteration, we set the input disk cover as \mathcal{D}_{i-1} , and the parameter $L_i = \log L_{i-1}$, to obtain an output disk set $\mathcal{D}_i \subseteq \mathcal{D}_{i-1}$ which is a sparse disk cover, for $i = 2, 3, \dots, t$ ($t = \log^* n$). Finally, we output \mathcal{D}_t . The details are shown in Table 3 (also in Figure 4).

4.1.1 Uniform Sampling Process

The *uniform sampling process* is a probabilistic algorithm that takes an input disk cover \mathcal{D} of the node set \mathcal{P} with a parameter L , and outputs a *sparse* disk cover \mathcal{D}' where the probability of each disk being selected is at most $\frac{c \log L}{L}$. In

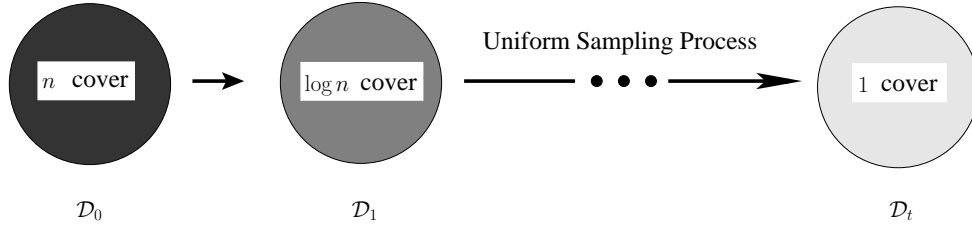


Figure 4: Illustration for Algorithm 3, the various shades reflect the sparsities of disk covers.

LP-based Algorithm:
Input: a disk set \mathcal{D} , a node set \mathcal{P} , $c : \mathcal{D} \mapsto \mathbb{R}^+$; Solve the LP relaxation in Equation (1), let $\{x_D : D \in \mathcal{D}\}$ be the output; for each disk $D \in \mathcal{D}$ add $\lfloor 2n \cdot x_D \rfloor$ copies of D with the same cost to \mathcal{D}_0 ; $t \leftarrow \log^* n$; for $i = 0$ to $t - 1$ apply uniform sampling process (Table 4) on \mathcal{D}_i ; let \mathcal{D}_{i+1} be the output; $i++$; delete the redundant disks from \mathcal{D}_t ; return \mathcal{D}_t .

Table 3: LP-based Algorithm

Uniform Sampling Process:
Input: \mathcal{D} , \mathcal{P} , a parameter L ; Construct the sequence $\sigma \leftarrow \langle D_1, \dots, D_m \rangle$; for $i = 1$ to m for each node $v \in \mathcal{P}$ covered by D_i if D_i is forced because of v add D_i to \mathcal{D}' ; else add D_i with probability $\frac{c \log L}{L}$ to \mathcal{D}' ; $i++$; return \mathcal{D}' .

Table 4: Uniform Sampling Process

addition, each node p that is L -covered by \mathcal{D} is at least $\log L$ -covered by \mathcal{D}' . Here we say a node p is L -covered if exactly L disks from \mathcal{D} cover p . In such case, the node p is also said to be covered with the *multiplicity* L .

We first only consider the subset \mathcal{P}' of nodes that are covered by \mathcal{D} with the multiplicity in $[L, 2L]$ (each node from \mathcal{P}' is covered by at least L disks, and at most $2L$ disks from \mathcal{D}), we will produce a subset $\mathcal{D}' \subseteq \mathcal{D}$ of disks, such that \mathcal{P}' will be at least $\log L$ -covered in \mathcal{D}' . We repeat the process for the nodes that are covered by \mathcal{D} with the multiplicity in $[2L, 4L]$, $[4L, 8L]$ and so on, the output is a series of sparse disk sets. We finally output the union of all output disks. We will prove that the probability of a disk being selected can still upper bounded by $O\left(\frac{c \log L}{L}\right)$.

Given a disk set \mathcal{D} and a node set \mathcal{P} , we define an *equivalence class* for \mathcal{P} as a subset of all nodes from \mathcal{P} that are covered by the same set of disks from \mathcal{D} , i.e., if two nodes are in the same equivalence class, then one node is covered by a disk from \mathcal{D} implies immediately that the other node is also covered by the same disk. Then, all the equivalence classes for \mathcal{P} induce a partition of \mathcal{P} . Note that, if a set \mathcal{D}' at least $\log L$ -cover one node in an equivalence class, then they at least $\log L$ -cover all nodes in that class. Thus, we can assume we have one representative node (can be any node from this class) from each class. We want to at least $\log L$ -cover these representative nodes.

Let $N_m = \mathcal{D}$, and let C_m denote the set of *equivalence classes* of nodes such that the nodes in each class is covered with multiplicity at most $2L$. By Lemma 7, $|C_m| \leq c' n_m L^2$, $n_m = |N_m|$. We compute a disk $D_m \in N_m$ that covers the least number of representative nodes. By pigeon-hole principle, D_m covers at most $2c' L^3$ classes of C_m . We will recursively compute a sequence of disks for a new instance for $N_{m-1} = N_m \setminus \{D_m\}$, and append the sequence to D_m . In the new instance for N_{m-1} , we consider the classes C_{m-1} whose coverage multiplicity in N_{m-1} is at most $2L$. Let σ be the reverse of this sequence:

$$\sigma = \langle D_1, \dots, D_m \rangle.$$

Note that the method for constructing the sequence σ is similar to *smallest last ordering* [16].

Based on the sequence σ , for each disk D_j , we make an instant decision on the addition of \mathcal{D}' , depending on whether D_j is forced or not. Here we call a disk $D_j \in N_j$ forced if not including D_j will result in a consequence that some node $p \in \mathcal{P}'$ can not be at least $\log L$ -covered. The details are shown in Table 4.

4.2 Approximation Ratio

First, we introduce two facts for the disk set \mathcal{D}_0 :

LEMMA 6. *The following facts are true:*

1. Each node p is at least n -covered by \mathcal{D}_0 ;
2. $w(\mathcal{D}_0) \leq 2n \cdot \lambda^*$, where λ^* is the optimal objective function value for the LP-relaxation in Equation (1).

PROOF. The proof is available in the appendix. \square

Next, we will prove that the algorithm described in Table 4

has the uniform sampling property. We will make use of the following lemma:

LEMMA 7. *Let \mathcal{D} and \mathcal{P} be a set of m disks and a set of nodes respectively, and $1 \leq L \leq m$ be an integer. There are $O(mL^2)$ nodes of \mathcal{P} covered by distinct subsets of \mathcal{D} , each of size at most L .*

Based on Lemma 7, we can prove Theorem 3, which is a variant of the result in [10, 22].

THEOREM 3. (Uniform Sampling Property) *Given a node set \mathcal{P} and a disk set \mathcal{D} , assume \mathcal{D} cover \mathcal{P} , the uniform sampling process (Table 4) produces a subset $\mathcal{D}' \subset \mathcal{D}$, such that for any node p , if p is L -covered in \mathcal{D} , then p is at least $\log L$ -covered in \mathcal{D}' and $\Pr(D \in \mathcal{D}') \leq \frac{c \log L}{L}$.*

PROOF. The proof is available in the appendix. \square

By combining Theorem 3 with Lemma 6, we can obtain the following main theorem easily.

THEOREM 4. *For the problem MCDC, there exists a randomized algorithm that produces a disk cover \mathcal{D}_t , and $c(\mathcal{D}_t) \leq 2^{O(\log^* n)} \cdot c^*$, with high probability, where c^* denotes the cost of an optimal solution, and $\log^* n$ is the smallest number of iterated “logarithms” applied to n to yield a constant.*

The proof of Theorem 4 is similar to Section 3.1 in [10].

5. MAX-LIFE WIRELESS COVERAGE

The problem **MLWC** can be formulated as the following linear program (LP):

$$\begin{aligned} \max \quad & \sum_{C \in \mathcal{C}} x_C \\ \text{s.t.} \quad & \sum_{C \in \mathcal{C}} x_C p_C(u) \leq b(u), \forall u \in V; \\ & x_C \geq 0, \forall C \in \mathcal{C}. \end{aligned} \quad (2)$$

This LP has $|V|$ constraints (excluding the trivial constraints $x_C \geq 0, \forall C \in \mathcal{C}$), and consequently there always exists an optimal solution using at most n wireless covers. However, since the number of variables $|\mathcal{C}|$ is prohibitively large (exponential in the number of nodes), standard LP solvers are not practical for solving this packing LP.

5.1 Min-Cost Wireless Cover

We first introduce a related problem. Given a price function $y \in \mathbb{R}_+^V$, the cost of a wireless cover C with respect to (*w.r.t.*) y is defined as $\sum_{u \in C} y(u) p(u)$. Note that if $y \cdot p$ is an all-one vector ($y(u) \cdot p(u) = 1$ for every node u), the cost of C *w.r.t.* to y is exactly $|C|$. For any wireless coverage task, given implicitly by a collection \mathcal{C} of wireless covers and a price function $y \in \mathbb{R}_+^V$, the problem **Min-Cost Wireless Cover (MCC)** seeks a wireless cover $C \in \mathcal{C}$ of minimum cost.

By mapping each wireless node to a disk with its covering range as the disk radius, we can reduce the problem MCC to the problems MCDC and MDC (when each wireless node has the same cost). Thus, we have the following lemma:

LEMMA 8. *For the problem MCC, we have:*

1. *There is a $2^{O(\log^* n)}$ -approximation algorithm.*
2. *When $p \cdot y$ is an all-one vector, there is an $(1 + \epsilon)$ -approximation algorithm.*

Next, we show that these two problems **MLWC** and **MCC** are intrinsically related to each other. We refer to the LP in equation (2) as the primal LP. The dual to this primal LP associates a price $y(u)$ for each node $u \in V$:

$$\begin{aligned} \min \quad & \sum_{u \in V} b(u) y(u) \\ \text{s.t.} \quad & \sum_{u \in V} p_C(u) y(u) \geq 1, \forall C \in \mathcal{C} \\ & y(u) \geq 0, \forall u \in V \end{aligned} \quad (3)$$

The above LP is referred to as dual LP, and the minimization problem defined by this dual LP is referred to as the dual of **MLWC**, which can be interpreted as follows. For any price function $y \in \mathbb{R}_+^V$, let

$$\begin{aligned} \alpha(y) &= \min_{C \in \mathcal{C}} \sum_{u \in V} p_C(u) y(u), \\ \beta(y) &= \sum_{u \in V} b(u) y(u). \end{aligned}$$

Then, $\alpha(y)$ is the cost of a min-cost wireless cover in \mathcal{C} *w.r.t.* y , and $\beta(y)$ is the total energy cost *w.r.t.* y . Thus, the dual of **MLWC** is equivalent to finding a price function $y \in \mathbb{R}_+^V$ such that $\beta(y)$ is minimized subject to $\alpha(y) \geq 1$. Let opt be the life of a max-life covering schedule. Then, we have the following relation

LEMMA 9. *For any $y \in \mathbb{R}_+^V$, $\alpha(y) \leq \frac{\beta(y)}{opt}$. In addition, there exists some $y \in \mathbb{R}_+^V$ such that $\alpha(y) = \frac{\beta(y)}{opt}$.*

PROOF. We begin with the first part. The first part holds trivially if $\alpha(y) = 0$. So, we assume that $\alpha(y) > 0$. Let $y' = \frac{y}{\alpha(y)}$. Then, y' is a feasible solution of the dual LP. Hence

$$opt \leq \beta(y') = \frac{\beta(y)}{\alpha(y)}.$$

Therefore, $\alpha(y) \leq \frac{\beta(y)}{opt}$. Next, we prove the second part. Suppose y is an optimal solution to dual LP. Then, $\alpha(y) = 1$ for otherwise, we can scale y down strictly to get a better solution, which causes contradiction. Thus, $opt = \beta(y)$. For such y , we have $\alpha(y) = \frac{\beta(y)}{opt}$. \square

Further relations between **MLWC** and **MCC** will be explored subsequently.

5.2 Ellipsoid Algorithm

The *ellipsoid method* is an ingenious approach for deciding whether a given non-degenerate convex set is empty (a convex set in n -dimensional space is said to be non-degenerate if it is either empty or has a volume at least ν for some $\nu > 0$ which depends only on n).

The idea of the ellipsoid method is very roughly the following. Consider a non-degenerate convex set S . We would like to test whether S is empty or not. We start with an ellipsoid which is guaranteed to contain S . At each iteration, we check whether the center of the current ellipsoid is in S or not. If yes, we can conclude that S is nonempty and stop. Otherwise, we take a hyperplane through the center such that S is contained in one of the two half-ellipsoids separated by this hyperplane. We take the smallest ellipsoid completely containing this half-ellipsoid, whose volume is substantially smaller than the volume of the previous ellipsoid. We iterate on this new ellipsoid. In the worst case, we need to iterate until the volume of the bounding ellipsoid gets below ν , in which case we can conclude that S is empty. It turns out that only a polynomial number of iterations are required in the case of linear programming. The algorithm does not require an explicit description of the linear program. All that is required is a polynomial time *separating oracle*, which checks whether a point lies in S or not, and returns a separating hyperplane in the latter case.

We will leverage the ellipsoid method for exponential-sized linear program (LP) with an (approximate) separation oracle to prove the following general theorem.

THEOREM 5. *Suppose that \mathcal{N} is a network class satisfying that there is a polynomial (respectively, a polynomial μ -approximation) algorithm for **MCC** for a wireless coverage task restricted to \mathcal{N} . Then, there is a polynomial (respectively, a polynomial μ -approximation) algorithm for **MLWC** for the same coverage task restricted to \mathcal{N} .*

PROOF. Let \mathcal{A} be a μ -approximation algorithm for **MCC** for a wireless coverage task restricted to \mathcal{N} (For simplicity of presentation, we treat a polynomial algorithm which produces an optimal solution as a polynomial 1-approximation algorithm). Consider a network $D = (V, A; c) \in \mathcal{N}$. We use \mathcal{C} to denote the collection of wireless covers for a coverage task. We run the ellipsoid algorithm on the dual LP using the algorithm \mathcal{A} as the approximate separation oracle. More precisely, let $S(L)$ denote the set of $y \in \mathbb{R}_+^V$ satisfying that

$$\begin{aligned} \sum_{u \in V} b(u) y(u) &\leq L, \\ \sum_{u \in C} p_C(u) y(u) &\geq 1, \forall C \in \mathcal{C}. \end{aligned}$$

We use binary search to find the smallest value of L for which $S(L)$ is nonempty. The separation oracle acts as follows: First, it checks the inequality $\sum_{u \in V} b(u) y(u) \leq L$. Next, it runs the algorithm \mathcal{A} to compute a wireless cover

$C \in \mathcal{C}$, using y as the price function. If C has power cost less than one, then we know that $y \notin S(L)$, and C gives us a separating hyperplane. If C has power cost at least one, then we accept $y \in S(L)$. Of course, since \mathcal{A} is just an approximation algorithm, the above conclusion might be incorrect, and $S(L)$ might actually be empty. However, since the approximation factor of \mathcal{A} is at most μ , we know that in this case, $\mu \cdot y \in S(\mu \cdot L)$. Therefore, if L^* is the minimum value of L for which the algorithm decides $S(L)$ is nonempty, then we know that $S(L^* - \varepsilon)$ is empty (where ε depends on the precision of the algorithm), and $S(\mu L^*)$ is nonempty. Therefore, the value of the dual LP, and hence, the value of the primal LP, is between L^* and μL^* .

The above algorithm computes the approximate value of the primal LP. Next, we describe how to compute the actual approximate solution. Now, let \mathcal{C}' denote the subset of wireless covers in \mathcal{C} corresponding to the separating hyper-planes found by the above separation oracle while running the ellipsoid algorithm on $S(L^* - \varepsilon)$. Then, \mathcal{C}' is of polynomial cardinality. Consider the restricted dual LP:

$$\begin{aligned} \min \quad & \sum_{u \in V} b(u) y(u) \\ \text{s.t.} \quad & \sum_{u \in C} p_C(u) y(u) \geq 1, \forall C \in \mathcal{C}' \\ & y(u) \geq 0, \forall u \in V \end{aligned}$$

Its value is also at least L^* . So, we solve the following restricted primal LP of polynomial size, which is the dual of the restricted dual LP:

$$\begin{aligned} \max \quad & \sum_{C \in \mathcal{C}'} x_C \\ \text{s.t.} \quad & \sum_{C \in \mathcal{C}'} x_C p_C(u) \leq b(u), \forall u \in V \\ & x_C \geq 0, \forall C \in \mathcal{C}' \end{aligned}$$

The optimal solution of this restricted LP has value at least L^* , which is a μ -approximation to the original primal LP. \square

Theorem 5 immediately implies the following algorithmic results on **MLWC**:

1. There is an $2^{O(\log^* n)}$ -approximation algorithm for **MLWC**.
2. There is an $(1 + \epsilon)$ -approximation algorithm for **MLWC**, when $p \cdot y$ is an all-one vector.

We remark that the approximation algorithms presented in this section are of theoretical interest only. They characterize the approximation hardness of the optimization problems studied in this section. However, the ellipsoid method with the approximation separation oracles is practically quite infeasible [21]. In Section 5.3, we will develop practically feasible approximation algorithms.

5.3 Price-Directive Algorithm

In this section, we present an iterative algorithm called Price-Directive Algorithm (PDA) for a given wireless coverage task, which is adapted from the algorithm in [8] for fractional packing problems.

The basic idea is that by setting the prices of the nodes with low residue energy relatively higher, the nodes with low residue energy are protected from getting drained of energy quickly while the nodes with high residue energy are enforced to contribute more energy. The algorithm utilizes a μ -approximation algorithm \mathcal{A} for **MCC** for the same coverage task (if $\mu = 1$, the algorithm \mathcal{A} is optimal). A constant parameter $\varepsilon \in (0, 1)$ is also part of the input, and the output solution has an approximation bound of at most $(1 + \varepsilon)\mu$. The algorithm maintains the following variables:

- \mathcal{C} : the set of chosen wireless covers;
- x_C for each $C \in \mathcal{C}$: the duration of C ;
- $z \in \mathbb{R}_+^V$: the energy consumption percentage vector defined by

$$z(u) = \frac{\sum_{C \in \mathcal{C}} x_C p_C(u)}{b(u)}, \forall u \in V;$$

- $\phi = \max_{u \in V} z(u)$: the maximum energy consumption percentage;
- $y \in \mathbb{R}_+^V$: the price vector;
- β : the total energy cost $\sum_{u \in V} b(u) y(u)$.

Initially, \mathcal{C} is empty and the price $y(u)$ of each node u is the reverse of its energy budget $b(u)$. Accordingly, both z and ϕ are initialized to zero, and β is initialized to n accordingly. Each iteration first computes a wireless cover $C \in \mathcal{C}$ using an algorithm \mathcal{A} together with the current price vector y . A node v is said to be a *bottleneck* if $b(v)/p_C(v)$ is the smallest among all nodes. Let v be the bottleneck node and set $t = b(v)/p_C(v)$. The duration x_C of C is increased by t , and both z and ϕ are updated accordingly. After that, the price vector y is reset properly and the variable β is updated accordingly. The stopping rule is that $0 < \phi \leq \frac{1+\varepsilon}{\varepsilon} \ln \frac{\beta}{n}$. Finally, each x_C for $C \in \mathcal{C}$ is scaled down by a factor of ϕ to obtain a feasible solution. The outline of the algorithm is described in Table 5.

The next theorem gives both the running time and approximation ratio of the algorithm **PDA**.

THEOREM 6. *The algorithm **PDA** produces an $(1 + \varepsilon)\mu$ -approximation in at most $n \left\lceil \frac{(1+\varepsilon) \ln n}{(1+\varepsilon) \ln(1+\varepsilon) - \varepsilon} \right\rceil$ iterations.*

Theorem 6 immediately implies the approximation bounds of **PDA** (described in Table 5).

1. There is an $(1 + \varepsilon) \cdot 2^{O(\log^* n)}$ -approximation algorithm for MLWC.
2. There is a PTAS for MLWC, when $p \cdot y$ is an all-one vector.

Price-Directive Algorithm (PDA):

```

 $\mathcal{C} \leftarrow \emptyset; \forall u \in V, z(u) \leftarrow 0; \phi \leftarrow 0;$ 
 $\forall u \in V, y(u) \leftarrow \frac{1}{b(u)}; \beta \leftarrow n;$ 
repeat
  compute an  $C \in \mathcal{C}$  using  $\mathcal{A}$  on  $(D, y)$ ;
   $t \leftarrow \min_{v \in V} b(v)/p_C(v);$ 
  if  $C \in \mathcal{C}$  then  $x_C \leftarrow x_C + t,$ 
    else  $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}, x_C \leftarrow t;$ 
   $\forall u \in V, z(u) \leftarrow z(u) + t \frac{p_C(u)}{b(u)};$ 
   $\phi \leftarrow \max_{u \in V} z(u);$ 
   $\forall u \in V, y(u) \leftarrow y(u) \left(1 + \varepsilon t \frac{p_C(u)}{b(u)}\right);$ 
   $\beta \leftarrow \sum_{u \in V} b(u) y(u);$ 
until  $0 < \phi \leq \frac{1+\varepsilon}{\varepsilon} \ln \frac{\beta}{n};$ 
return  $\{(C, x_C/\phi) : C \in \mathcal{C}\}.$ 

```

Table 5: Price-Directive Algorithm

The rest of this section is devoted to the proof of the Theorem 6. We first introduce the following algebraic inequality.

LEMMA 10. $\forall \varepsilon > 0$ and $0 \leq t \leq 1, t \leq \log_{1+\varepsilon}(1 + \varepsilon t)$.

Let $\mathcal{C}_0, z_0, \phi_0, y_0$ and β_0 be the initial values of \mathcal{C}, z, ϕ, y and β respectively. For each $j \geq 1$, let $\mathcal{C}_j, z_j, \phi_j, y_j$ and β_j be the values of \mathcal{C}, z, ϕ, y and β respectively at the end of the j -th iteration. In addition, for each $j \geq 1$, let

$$\tau_j = \max_{u \in V} y_j(u) b(u).$$

We first claim that

$$\phi_j \leq \log_{1+\varepsilon} \tau_j.$$

Indeed, for each $j \geq 1$ and each $u \in V$, by Lemma 10,

$$\begin{aligned} z_j(u) - z_{j-1}(u) &\leq \log_{1+\varepsilon}(1 + \varepsilon(z_j(u) - z_{j-1}(u))) \\ &= \log_{1+\varepsilon} \frac{y_j(u)}{y_{j-1}(u)}, \end{aligned}$$

which implies $z_j(u) \leq \log_{1+\varepsilon} \frac{y_j(u)}{y_0(u)} = \log_{1+\varepsilon}(y_j(u) b(u))$. Hence, the claim holds.

Now, let $K = n \left\lceil \frac{(1+\varepsilon) \ln n}{(1+\varepsilon) \ln(1+\varepsilon) - \varepsilon} \right\rceil$ and we prove that the number of iterations is at most K . Assume to the contrary that the number of iterations is greater than K . Then, at least one node v appears as a bottleneck node in at least K/n iterations among the first K iterations. For such node v , $y(v)$ is increased by a factor of $1 + \varepsilon$ at the end of each iteration when v is a bottleneck node. Therefore,

$$y_K(v) \geq y_0(v) (1 + \varepsilon)^{K/n} = \frac{(1 + \varepsilon)^{K/n}}{b(v)},$$

which implies

$$\tau_K \geq y_K(v) b(v) \geq (1 + \varepsilon)^{K/n}.$$

Hence,

$$\begin{aligned}
\frac{\phi_K}{\ln \frac{\beta_K}{n}} &\leq \frac{\log_{1+\varepsilon} \tau_K}{\ln \frac{\tau_K}{n}} = \frac{1}{\ln(1+\varepsilon) - \frac{\ln n}{\log_{1+\varepsilon} \tau_K}} \\
&\leq \frac{1}{\ln(1+\varepsilon) - \frac{\ln n}{\frac{n}{K}}} \\
&= \frac{1}{\ln(1+\varepsilon) - \frac{\ln n}{\sqrt{\frac{(1+\varepsilon)\ln n}{(1+\varepsilon)\ln(1+\varepsilon)-\varepsilon}}}} \\
&\leq \frac{1}{\ln(1+\varepsilon) - \frac{\ln n}{\frac{(1+\varepsilon)\ln n}{(1+\varepsilon)\ln(1+\varepsilon)-\varepsilon}}} \\
&= \frac{1}{\ln(1+\varepsilon) - \frac{(1+\varepsilon)\ln(1+\varepsilon)-\varepsilon}{1+\varepsilon}} \\
&= \frac{1+\varepsilon}{\varepsilon}.
\end{aligned}$$

By the stopping rule, the number of iterations is at most K , which is a contradiction.

From now on, we assume that the number of iterations is k . It's easy to prove by induction on j that by the end of the j -th iteration for $1 \leq j \leq k$, for each node $u \in V$,

$$z_j(u) = \frac{\sum_{C \in \mathcal{C}_j} x_C p_C(u)}{b(u)}, \forall u \in V.$$

In other words, $z_j(u)$ is indeed the energy consumption percentage of u by the end of the j -th iteration. Therefore, the scaling by a factor ϕ_k in the scaling phase results in a feasible solution.

Next, we show that the approximation bound of the output solution is $(1+\varepsilon)$. For each $1 \leq j \leq k$, let t_j be the value of t computed in the j -th iteration. Then, by the end of the j -th iteration for $1 \leq j \leq k$, we have

$$\sum_{C \in \mathcal{C}_j} x_C = \sum_{i=1}^j t_i.$$

So, the life of the output solution is

$$\sum_{C \in \mathcal{C}_k} \frac{x_C}{\phi_k} = \frac{\sum_{C \in \mathcal{C}_k} x_C}{\phi_k} = \frac{\sum_{j=1}^k t_j}{\phi_k}.$$

Let opt be the life of an optimal solution. We claim that for each $1 \leq j \leq k$,

$$t_j \geq \frac{1}{\varepsilon \mu} \frac{\beta_j - \beta_{j-1}}{\beta_{j-1}} opt.$$

Indeed, let C_j be the wireless cover computed in the j -th iteration. By Lemma 9,

$$\sum_{u \in V} p_{C_j}(u) y_{j-1}(u) \leq \mu \frac{\beta_{j-1}}{opt}.$$

Thus,

$$\begin{aligned}
\beta_j &= \sum_{u \in V} b(u) y_j(u) \\
&= \sum_{u \in V} b(u) y_{j-1}(u) \left(1 + \varepsilon t_j \frac{p_{C_j}(u)}{b(u)}\right) \\
&= \sum_{u \in V} b(u) y_{j-1}(u) + \varepsilon t_j \left(\sum_{u \in V} p_{C_j}(u) y_{j-1}(u)\right) \\
&= \beta_{j-1} + \varepsilon t_j \left(\sum_{u \in V} p_{C_j}(u) y_{j-1}(u)\right) \\
&\leq \beta_{j-1} + \varepsilon t_j \cdot \mu \frac{\beta_{j-1}}{opt},
\end{aligned}$$

from which the claim follows. Therefore,

$$\sum_{j=1}^k t_j \geq \frac{opt}{\varepsilon \mu} \sum_{j=1}^k \frac{\beta_j - \beta_{j-1}}{\beta_{j-1}} \geq \frac{opt}{\varepsilon \mu} \ln \frac{\beta_k}{\beta_0} = \frac{opt}{\varepsilon \mu} \ln \frac{\beta_k}{n},$$

So,

$$\frac{\sum_{j=1}^k t_j}{\phi_k} \geq \frac{1}{\varepsilon \mu} \frac{\ln \frac{\beta_k}{n}}{\phi_k} opt \geq \frac{1}{\varepsilon \mu} \frac{\varepsilon}{1+\varepsilon} opt = \frac{opt}{(1+\varepsilon)\mu},$$

where the second inequality follows from the stopping rule.

6. CONCLUSION

Many challenging issues in wireless networks are intrinsically related to coverage. In this work, we have presented several approximation algorithms for three mutually related coverage problems: MDC, MCDC, MLWC. For the problem MDC, we are the first to achieve a PTAS when disks have arbitrary radii. For the problem MLWC, this is the first time in the literature that the problem of maximizing lifetime wireless coverage was formulated, and the approximation algorithmic results was obtained.

7. REFERENCES

- [1] ABRAMS, Z., GOEL, A., AND PLOTKIN, S. 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* (2004), ACM, pp. 424–432.
- [2] AMBUHL, C., ERLEBACH, T., MIHALAK, M., AND NUNKESSER, M. Constant-Factor Approximation for Minimum-Weight (Connected) Dominating Sets in Unit Disk Graphs. *Lecture Notes in Computer Science 4110* (2006), 3.
- [3] BRONNIMANN, H., AND GOODRICH, M. Almost optimal set covers in finite VC-dimension. *Discrete and Computational Geometry 14*, 1 (1995), 463–479.
- [4] CĂLINESCU, G., MANDOIU, I., WAN, P., AND ZELIKOVSKY, A. Selecting forwarding neighbors in wireless ad hoc networks. *Mobile Networks and Applications 9*, 2 (2004), 101–111.
- [5] CARMİ, P., KATZ, M., AND LEV-TOV, N. Covering points by unit disks of fixed location. *Algorithms and Computation* (2007), 644–655.
- [6] DAI, D., AND YU, C. A $5 + \varepsilon$ -approximation algorithm for minimum weighted dominating set in unit disk graph. *Theoretical Computer Science 410*, 8-10 (2009), 756–765.
- [7] ERLEBACH, T., AND MIHALÁK, M. A $(4 + \varepsilon)$ -Approximation for the Minimum-Weight Dominating Set Problem in Unit Disk Graphs. *Approximation and Online Algorithms* (2010), 135–146.

- [8] GARG, N., AND K@ARTICLEHOU2009COVERAGE, TITLE=COVERAGE IN WIRELESS SENSOR NETWORKS, AUTHOR=HOU, J.C. AND YAU, D.K.Y. AND MA, C.Y.T. AND YANG, Y. AND ZHANG, H. AND HOU, I.H. AND RAO, N.S.V. AND SHANKAR, M., JOURNAL=GUIDE TO WIRELESS SENSOR NETWORKS, PAGES=47–79, YEAR=2009, PUBLISHER=SPRINGER
- [9] GIBSON, M., AND PIRWANI, I. Algorithms for Dominating Set in Disk Graphs: Breaking the logn Barrier. *Algorithms–ESA 2010*, 243–254.
- [10] GIBSON, M., AND PIRWANI, I. Approximation algorithms for dominating set in disk graphs. *Arxiv preprint arXiv:1004.3320* (2010).
- [11] GONZALEZ, T. Covering a set of points in multidimensional space* 1. *Information processing letters* 40, 4 (1991), 181–188.
- [12] HOCHBAUM, D., AND MAASS, W. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)* 32, 1 (1985), 130–136.
- [13] HOU, J., YAU, D., MA, C., YANG, Y., ZHANG, H., HOU, I., RAO, N., AND SHANKAR, M. Coverage in wireless sensor networks. *Guide to Wireless Sensor Networks* (2009), 47–79.
- [14] HUANG, Y., GAO, X., ZHANG, Z., AND WU, W. A better constant-factor approximation for weighted dominating set in unit disk graph. *J. Comb. Optim* 1382 (2008), 6905.
- [15] JOHNSON, D. The NP-completeness column: an ongoing guide. *Journal of algorithms* 13, 3 (1992), 502–524.
- [16] MATULA, D., AND BECK, L. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM (JACM)* 30, 3 (1983), 427.
- [17] MUSTAFA, N., AND RAY, S. Improved results on geometric hitting set problems. www.mpi-inf.mpg.de/~saurabh.Papers/Hitting-Sets.pdf (2009).
- [18] MUSTAFA, N., AND RAY, S. PTAS for geometric hitting set problems via local search. In *Proceedings of the 25th annual symposium on Computational geometry* (2009), ACM, pp. 17–22.
- [19] NARAYANAPPA, S., AND VOJTECHOVSKÝ, P. An improved approximation factor for the unit disk covering problem. In *Proc. Can. Conf. on Comp. Geom* (2006), Citeseer.
- [20] RAZ, R., AND SAFRA, S. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997), ACM, pp. 475–484.
- [21] SCHRUIVER, A. *Combinatorial optimization: polyhedra and efficiency*. Springer Verlag, 2003.
- [22] VARADARAJAN, K. Weighted geometric set cover via quasi-uniform sampling. In *Proceedings of the 42nd ACM symposium on Theory of computing* (2010), ACM, pp. 641–648.
- [23] WAN, P., AND YI, C. Coverage by randomly deployed wireless sensor networks. *IEEE Transactions on Information Theory*, vol. 52, pp. 2658–2669 (2006).
- [24] WANG, J., AND ZHONG, N. Efficient point coverage in wireless sensor networks. *Journal of Combinatorial Optimization* 11, 3 (2006), 291–304.
- [25] YUN, Z., BAI, X., XUAN, D., LAI, T., AND JIA, W. Optimal Deployment Patterns for Full Coverage and k-Connectivity ($k \leq 6$) Wireless Sensor Networks. *IEEE/ACM TRANSACTIONS ON NETWORKING* 18, 3 (2010).
- [26] ZOU, F., WANG, Y., XU, X., LI, X., DU, H., WAN, P., AND WU, W. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science* (2009).

APPENDIX

Proof of Claim 2

Assume after increasing the radius of D_1 by ℓ , a new degenerate quadruple (D_5, D_6, D_7, D_8) is introduced. There

exists at most one point that can be shared by four weighted Voronoi cells. This means that some point x becomes a new degenerate point after the scaling D_1 . Then, there must exist at most one disk (assumed to be D_8) such that $p \notin \text{cell}(D_8)$ before scaling D_1 ; but $p \in \text{cell}(D_8)$ after this scaling operation.

Note that, in the scaling operation, we increase only the radius of disk D_1 , the Voronoi cells of other disks can only shrink, while the only cell that may grow is $\text{cell}(D_1)$, thus $D_8 = D_1$, and at the same time, p must already lie at the boundary point of $\text{cell}(D_5), \text{cell}(D_6), \text{cell}(D_7)$ before the scaling operation. We can calculate a critical value ℓ , such that after scaling D_1 , (D_1, D_5, D_6, D_7) will become a degenerate quadruple. There are at most $\binom{n}{3} = O(n^3)$ such triples (D_5, D_6, D_7) with which D_1 can form a new degenerate quadruple, thus there are at most $O(n^3)$ such critical values.

Proof of Lemma 6

To verify the first fact, consider any node p , for any disk $D \in \mathcal{D}_p$, if $x_D \geq \frac{1}{2}$, we add $\lfloor 2n \cdot x_D \rfloor \geq n$ copies of D to \mathcal{D}_0 , then p is at least n -covered. Otherwise, we define n disjoint subsets of disks, the i -th ($1 \leq i \leq n$) subset \mathcal{D}_p^i contains every disk $D \in \mathcal{D}_p$ satisfying $\frac{i-1}{2n} \leq x_D < \frac{i}{2n}$. Since $x_D < \frac{1}{2}$ for each disk $D \in \mathcal{D}_p$, these n subsets form a partition of \mathcal{D}_p . Let x_1, x_2, \dots, x_n denote the cardinalities of the n subsets respectively. We have $\sum_{1 \leq i \leq n} x_i \cdot \frac{i}{2n} > \sum_{D \in \mathcal{D}_p} x_D \geq 1$, this means that $\sum_{1 \leq i \leq n} i \cdot x_i > 2n$. Since $\sum_{1 \leq i \leq n} x_i \leq n$, we have $\sum_{1 \leq i \leq n} (i-1) \cdot x_i > n$, thus $\sum_{D \in \mathcal{D}_p} \lfloor 2n \cdot x_D \rfloor \geq \sum_{1 \leq i \leq n} (i-1) \cdot x_i > n$. This implies that \mathcal{D}_0 at least n -cover p .

To verify the second fact, observe that $w(\mathcal{D}_0) = \sum_{D \in \mathcal{D}} \lfloor 2n \cdot x_D \rfloor < \sum_{D \in \mathcal{D}} 2n \cdot x_D = 2n \cdot \sum_{D \in \mathcal{D}} x_D = 2n \cdot \lambda^*$.

Proof of Lemma 7

Given a disk set \mathcal{D} , let us define a *critical radius* for a node p as:

$$\gamma(p) = \min_{D \in \mathcal{D}} \{\omega(p, D) : \omega(p, D) > 0\}.$$

Here the weight function ω is defined in Section 3.3.1. Given a disk set \mathcal{D} and a node set \mathcal{P} , we map any node $p \in \mathcal{P}$ to a disk Q_p with the radius strictly smaller than the critical radius for p . Clearly, for any disk $D \in \mathcal{D}$, when $p \notin D$, then Q_p does not intersect D ; when $p \in D$, then Q_p intersects D .

We then introduce a similar result by Gibson *et al.* [10]:

LEMMA 11. *Let \mathcal{D} be a set of m disks, and $1 \leq L \leq m$ be an integer. Let \mathcal{Q} be a set of disks (possibly infinite). There are $O(mL^2)$ disks of \mathcal{Q} that intersect distinct subsets of \mathcal{D} , each of size at most L .*

Let \mathcal{Q} be the set of all the mapped disks for the nodes in \mathcal{P} , based on Lemma 11, there are $O(mL^2)$ disks of \mathcal{Q} that intersect distinct subsets of \mathcal{D} , each of size at most L . Correspondingly, there are $O(mL^2)$ nodes of \mathcal{P} that are covered

by distinct subsets of \mathcal{D} with size at most L . This finishes the proof.

Proof of Theorem 3

By the proposed algorithm in Table 4, the output \mathcal{D}' at least $\log L$ cover every node p if p is L -covered in \mathcal{D} . We next show that $\Pr(D \in \mathcal{D}') \leq \frac{c \log L}{L}$. If a disk D is not forced, clearly, $\Pr(D \in \mathcal{D}') = \frac{\log L}{L} \leq \frac{c \log L}{L}$. We are only left to upper bound the probability that a disk is forced.

For each disk D_j in the sequence σ , If the disk D_j is forced for node p , then all the disks $D_{j'}$ (with $j' \geq j$) that cover p are also forced, and the number of such disks is at most $\log L - 1$ (since otherwise D_j will not be forced). At the same time, some disk D_i ($i \leq j$) must be *first forced* for node p . Here a disk is first forced for node p if it appears as the first one among all forced disks for node p according to the order σ . We have the following inequality:

$$\begin{aligned} & \Pr(D_j \text{ is forced for node } p) \\ & \leq \log L \cdot \Pr(D_i (i \leq j) \text{ is first forced for node } p) \end{aligned}$$

We next compute the probability of a disk D_i being first forced for node p . Let $\mathcal{D}_{p,i}$ be the set of all disks in $\{D_k : k \leq i\}$ that cover the node p , and n_1 be the number of disks in $\mathcal{D}_{p,i}$ that is finally selected, then the following three facts hold:

- (1) D_i covers p ,
- (2) $|\mathcal{D}_{p,i}| \geq L - \log L \geq \frac{L}{2}$,
- (3) $n_1 \leq \log L$.

Clearly, $\Pr(D_i \text{ is first forced for node } p)$ is at most the probability that fact (3) holds when both facts (1), (2) are true, which means less than $\log L$ disks are finally selected for $\mathcal{D}_{p,i}$ while each disk from $\mathcal{D}_{p,i}$ is selected with probability at least $\frac{c \log L}{L}$. This can be reduced to a coin toss problem: in a sequence of at least $\frac{L}{2}$ coin tosses, each coin turns up head with probability of at least $\frac{c \log L}{L}$, less than $\log L$ coins turn up heads. Based on Chernoff bound, the probability is at most $\frac{1}{e^{\frac{c \log L}{16}}}$. Thus

$$\Pr(D_i \text{ is first forced for node } p) \leq \frac{1}{e^{\frac{c \log L}{16}}}$$

Since there are at most $2c'L^2$ classes of C_j covered with multiplicity in $[L, 2L]$ and covered by D_j , D_j can be forced for any one of the at most $2c'L^3$ representative nodes. To sum up, we have the following inequalities:

$$\begin{aligned} & \Pr(D_j \text{ is forced}) \\ & \leq 2c'L^3 \cdot \Pr(D_j \text{ is forced for node } p) \\ & \leq 2c'L^3 \cdot \log L \cdot \Pr(D_i (i \leq j) \text{ is first forced for node } p) \\ & \leq 2c'L^3 \cdot \log L \cdot \frac{1}{e^{\frac{c \log L}{16}}} \end{aligned}$$

By setting appropriate values for the constants c and c' , we can ensure that the probability of a given disk being selected is at most $\frac{c \log L}{L}$ in the proposed algorithm (in Table 4). This finishes the proof.

Proof of Lemma 8

Given an instance of remember a wireless cover is defined as a subset of wireless nodes such that each target is covered by some wireless node(s). By mapping each node u to a disk D_u with $y(u) p_C(u)$ as the cost of disk D_u , and each target by a node to be covered, we can reduce the problem MCC to MCDG immediately, which imply the algorithmic results for MCC listed in Lemma 8. In addition, when $p \cdot y$ is an all-one vector, the cost of a wireless cover C w.r.t. to y is exactly $|C|$, i.e. the number of wireless nodes, this corresponds exactly to the problem MDC.

Proof of Lemma 10

Let

$$f(t) = (1 + \varepsilon)^z - (1 + \varepsilon z).$$

Clearly $f(0) = f(1) = 0$. Since $(1 + \varepsilon)^t$ is convex and $1 + \varepsilon t$ is linear, $f(t)$ is also convex. Therefore, for any $t \in [0, 1]$,

$$f(t) \leq \max\{f(0), f(1)\} = 0,$$

which implies that

$$(1 + \varepsilon)^t \leq 1 + \varepsilon t.$$

Taking the logarithm with base $1 + \varepsilon$, the lemma follows.