Journal of Combinatorial Optimization (2022) 44:1913–1923
https://doi.org/10.1007/s10878-020-00567-3

A primal-dual algorithm for the minimum power partial cover problem

Menghong Li[1] · Yingli Ran[1] · Zhao Zhang[1]

Published online: 7 April 2020

Abstract
In this paper, we study the minimum power partial cover problem (MinPPC). Suppose X is a set of points and $\mathcal{S}$ is a set of sensors on the plane, each sensor can adjust its power and the covering range of a sensor s with power $p(s)$ is a disk of radius $r(s)$ satisfying $p(s) = c \cdot r(s)^{\alpha}$. Given an integer $k \leq |X|$, the MinPPC problem is to determine the power assignment on every sensor such that at least k points are covered and the total power consumption is the minimum. We present a primal-dual algorithm for MinPPC with approximation ratio at most 3^{α}. This ratio coincides with the best known ratio for the minimum power full cover problem, and improves previous ratio $(12 + \varepsilon)$ for MinPPC which was obtained only for $\alpha = 2$.

Keywords Power · Partial cover · Primal dual · Approximation algorithm

1 Introduction

With the rapid development of wireless sensor networks (WSNs), intensive studies on WSNs have emerged, especially on the coverage problem. In a coverage problem, the most basic requirement is to keep all points of interest under monitoring. In a typical WSN, the service area of a sensor is a disk centered at the sensor whose radius is determined by the power of the sensor. A typical relation between the power $p(s)$ of sensor s and the radius $r(s)$ of its service area is

$$p(s) = c \cdot r(s)^{\alpha}, \tag{1}$$

✉ Yingli Ran
724609171@qq.com

✉ Zhao Zhang
hxhzz@163.com

1 College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

where c and $\alpha \geq 1$ are some constants (α is usually called the *attenuation factor*). So, a larger service area needs more power. In other words, the consumption of energy and the quality of service are two conflicting factors. The question is how to balance these two conflicting factors by adjusting power at the sensors so that the desired service can be accomplished using the minimum total power. This question is motivated by the intention to extend the lifetime of WSN under limited energy supply, and we call it the *minimum power coverage* problem (MinPC).

In the real world, it is often too costly to satisfy the covering requirement of every point of interest (Liu and Huang 2018). It is not cost-effective to sacrifice a lot of power on serving some distant outliers. So, it is beneficial to study the *minimum power partial coverage* problem (MinPPC), in which it is sufficient to cover at least $k < |X|$ points. The problem is motivated by the purpose of further saving energy while keeping an acceptable quality of service.

The MinPPC problem can be viewed as a special case of the *minimum weight partial set cover* problem (MinWPSC). Given a set E of elements, a collection of sets $\mathcal{S}$, a weight function $w : \mathcal{S} \mapsto \mathbb{R}^+$, and an integer $k \leq |E|$, the MinWPSC problem is to find the minimum weight sub-collection of sets $\mathcal{F} \subseteq \mathcal{S}$ such that at least k elements are covered by $\mathcal{F}$, i.e., $|\bigcup_{S\in\mathcal{F}} S| \geq k$ and $w(\mathcal{F}) = \sum_{S\in\mathcal{F}} w(S)$ is minimum. Notice that in a MinPPC problem, the power at a sensor can be discretized by assuming that there is a point of interest on the boundary of the disk supported by the assigned power. We call such a disk as a *canonical disk*. So, if we associate with each sensor $|X|$ canonical disks, each disk corresponds to the set of points of interest contained in it, and the weight of the disk equals the power supporting the disk which is determined by Eq. (1), then the MinPPC problem is reduced to the MinWPSC problem.

It is known that the MinWPSC problem has a $(\ln(\min\{\lceil k\rceil, \Delta\})+1)$-approximation (Slavík 1997) and an f-approximation (Bar-Yehuda 2001), where Δ is the size of a maximum set and f is the maximum frequency of an element (that is, the maximum number of sets containing a common element). For the MinWPSC problem obtained by the above reduction from a MinPPC problem, Δ equals to the number of points to be covered, f equals the number of sensors, and k can be as large as $\Theta(n)$. So, the above ratios for MinWPSC are too large to be good approximation factors for MinPPC. The main purpose of this paper is to explore geometric properties of the MinPPC problem to obtain a better approximation.

1.1 Related works

The *minimum weight set cover* problem (MinWSC) is a classic combinatorial problem. It is well-known that MinSC admits approximation ratio $H(\Delta)$ (Chvatal 1979; Johnson 1974), where $H(\Delta) = 1 + \frac{1}{2} + \cdots + \frac{1}{\Delta}$ is the Harmonic number and Δ denotes the size of the largest set (it is known that $H(\Delta) \leq \ln \Delta + 1$). It is also known that a simple LP-rounding algorithm can achieve an approximation ratio of f, where f is the maximum number of sets containing a common element [see for example Chapter 12 of the book Vazirani 2001].

For the *minimum weight partial set cover* problem (MinWPSC), (Slavík 1997) obtained an $H(\min\{\lceil k\rceil, \Delta\})$-approximation using greedy strategy, (Bar-Yehuda 2001) obtained an f-approximation using local ratio method, (Gandhi et al. 2004) also obtained f approximation using primal-dual method. Very recently, Inamdar and Varadarajan (2018) designed an LP-rounding algorithm, obtaining approximation ratio $2\beta + 2$, where β is the integrality gap for the linear program of the minimum weight (full) set cover problem.

For the *geometric minimum weight set cover* problem, much better approximation factors can be achieved. Using partition and shifting method, (Hochbaum 1982) obtained a PTAS for the minimum unit disk cover problem in which the disks are uniform and there are no prefixed locations for the disks. For the minimum disk cover problem in which disks may have different sizes, Mustafa and Ray (2010) designed a PTAS using a local search method. This PTAS was generalized by Roy et al. (2018) to non-piercing regions including pseudo-disks. These are results for the cardinality version of the geometric set cover problem. Considering weight, Varadarajan (2010) presented a clever quasi-uniform sampling technique, which was improved by Chan et al. (2012), yielding a constant approximation for the minimum weight disk cover problem. This constant approximation was generalized by Bansal and Pruhs (2012) for the minimum weight disk multi-cover problem in which every point has to be covered multiple times. Using a separator framework, Mustafa et al. (2015) obtained a quasi-PTAS for the minimum weight disk cover problem.

To our knowledge, there are two papers studying the *geometric minimum partial set cover* problem. The first paper is Gandhi et al. (2004), in which Gandhi et al. presented a PTAS for the minimum (cardinality) partial unit disk cover problem using partition and shifting method. Notice that this result only works for the case when the centers of the disks are not prefixed. Another paper is due to Inamdar and Varadarajan (2018), in which a $(2\beta + 2)$-approximation was obtained for the *general* minimum weight partial set cover problem, where β is the integrality gap of the natural linear program for the minimum weight (full) set cover problem. As a consequence, for those geometric set cover problems (including the disk cover problem) in which β is a constant, the approximation ratio for the partial version is also a constant.

Recently, there are a lot of works studying the *minimum power multi-cover* problem (MinPMC), in which every point x is associated with a covering requirement cr_x, and the goal is to find a power assignment with the minimum total power such that every point x is covered by at least cr_x disks. Let cr_{max} be the maximum number of times that a point requires to be covered. Using local ratio method, Bar-Yehuda and Rawitz (2013) presented a $3^\alpha \cdot cr_{max}$-approximation algorithm. The dependence on $cr_{\max}$ was removed by Bhowmick et al. (2015), achieving an approximation ratio of $4 \cdot (27\sqrt{2})^\alpha$. This result was further generalized to any metric space in Bhowmick et al. (2017), the approximation ratio is at most $2 \cdot (16 \cdot 9)^\alpha$. For the minimum power (*single*) cover problem, the best known ratio is 3^α (as a consequence of Bar-Yehuda and Rawitz (2013) and the fact $cr_{\max} = 1$ in this case).

Prior to our study, there is only one paper (Freund and Rawitz 2003) studying the minimum power *partial* (single) cover problem, and the study is on the special case when $\alpha = 2$. The approximation ratio obtained in Freund and Rawitz (2003) is $(12 + \varepsilon)$, where ε is an arbitrary constant greater than zero, by a reduction to a prize-collecting coverage problem.

1.2 Contribution

In this paper, we show that the MinPPC problem can be approximated within factor 3^α, which coincides with the best known ratio for the MinPC problem (the full version of the minimum power coverage problem). When applied to the case when $\alpha = 2$, our ratio is 9, which is better than $12 + \varepsilon$ obtained in Freund and Rawitz (2003).

In the conference version of this paper (Li et al. (2019)), we have shown that ratio 3^α can be achieved by a local ratio method. In this paper, we find that a primal-dual method can also achieve the same ratio, furthermore, the algorithm and the analysis can be even simpler than the local ratio method.

A difficulty of implementing a primal-dual framework on the partial cover problem is that the natural LP for the partial cover problem has integrality gap arbitrarily large. The reason is that the last disk chosen into the solution may cover much more points than needed, and thus its cost cannot be controlled. Based on this observation, by guessing a disk with the largest radius in an optimal solution and working on an LP which is constructed on the residual instance, we could get a better approximation..

The remaining part of this paper is organized as follows. In Sect. 2, we formally define the problem and introducing the preprocessing step of guessing. In Sect. 3, the primal-dual algorithm is presented, together with a strict analysis on its time complexity and approximation ratio. Section 4 concludes the paper.

2 The problem and a preprocessing

We first give a formal definition of the MinPPC problem.

Definition 2.1 [Minimum Power Partial Cover (MinPPC)] Suppose X is a set of n points and S is a set of m sensors on the plane, k is an integer satisfying $0 \le k \le n$. A point $x \in X$ is covered by a sensor $s \in S$ with power $p(s)$ if x belongs to the disk supported by $p(s)$, that is $x \in Disk(s, r(s))$, where $Disk(s, r(s))$ is the disk centered at s whose radius $r(s)$ is determined by $p(s)$ through equation (1). A point is covered by a power assignment $p : S \mapsto \mathbb{R}^+$ if it is covered by some disk supported by p. The goal of MinPPC is to find a power assignment p covering at least k points such that the total power $\sum_{s \in S} p(s)$ is as small as possible. Here, we assume that there is no limit on the power at a sensor.

In an optimal solution, we may assume that for any sensor s, there is at least one point that is on the boundary of the disk $Disk(s, r(s))$, since otherwise we may reduce $p(s)$ to cover the same set of points, resulting in a lower power consumption. Therefore, at most mn disks need to be considered. We denote the set of such disks by $\mathcal{D}$. In the following, denote by $(X, \mathcal{D}, k)$ an instance of the MinPPC problem, and use $opt(X, \mathcal{D}, k)$ to denote the optimal power for the instance $(X, \mathcal{D}, k)$. To simplify the notation, we use D to represent both a disk in $\mathcal{D}$ and the set of points contained in D, and use $r(D)$ and $p(D)$ to denote the radius and the power of disk D, where $p(D) = c \cdot r(D)^\alpha$. For a set of disks $\mathcal{D}$, we shall use $\mathcal{C}(\mathcal{D}) = \bigcup_{D \in \mathcal{D}} D$ to denote the set of points covered by the union disks of $\mathcal{D}$.

In order to control the approximation factor of our algorithm, we need a preprocessing step: guessing the maximum power of a sensor (or equivalently, the radius of a maximum disk) in an optimal solution. Suppose $D_{\max} \in \mathcal{D}$ is the guessed disk. Denote by $\mathcal{D}_{\leq r(D_{\max})}$ the subset of disks of $\mathcal{D}$ whose radii are no greater than the radius of $D_{\max}$ (excluding $D_{\max}$), and denote by $(X \backslash D_{\max}, \mathcal{D}_{\leq r(D_{\max})}, k - |D_{\max}|)$ the *residual instance* after guessing $D_{\max}$. The following lemma is obvious.

Lemma 2.2 *Suppose* $D_{\max}$ *is the correctly guessed disk with the maximum power in an optimal solution of instance* $(X, \mathcal{D}, k)$*. Then*

$$opt(X, \mathcal{D}, k) = opt(X \backslash D_{\max}, \mathcal{D}_{\leq r(D_{\max})}, k - |D_{\max}|) + p(D_{\max}).$$

3 A primal dual algorithm

In this section, we present a primal-dual algorithm for the MinPPC problem on the instance $(X \backslash D_{\max}, \mathcal{D}_{\leq r(D_{\max})}, k - |D_{\max}|)$. And then show how to make use of it to find a power assignment for the MinPPC problem.

3.1 Algorithm after the preprocessing

For simplicity of notation in this section, we still use $(X, \mathcal{D}, k)$ to denote the residual instance after the guessing, assuming that every disk in $\mathcal{D}$ has radius at most $r(D_{\max})$.

The algorithm consists of three steps.

(i) In the first step, a primal dual method is employed to find a feasible solution $\bar{\mathcal{D}}$, that is, $\bar{\mathcal{D}}$ covers at least k points.

(ii) Before going into the second step, remove the disk D_{rmv} which is the last disk added into $\bar{\mathcal{D}}$. Then, in the second step, a *maximal independent set of disks* $\mathcal{I} \subseteq \bar{\mathcal{D}} \backslash \{D_{rmv}\}$ is computed in a greedy manner, that is, disks in $\mathcal{I}$ are mutually disjoint, while any disk $D \in \bar{\mathcal{D}} \backslash \{D_{rmv}\}$ which is not picked into $\mathcal{I}$ intersects some disk in $\mathcal{I}$.

(iii) In the third step, every disk in $\mathcal{I}$ has its radius enlarged three times. Such set of disks together with $\{D_{\max}, D_{rmv}\}$ are the output of the algorithm.

The first step is accomplished by Algorithm 1, in which the MinPPC instance $(X, \mathcal{D}, k)$ is viewed as an instance of the minimum weight partial set cover problem, where X serves as the set of elements to be covered, $\mathcal{D}$ serves as the collection of sets, and the weight of each $D \in \mathcal{D}$ is $p(D)$.

The MinPPC problem can be formulated as an integer program. Variable $z_D \in \{0, 1\}$ indicates whether disk $D \in \mathcal{D}$ is picked, that is, $z_D = 1$ if and only if D is picked. Variable $y_x \in \{0, 1\}$ indicates whether point $x \in X$ is covered, here $y_x = 0$ if and only if x is covered. The following is the LP-relaxation of the integer program.

$$
\begin{aligned}
\min \quad & \sum_{D \in \mathcal{D}} c \cdot r(D)^{\alpha} z_D \\
\text{s.t.} \quad & \sum_{D:\, x \in D} z_D + y_x \geq 1, \quad \forall x \in X \\
& \sum_{x \in X} y_x \leq n - k \\
& z_D \geq 0, \quad \forall D \in \mathcal{D} \\
& y_x \geq 0, \quad \forall x \in X
\end{aligned}
\tag{2}
$$

Notice that we need not add the constraints $z_D \leq 1$ and $y_x \leq 1$ since they are automatically satisfied in an optimal solution of (2). The dual program is:

$$
\begin{aligned}
\max \quad & \sum_{x \in X} \beta_x - (n-k)\gamma \\
s.t. \quad & \sum_{x \in D} \beta_x \leq c \cdot r(D)^{\alpha}, \quad \forall D \in \mathcal{D} \\
& 0 \leq \beta_x \leq \gamma, \quad \forall x \in X \\
& \gamma \geq 0
\end{aligned}
$$

For a dual feasible solution (β, γ), we say that a disk $D \in \mathcal{D}$ is *tight* if $\sum_{x \in D} \beta_x = c \cdot r(D)^{\alpha}$. The subprocedure PD follows the classic primal-dual method: starting from the trivial dual feasible solution zero, it increases dual variables simultaneously until some disk becomes tight. Pick such a tight disk and iterate until a feasible solution is obtained. In line 5 of Algorithm 1, those points which have been covered by a tight disk is removed from X, the purpose of this step is to *freeze* the dual variables of these points in the sense that β_x will no longer increase for any point x which have been covered by tight disks. Furthermore, γ keeps increasing until a feasible solution is found. So, $\gamma = \max\{\beta_x : x \in X\}$ all the time. Hence the dual feasibility is kept throughout the algorithm.

Algorithm 1 $PD(X, \mathcal{D}, p, k)$.

Input: A set of points X, a set of disks $\mathcal{D}$, a weight function $p : \mathcal{D} \mapsto \mathbb{R}^+$, a covering requirement k.
Output: A subset of disks $\bar{\mathcal{D}}$ covering at least k points.

1: $\bar{\mathcal{D}} \leftarrow \emptyset$, $\beta_x \leftarrow 0$ for each $x \in X$, $\gamma \leftarrow 0$.
2: **while** $|\mathcal{C}(\bar{\mathcal{D}})| < k$ **do**
3: Increase $\{\beta_x\}_{x \in X}$ and γ simultaneously until some disk D becomes tight.
4: $\bar{\mathcal{D}} \leftarrow \bar{\mathcal{D}} \cup \{D\}$
5: $X \leftarrow X \setminus D$
6: **end while**
7: Return $\bar{\mathcal{D}}$.

Algorithm 2 finds a maximal independent set $\mathcal{I}$ of $\bar{\mathcal{D}} \setminus \{D_{rmv}\}$, where D_{rmv} is the last disk added into $\bar{\mathcal{D}}$. The removal of D_{rmv} is crucial for the estimation of $p(\mathcal{I})$.

Algorithm 2 $IS(\mathcal{D})$.

Input: A set of disks $\mathcal{D}$.
Output: A maximal independent set of disks $\mathcal{I}$.
1: $\mathcal{I} \leftarrow \emptyset$
2: **while** $\mathcal{D} \neq \emptyset$ **do**
3: $D' \leftarrow \arg\max_{D \in \mathcal{D}} r(D)$
4: $\mathcal{I} \leftarrow \mathcal{I} \cup \{D'\}$
5: $\mathcal{N} \leftarrow$ the set of disks of $\mathcal{D}$ that intersect D'
6: $\mathcal{D} \leftarrow \mathcal{D} \backslash \mathcal{N}$
7: **end while**
8: Return $\mathcal{I}$

Algorithm 3 combines the above two algorithms to compute a feasible solution $\mathcal{M}$ to the residual instance. We use $c(D)$ and $r(D)$ to denote the center and the radius of disk D, respectively. So, $Disk(c(D), 3r(D))$ represents the disk with center $c(D)$ and radius $3r(D)$ (which is a disk obtained from D by enlarging its radius by three times). Notice that $\mathcal{M}$ is no longer confined to be a subset of $\mathcal{D}$.

Algorithm 3 $Cov(X, \mathcal{D}, k)$

Input: A residual instance $(X, \mathcal{D}, k)$.
Output: a set of disks $\mathcal{M}$ covering at least k points.
1: $\bar{\mathcal{D}} \leftarrow PD(X, \mathcal{D}, p, k)$
2: $D_{rmv} \leftarrow$ The last disk added into $\bar{\mathcal{D}}$
3: $\mathcal{I} \leftarrow \text{IS}(\bar{\mathcal{D}} \backslash \{D_{rmv}\})$
4: $\mathcal{M} \leftarrow \{Disk(c(D), 3r(D)) \colon D \in \mathcal{I}\} \cup \{D_{rmv}\}$
5: Return $\mathcal{M}$

Lemma 3.1 *Suppose $\mathcal{D}^*$ is an optimal solution for $(X, \mathcal{D}, k)$. Then the independent set of disks $\mathcal{I}$ output by Algorithm 2 based on $\bar{\mathcal{D}}$ computed by Algorithm 1 satisfies $p(\mathcal{I}) \leq p(\mathcal{D}^*)$.*

Proof Since any disk $D \in \mathcal{I} \subseteq \bar{\mathcal{D}}$ is tight, we have

$$\begin{aligned} p(\mathcal{I}) &= \sum_{D \in \mathcal{I}} c \cdot r(D)^\alpha = \sum_{D \in \mathcal{I}} \sum_{x \in D} \beta_x = \sum_{x \in \mathcal{C}(\mathcal{I})} \beta_x |\{D \colon D \in \mathcal{I}, x \in D\}| \\ &= \sum_{x \in \mathcal{C}(\mathcal{I})} \beta_x = \sum_{x \in X} \beta_x - \sum_{x \in X \backslash \mathcal{C}(\mathcal{I})} \beta_x, \end{aligned} \tag{3}$$

where the fourth equality holds because $\mathcal{I}$ is an independent set and thus any point $x \in \mathcal{C}(\mathcal{I})$ is covered by exactly one disk of $\mathcal{I}$.

Notice that $X \backslash \mathcal{C}(\mathcal{I}) \supseteq X \backslash \mathcal{C}(\bar{\mathcal{D}} \backslash \{D_{rmv}\})$. Observe that

$$\beta_x = \gamma \text{ for any } x \in X \backslash \mathcal{C}(\bar{\mathcal{D}} \backslash \{D_{rmv}\}),$$

where both β and γ refer to the values at the end of Algorithm 1. In fact, since D_{rmv} is the *last* disk added into $\bar{\mathcal{D}}$, a point $x \in X \backslash \mathcal{C}(\bar{\mathcal{D}} \backslash \{D_{rmv}\})$ implies that x is not covered by any disk before the last iteration, and thus its dual variable β_x keeps increasing *at the same rate* as γ until the termination of the algorithm. The reason why D_{rmv} is added into $\bar{\mathcal{D}}$ is because $|\mathcal{C}(\bar{\mathcal{D}}\backslash\{D_{rmv}\})| < k$. So

$$|X\backslash\mathcal{C}(\bar{\mathcal{D}}\backslash\{D_{rmv}\})| > n - k.$$

It follows that

$$\sum_{x\in X\backslash\mathcal{C}(\mathcal{I})} \beta_x \geq \sum_{x\in X\backslash\mathcal{C}(\bar{\mathcal{D}}\backslash\{D_{rmv}\})} \beta_x \geq \gamma(n-k).$$

Substituting this inequality into (3),

$$p(\mathcal{I}) \leq \sum_{x\in X} \beta_x - \gamma(n-k).$$

The righthand side of this inequality is exactly the objective value of the dual program (3), which provides a lower bound for $p(\mathcal{D}^*)$. The lemma is proved. □

Theorem 3.2 *The set of disks $\mathcal{M}$ computed by Algorithm 3 covers at least k points.*

Proof The set of disks in $\bar{\mathcal{D}}$ computed in line 1 of Algorithm 3 cover at least k points. For any point x which is covered by $\bar{\mathcal{D}}$, if x is covered by D_{rmv} or any disk in $\mathcal{I}$, then it is also covered by $\mathcal{M}$. Otherwise, x is covered by a disk D which is removed in line 6 of Algorithm 2. This disk D is removed because it intersects a disk $D' \in \mathcal{I}$. Because of the greedy choice of disk D' in line 3 of Algorithm 2, we have $r(D) \leq r(D')$. Hence $d(x, c(D')) \leq d(x, c(D)) + d(c(D), c(D')) \leq r(D) + (r(D) + r(D')) \leq 3r(D')$, where $d(\cdot, \cdot)$ denotes the Euclidean distance. So, x is covered by $disk(c(D'), 3r(D')) \in \mathcal{M}$. □

The next theorem estimates the approximation effect of Algorithm 3.

Theorem 3.3 *Suppose $\mathcal{C}^*$ is an optimal solution on instance $(X, \mathcal{D}, p, k)$, and $\mathcal{M}$ is the output of Algorithm 3. Then*

$$p(\mathcal{M}) \leq 3^\alpha p(\mathcal{C}^*) + p(D_{rmv}).$$

Proof For each disk $D \in \mathcal{M}\backslash\{D_{rmv}\}$, it comes from a disk $D' \in \mathcal{I}$ by expanding the radius by three times. Hence by (1), $p(D) = 3^\alpha p(D')$. So $p(\mathcal{M}) = 3^\alpha p(\mathcal{I}) + p(D_{rmv})$, and the theorem follows from Lemma 3.1. □

By Theorem 3.3, the approximate effect is related with $p(D_{rmv})$. The reason why we should guess a disk $D_{\max}$ with the largest radius in an optimal solution is now clear: to control the term $p(D_{rmv})$ to be not too large. The algorithm combining the guessing technique is presented as follows.

3.2 The whole algorithm

Algorithm 4 is the whole algorithm for the MinPPC problem. It first guesses a disk with the maximum radius in an optimal solution, takes it, and then uses Algorithm 3 on the residual instance. For a guessed disk D, the residual instance consists of all those disks $\mathcal{D}_{\le r(D)}$ whose radii are no larger than $r(D)$ (excluding D itself), and the goal is to cover the remaining elements $X \backslash D$ beyond the remaining covering requirement $\max\{0, k - |D|\}$. The weight function, denoted as p_D, is determined by (1). If for a guessed disk D, Algorithm 3 does not return a feasible solution, then we regard the solution to have cost ∞. Algorithm 4 returns the best solution among all the guesses.

Algorithm 4 $MinPPC(X, \mathcal{D}, k, p)$

Input: A set of points X, a set of sensors $\mathcal{S}$, a covering requirement k.
Output: A power assignment p to cover at least k points.

Construct the set $\mathcal{D}$ of canonical disks, and determine the weight of disks by (1).
for $D \in \mathcal{D}$ **do**
  $\mathcal{M}_D \leftarrow Cov(X\backslash D, \mathcal{D}_{\le r(D)}, \max\{0, k - |D|\})$
  $\mathcal{F}_D \leftarrow \mathcal{M}_D \cup \{D\}$
end for
$\widetilde{D} \leftarrow \arg\min_{D \in \mathcal{D}}\{p(\mathcal{F}_D)\}$
Return the power assignment corresponding to $\mathcal{F}_{\widetilde{D}}$

Theorem 3.4 *Algorithm 4 is a 3^α-approximation algorithm for MinPPC, which runs in time $O(kn^2m^2)$.*

Proof Suppose $\mathcal{C}^*$ is an optimal solution to the MinPowerPartCov problem, and $D_{\max}$ is a disk with the maximum radius in $\mathcal{C}^*$. By Theorem 3.3 and the fact $p(D_{max,rmv}) \le p(D_{max})$ (where $D_{max,rmv}$ is the removed disk when the guessed disk is D_{max}), we have

$$\begin{aligned} p(\mathcal{F}_{D_{\max}}) = p(\mathcal{M}_{D_{\max}}) + p(D_{\max}) &\le 3^\alpha p(\mathcal{C}^*_{D_{\max}}) + 2p(D_{\max}) \\ &\le 3^\alpha \left(p(\mathcal{C}^*_{D_{\max}}) + p(D_{\max})\right) = 3^\alpha opt, \end{aligned}$$

where opt is the optimal power, and $\mathcal{C}^*_{D_{\max}}$ is the optimal solution when the guessed disk is $D_{\max}$. Since the set $\mathcal{F}_{\widetilde{D}}$ computed by Algorithm 4 satisfies $p(\mathcal{F}_{\widetilde{D}}) \le p(\mathcal{F}_{D_{\max}})$, the approximation ratio 3^α is proved.

The for loop of Algorithm 4 is executed $O(nm)$ times. Since in each while loop of Algorithm 1, the number of covered points is increased by at least one, the number of iterations before at least k points are covered is $O(k)$. Furthermore, the running time of line 3 in Algorithm 1 is $O(mn)$. So, the overall time complexity for Algorithm 1 is $O(kmn)$. Since the output $\bar{\mathcal{D}}$ of Algorithm 1 has $O(k)$ disks, the running time for Algorithm 2 is $O(k \log k)$, which is the time needed to order the $O(k)$ disks of $\bar{\mathcal{D}}$. Therefore, the overall time complexity of Algorithm 4 is $O(kn^2m^2)$. □

4 Conclusion

In this paper, we presented an approximation algorithm for the minimum power partial cover problem achieving approximation ratio 3^α, using a primal-dual method. This ratio improves the ratio of $(12 + \varepsilon)$ in Freund and Rawitz (2003) which was obtained only for $\alpha = 2$, and matches the best known ratio for the minimum power (full) cover problem in Bar-Yehuda and Rawitz (2013).

Recently, there are a lot of studies on the minimum power multi-cover problem (Bhowmick et al. 2015, 2017). A problem which deserves to be explored is the minimum power partial multi-cover problem (MinPPMC). This problem can be viewed as a special case of the minimum weight partial set multi-cover problem (MinWPSMC), in which every element x has a covering requirement cr_x and x is *fully covered* only when x is contained in at least cr_x selected sets. The goal of MinWPSMC is to select a minimum weight collection of subsets such that at least k elements are fully covered. According to current studies on MinWPSMC (Ran et al. 2017a, b, 2019), it seems that studying the combination of multi-cover and partial cover in a general setting is very difficult. An interesting question is how about the problem in some special setting? The speciality of MinPPMC lies not only in its intrinsic geometry, but also in the special weight function which relates the power and the radius of a disk. Such speciality might lead to better approximation.

Acknowledgements This research is supported in part by NSFC (11771013, 61751303, 11531011, 11901533) and ZJNSFC (LD19A010001, LA19A010018).

References

Bansal N, Pruhs K (2012) Weighted geometric set multi-cover via quasi-uniform sampling. In: ESA, pp 145–156

Bar-Yehuda R (2001) Using homogeneous weights for approximating the partial cover problem. J Algorithms 39(2):137–144

Bar-Yehuda R, Rawitz D (2013) A note on multicovering with disk. Comput Geom 46(3):394–399

Bhowmick S, Varadarajan K, Xue S-K (2015) A constant-factor approximation for multi-covering with disks. Comput Geom 6(1):220–24

Bhowmick S, Inamdar T, Varadarajan K (2017) On metric multi-covering problems. Comput Geom arxiv:1602.04152

Chan TM, Granty E, Konemanny J, Sharpe M (2012) Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In: SODA, pp 1576–1585

Chvatal V (1979) A greedy heuristic for the set-covering problem. Math Oper Res 4:233–235

Freund A, Rawitz D (2003) Combinatorial interpretations of dual fitting and primal fitting. A conference version in WAOA 137–150 A full version in http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.585.9484

Gandhi R, Khuller S, Srinivasan A (2004) Approximation algorithms for partial covering problems. J Algorithms 53(1):55–84

Hochbaum DS (1982) Approximation algorithms for the set covering and vertex cover problems. SIAM J Comput 11:555–556

Inamdar T, Varadarajan K (2018) On partial covering for geometric set system. Comput Geom 47:1–14

Johnson DS (1974) Approximation algorithms for combinatorial problems. J Comput Syst Sci 9:256–278

Li MH, Ran YL, Zhang Z (2019) Approximation algorithms for the minimum power partial cover problem. To appear in AAIM'19

Liu P, Huang X (2018) Approximation algorithm for partial set multicover versus full set multicover. Discrete Math Algorithms Appl 10(2):1850026
Mustafa NH, Ray S (2010) Improved results on geometric hitting set problems. Discrete Comput Geom 44:883–895
Mustafa NH, Raman R, Ray S (2015) Quasi-polynomial time approximation scheme for weighted geometric set cover on pseudodisks. SIAM J Comput 44(6):1650–1669
Ran Y, Zhang Z, Du H, Zhu Y (2017a) Approximation algorithm for partial positive influence problem in social network. J Combin Optim 33:791–802
Ran Y, Shi Y, Zhang Z (2017b) Local ratio method on partial set multi-cover. J Combin Optim 34(1):1–12
Ran Y, Shi Y, Tang C, Zhang Z (2019) A primal-dual algorithm for the minimum partial set multi-cover problem. J Combin Optim. https://doi.org/10.1007/s10878-019-00513-y
Roy AB, Govindarajan S, Raman R, Ray S (2018) Packing and covering with non-piercing regions. Discrete Comput Geom 60:471–492
Slavík P (1997) Improved performance of the greedy algorithm for partial cover. Inf Process Lett 64(5):251–254
Varadarajan K (2010) Weighted geometric set cover via quasi-uniform sampling. In: STOC'10, pp 641–648
Vazirani VV (2001) Approximation algorithms. Springer, Berlin