

Article

An Improved Approximation Algorithm for the Minimum Power Cover Problem with Submodular Penalty

Han Dai

School of Mathematics and Statistics, Yunnan University, Kunming 650504, China; 18283649705@163.com

Abstract: In this paper, we consider the minimum power cover problem with submodular penalty (SPMPC). Given a set U of n users, a set S of m sensors and a penalty function $\pi : 2^U \rightarrow \mathbb{R}_+$ on the plane, the relationship that adjusts the power $p(s)$ of each sensor s and its corresponding radius $r(s)$ is: $p(s) = c \cdot r(s)^\alpha$, where $c > 0$ and $\alpha \geq 1$. The SPMPC problem is to determine the power assignment on each sensor such that each user $u \in U$ is either covered by the sensor or penalized and the sum of the total power consumed by sensors in S plus the penalty of all uncovered users is minimized, the penalty here is determined by the submodular function. Based on the primal dual technique, we design an $O(\alpha)$ -approximation algorithm.

Keywords: power cover; submodular function; primal dual; approximation algorithm



Citation: Dai, H. An Improved Approximation Algorithm for the Minimum Power Cover Problem with Submodular Penalty. *Computation* **2022**, *10*, 189. <https://doi.org/10.3390/computation10100189>

Academic Editors: Akbar Ali, Guojun Li, Mingchu Li, Rao Li, Colton Magnant and Madhumangal Pal

Received: 17 September 2022

Accepted: 15 October 2022

Published: 19 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The minimum power cover problem is defined as follows. Given a set U of n users and a set S of m sensors on the plane, the relationship that adjusts the power $p(s)$ of each sensor s and its corresponding radius $r(s)$ is: $p(s) = c \cdot r(s)^\alpha$, where $c > 0$ and $\alpha \geq 1$. The minimum power cover problem is to find a power assignment of each sensor such that all users on the plane are covered and the total power consumption is minimized. Lev-Tov et al. [1] studied the 1-dimensional minimum power cover problem and obtained a fast algorithm in polynomial time. Charikar et al. [2] designed a primal dual algorithm and obtained a constant approximation. Since the power consumption of signal coverage will increase sharply with the increase of radius, it is very important to control the power consumption. Zhang et al. [3] studied wireless access points with limited capacity in mobile edge computing, proposed a local ratio based algorithm, and verified by simulation experiments that the algorithm can get a near optimal result in polynomial time. subsequently, Zhang et al. [4] studied the power control problem of the edge server antenna with limited capacity, proposed an m -approximation algorithm of the primal dual frame, and verified the performance and practicability of the algorithm through simulation experiments.

The minimum power partial cover problem is to find a power assignment that covers at least k users on the plane and has the smallest total power consumption. Freund and Rawitz [5] introduced this problem, and when $\alpha = 2$, they proposed a $(12 + \epsilon)$ -approximation algorithm. Li et al. [6] designed a primal dual algorithm to obtain a 3^α -approximation solution. Recently, Dai et al. [7] improved their algorithm and obtained an $O(\alpha)$ -approximation solution.

The minimum power multiple cover problem is a generalization of the minimum-power cover problem, where each user u has a cover requirement d_u , finding a power assignment for each sensor on the premise that each user satisfies its coverage requirements. Abu-Affash et al. [8] studied the special case where $\alpha = 2$ and all users $u \in U$ have the same coverage requirements $d_u = d$, and they presented an approximation algorithm with an approximation ratio of at most $23.02 + 63.91(d - 1)$. Bar-Yehuda and Rawitz [9] considered the case of general metric space and obtained a $3^\alpha \cdot d_{u_{max}}$ -approximate solution based on a local ratio algorithm, where $d_{u_{max}}$ is the maximum coverage requirement. Liang et al. [10]

considered the situation where all users and sensors are on a same straight line, and designed an optimal algorithm based on dynamic programming when the maximum cover requirement has a constant upper bound. Ran et al. [11] designed a polynomial time approximation scheme (PTAS) under the assumption that all users cover requirements have a constant upper bound.

The minimum power partial cover problem with penalty is also a generalization of the minimum power cover problem and the minimum power partial cover problem. It is assumed that each user has a penalty, at least k users are required to be covered, and the sum of the total power of covered users plus the penalty of uncovered users is minimized. Recently, Liu et al. [12] studied this problem and designed a 3^α -approximation algorithm based on the primal dual technique. In real life, the performance detection of sensor networks meets the property of submodule-diminishing marginal benefit. If a user is not covered, the punishment can be regarded as a loss of a reputation of company. As the number of users not covered increases, the loss of a reputation of company may remain unchanged. Therefore, when the penalty function is a submodular function, Liu et al. [13] designed a polynomial time $(5 \cdot 2^\alpha + 1)$ -approximation algorithm based on the primal dual framework for the minimum power partial cover problem with penalty on the plane. And Liu et al. [14] studied the minimum power cover problem with submodular penalty and obtained a $(3^\alpha + 1)$ -approximation algorithm, when the penalty function is a linear function, based on the plane subdivision technique, and a PTAS was given. In this paper, our algorithm improves on the minimum power cover problem with submodular penalty by Liu et al. [14].

The main structure of this paper is as follows. The first part describes the minimum power cover problem with submodular penalty; the second part gives a brief overview of the algorithm of [14]; the third part designs an improved primal dual algorithm for this problem; the fourth part gives a conclusion of this question.

2. Preliminary

We are given a set of n users $U = \{u_1, u_2, \dots, u_n\}$, a set of m sensors $S = \{s_1, s_2, \dots, s_m\}$ on the plane and a real-valued function $f : 2^U \rightarrow \mathbb{R}_+$ defined on all subsets of U . If for any subset $U_1 \subseteq U_2 \subseteq U$, $f(U_1) \leq f(U_2)$, then f is called monotonic. If $f(\emptyset) = 0$, then f is called normalized. If for any subset $U_1, U_2 \subseteq U$,

$$f(U_1) + f(U_2) \geq f(U_1 \cup U_2) + f(U_1 \cap U_2),$$

then f is called submodular.

In this paper, we consider the minimum power cover problem with submodular penalty (SPMPC). Given n users U , m sensors S and a penalty function $\pi : 2^U \rightarrow \mathbb{R}_+$ on the plane, the power $p(s)$ generated by each sensor s can be adjusted by its corresponding radius $r(s)$, and the relationship between $p(s)$ and $r(s)$ is:

$$p(s) = c \cdot r(s)^\alpha, \text{ where } c > 0, \alpha \geq 1.$$

For any user $u \in Disk(s, r(s))$, the user u is said to be covered by a circular cover area $Disk(s, r(s))$ with sensor center s and sensor radius $r(s)$. The SPMPC problem is to find a power assignment for each sensor such that each user $u \in U$ is either covered by the sensor or penalized and the sum of the total power consumed by sensors in S plus the penalties of all uncovered users is minimized, where the penalty is determined by the submodular function.

Let (U, S, p, π) be an instance of the SPMPC problem. Each circular coverage area in the optimal solution of this instance, there will be at least one point on its boundary, otherwise its radius can be reduced to cover the same point resulting in a smaller assignment. Therefore, each sensor has at most n disks of different radius, and all sensors have at most mn disks to be considered. Next, we denote such a disk set by \mathcal{D} . For any disk $D \in \mathcal{D}$, we use $U(D)$ to represent the set of users covered by the disk D , $c(D)$ to represent the center

of the disk D , and $p(D)$ to represent the power of the disk D . For any $\mathcal{D}' \subseteq \mathcal{D}$, $U(\mathcal{D}')$ represents the set of users covered by the union of all disks in \mathcal{D}' . Here, we assume that the power of all sensors is not limited.

For each subset $T \subseteq U$, let $z_T = 1$ indicate that the subset T is rejected; otherwise $z_T = 0$. For each disk $D \in \mathcal{D}$, let $x_D = 1$ indicate that the disk D is selected; otherwise $x_D = 0$.

The integer linear programming of the SPMPC problem is as follows:

$$\begin{aligned}
 & \min \sum_{D:D \in \mathcal{D}} c \cdot r(D)^\alpha x_D + \sum_{T:T \subseteq U} \pi(T)z_T \\
 \text{s.t.} \quad & \sum_{D:u \in U(D)} x_D + \sum_{T:u \in T \subseteq U} z_T \geq 1, \forall u \in U, \\
 & x_D \in \{0, 1\}, \forall D \in \mathcal{D}, \\
 & z_T \in \{0, 1\}, \forall T \subseteq U.
 \end{aligned} \tag{1}$$

Relaxing $x_D \in \{0, 1\}$ and $z_T \in \{0, 1\}$, the relaxed linear programming of the SPMPC problem is as follows:

$$\begin{aligned}
 & \min \sum_{D:D \in \mathcal{D}} c \cdot r(D)^\alpha x_D + \sum_{T:T \subseteq U} \pi(T)z_T \\
 \text{s.t.} \quad & \sum_{D:u \in U(D)} x_D + \sum_{T:u \in T \subseteq U} z_T \geq 1, \forall u \in U, \\
 & x_D \geq 0, \forall D \in \mathcal{D}, \\
 & z_T \geq 0, \forall T \subseteq U.
 \end{aligned} \tag{2}$$

The dual programming of the SPMPC problem is as follows:

$$\begin{aligned}
 & \max \sum_{u:u \in U} y_u \\
 \text{s.t.} \quad & \sum_{u:u \in U(D)} y_u \leq c \cdot r(D)^\alpha, \forall D \in \mathcal{D}. \\
 & \sum_{u:u \in T} y_u \leq \pi(T), \forall T \subseteq U. \\
 & y_u \geq 0, \forall u \in U.
 \end{aligned} \tag{3}$$

In order to intuitively understand the dual programming, for each user $u \in U$, we can consider y_u as the “paid” price of user u being covered. For each disk $D \in \mathcal{D}$, if $\sum_{u:u \in U(D)} y_u = c \cdot r(D)^\alpha$, the power consumed by disk D is “paid” and the disk D is called *tight*. For each user set $T \subseteq U$, if $\sum_{u:u \in T} y_u = \pi(T)$, the penalty cost of user subset T is “paid” and subset T is called *tight*.

3. Overview of the Algorithm of [14]

Let \mathcal{D}^{tight} denote the set of tight disks, and R^{temp} denote the set of temporarily rejected points. Our algorithm improves on the algorithm of [14]. We replace ball B in [14] with disk D and repeat the algorithm in three phases.

In the first phase, we start from the trivial dual feasible solution 0, and simultaneously increase the dual variable $\{y_u\}_{u \in U}$ of all users $u \in U$ until some disk D becomes tight, and add D to the tight disk set \mathcal{D}^{tight} , and the dual values of all users covered by D are frozen; otherwise, some user subset $T \subseteq U$ becomes tight, all users in T are added to the temporary rejection point set R^{temp} , and the dual values of all users in T are frozen. The above process iterates until all users in U are frozen. Therefore, dual feasibility is maintained throughout the algorithm.

In the second phase, we consider all tight disks in \mathcal{D}^{tight} , by descending order of radius, if $\mathcal{D}^{tight} \neq \emptyset$, select a disk D' with the largest radius in \mathcal{D}^{tight} for each iteration, and

for any $D \in \mathcal{D}^{tight}$, if $U(D) \cap U(D') \neq \emptyset$, delete D from \mathcal{D}^{tight} . This step iterates, and we finally get a maximum independent set I .

In the last phase, we expanded the radius of all disks in I by three times. Let p be the power assignment corresponding to I and R be the set of users that are not covered by any ball supported by p . Then, the output is (p, A, R) , where $A = U \setminus R$.

Note that the main idea of the algorithm of [14] is to greedily select a maximal independent set I from \mathcal{D}^{tight} in which any two disks D, D' satisfy $U(D) \cap U(D') \neq \emptyset$, and for any user $u \in U$ to be covered by at most one disk in the maximum independent set I , i.e.,

$$|\max_{u \in U} \{D | u \in D \in I\}| = 1.$$

then expand the radius of each disk in I by three times to cover all users in A . So the algorithm of [14] yields an approximate solution of $3^\alpha + 1$. The improved algorithm we designed in the third section will be different here.

4. An Improved Primal Dual Algorithm

Before the algorithm starts, we cite [7] for the definition of ρ -relaxed independent set. For any two disks $D_1, D_2 \in \mathcal{D}'$, if $U(D_1) \cap U(D_2) = \emptyset$ or

$$d(c(D_1), c(D_2)) > \rho \max\{r(D_1), r(D_2)\},$$

where $\rho \in [0, 2]$, $d(\cdot, \cdot)$ represents the Euclidean distance, then \mathcal{D}' is called a ρ -relaxed independence set. Our algorithm mainly replaces the “independent set” in [14] with the “relaxed independent set”. In each relaxed independent set, we no longer choose a disk but choose $t - 1$ disks, where the integer $t \in \{2, 3, \dots\}$. So we get the following critical Lemma 1.

Lemma 1 (Lemma 1 of [7]). *For any integer $t \in \{2, 3, \dots\}$, any user $u \in U$ is covered by at most $t - 1$ disks in the $2\sin\frac{\pi}{t}$ -relaxed independent set.*

Our algorithm is based on the algorithm of [7], which is mainly divided into the following three stages.

In the first stage, in order to maintain the feasibility of the duality. Start with the trivial dual feasible solution 0, and simultaneously increase all dual variables $\{y_u\}_{u \in U}$ until some disk D becomes tight or some user set T becomes tight. If a disk D becomes tight, add D to the set of tight disks \mathcal{D}^{tight} , and freeze the dual values of all users in $U(D)$. Otherwise, a user set T becomes tight, and all users in T are added to the temporary rejected user set R^{temp} , and the dual values of all users in T are frozen. The above process iterates until the dual values of all users in U are frozen.

In the second stage, for any disk $D \in \mathcal{D}^{tight}$, if there is a disk D' of $r(D') \geq r(D)$ such that $U(D') \cap U(D) \neq \emptyset$ and

$$d(c(D'), c(D)) \leq 2\sin\frac{\pi}{t}r(D'), t \in \{2, 3, \dots\}.$$

we remove D from \mathcal{D}^{tight} .

In the third stage, set $I := \mathcal{D}^{tight}$. For any integer $t \in \{2, 3, \dots\}$, expand the radius of all disks in I by a factor of $1 + 2\sin\frac{\pi}{t}$. Set such a disk set as F and a penalty user set as $R = U \setminus U(F)$, and finally the algorithm outputs (F, R) . Our algorithm pseudocode is Algorithm 1:

Algorithm 1 Primal Dual Algorithm

Input: An instance (U, S, p, π) .
Output: (F, R) .

- 1 Initially, for any user $u \in U, y_u = 0$. Set $\mathcal{D}^{tight} = I = R^{temp} = \emptyset$.
- 2 **while** $U \neq \emptyset$ **do**
- 3 Simultaneously increase the dual variable y_u of all users $u \in U$ until some disk D becomes tight or some user set T becomes tight.
- 4 **if** disk D gets tight **then**
- 5 $\mathcal{D}^{tight} := \mathcal{D}^{tight} \cup \{D\}$ and $U := U \setminus U(D)$.
- 6 **else**
- 7 $R^{temp} := R^{temp} \cup T$ and $U := U \setminus T$.
- 8 **for** $D \in \mathcal{D}^{tight}$ **do**
- 9 **if** There exists a disk D' with a larger radius that satisfies $U(D) \cap U(D') \neq \emptyset$ and $d(c(D), c(D')) \leq 2\sin\frac{\pi}{t}r(D'), t \in \{2, 3, \dots\}$. **then**
- 10 $\mathcal{D}^{tight} := \mathcal{D}^{tight} \setminus \{D\}$.
- 11 Set $I := \mathcal{D}^{tight}$.
- 12 **for** $D \in I$ **do**
- 13 $p(c(D)) := c \cdot ((1 + 2\sin\frac{\pi}{t})r(D))^\alpha$.
- 14 Set $F := \{Disk(s, (1 + 2\sin\frac{\pi}{t})r(D)) | D \in I\}, R := U \setminus U(F)$. Output (F, R) .

To facilitate the understanding of the algorithm, we give a simple example (see Figure 1). $U = \{u_1, u_2, u_3, u_4, u_5\}, S = \{s_1, s_2, s_3\}$. For any subset $T \subseteq U$, if $|T| = 1, \pi(T) = 3; |T| = 2, \pi(T) = 4; |T| \geq 3, \pi(T) = 5$. We set $c = 1, \alpha = 2, t = 6$. Initially, we set the dual value $y_{u_1} = y_{u_2} = y_{u_3} = y_{u_4} = y_{u_5} = 0$ for all users. We execute the algorithm and simply describe it in the following five steps.

- (a). Increase the dual value $\{y_u\}_{u \in U}$ of all users $u \in U$ simultaneously until disk D_1 becomes tight, $\mathcal{D}^{tight} = \{D_1\}$, and the values of y_{u_3} and y_{u_4} are frozen.
- (b). Simultaneously increase the dual values of users u_1, u_2, u_5 of $y_{u_1}, y_{u_2}, y_{u_5}$ until the subset $T = \{u_1, u_2\}$ becomes tight, set $R^{temp} = \{u_1, u_2\}$, and the values of y_{u_1}, y_{u_2} are frozen.
- (c). Increase the values of y_{u_5} until the disk D_2 becomes tight, set $\mathcal{D}^{tight} = \{D_1, D_2\}$, and the value of y_{u_5} is frozen.
- (d). Because $U(D_1) \cap U(D_2) \neq \emptyset$ and $d(D_1, D_2) \leq r(D_2)$, D_1 is removed from \mathcal{D}^{tight} , setting $I := \mathcal{D}^{tight} = \{D_2\}$.
- (e). Expand the radius of $D_2 \in I$ by twice to become D_2' , set $R = \{u_1\}$ and $F = \{D_2'\}$. Therefore, we obtain a feasible solution (F, R) .

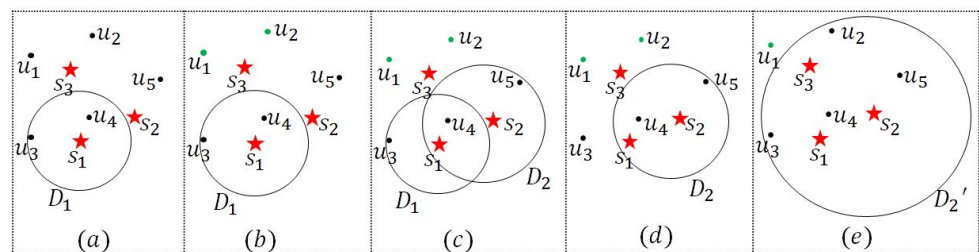


Figure 1. A simple example $U = \{u_1, u_2, u_3, u_4, u_5\}, S = \{s_1, s_2, s_3\}$ and $t = 6$.

Lemma 2 (Lemma 3.1 of [14,15]). *Algorithm 1 can be performed in polynomial time.*

Lemma 3. *(F, R) is a feasible solution to the SPMPC problem.*

Proof. Let $U(I)$ and $U(F)$ represent the set of users covered by the union of all disks in I and F , respectively. According to Algorithm 1, we know that $R^{temp} \cup U(\mathcal{D}^{tight}) = U$. For any user $u \in U(I)$, we have $u \in U(F)$. Then for any user $u \in U(I)$, u is covered by the disk in F .

Otherwise, u is covered by the disk removed in \mathcal{D}^{tight} , that is, for any user $u \in D' \subseteq \mathcal{D}^{tight} \setminus I$, there is a disk $D \in I$ satisfies $U(D') \cap U(D) \neq \emptyset$ and $r(D) \geq r(D')$, then according to the triangle inequality, we have

$$\begin{aligned} d(u, c(D)) &\leq d(u, c(D')) + d(c(D'), c(D)) \\ &\leq r(D') + 2\sin\frac{\pi}{t}r(D) \\ &\leq (1 + 2\sin\frac{\pi}{t})r(D) \end{aligned}$$

that is, for any integer $t \in \{2, 3, \dots\}$, expanding the radius of the disk D by $1 + 2\sin\frac{\pi}{t}$ times can cover the user u . Therefore, for any integer $t \in \{2, 3, \dots\}$, enlarging all disks $D \in I$ by $1 + 2\sin\frac{\pi}{t}$ times can cover all users in \mathcal{D}^{tight} , that is, F can cover all users in \mathcal{D}^{tight} . Since $R = U \setminus U(F)$ and $R \subseteq R^{temp}$, so (F, R) is a feasible solution. \square

Lemma 4. The temporary rejection point set R^{temp} satisfies $\pi(R^{temp}) = \sum_{u:u \in R^{temp}} y_u$.

Proof. Consider all tight subsets T_1, \dots, T_l formed by all users in R^{temp} , i.e., $\bigcup_{i=1}^l T_i = R^{temp}$. According to the Algorithm 1, for any subset $T_i, i \in \{1, \dots, l\}, \bigcap_{i=1}^l T_i = \emptyset$, we have

$$\pi\left(\bigcap_{i=1}^l T_i\right) = 0. \tag{4}$$

Since any subset $T_i, i \in \{1, \dots, l\}$ is a tight set, there is $\pi(T_i) = \sum_{u:u \in T_i} y_u$. According to the (4) and submodularity of $\pi(\cdot)$, there is $\sum_{i=1}^l \pi(T_i) \geq \pi(\bigcup_{i=1}^l T_i)$, i.e.,

$$\sum_{i=1}^l \sum_{u:u \in T_i} y_u = \sum_{u:u \in \bigcup_{i=1}^l T_i} y_u \geq \pi\left(\bigcup_{i=1}^l T_i\right).$$

Because of dual programming (3), for any $\bigcup_{i=1}^l T_i \subseteq U$, there is

$$\sum_{u:u \in \bigcup_{i=1}^l T_i} y_u \leq \pi\left(\bigcup_{i=1}^l T_i\right).$$

Therefore, we have $\sum_{u:u \in \bigcup_{i=1}^l T_i} y_u = \pi(\bigcup_{i=1}^l T_i)$, i.e., $\sum_{u:u \in R^{temp}} y_u = \pi(R^{temp})$. \square

According to Lemma 1, we know that for any user $u \in U$ can be covered by at most $t - 1$ disks in I , so we can get the following lemma.

Lemma 5. For any integer $t \in \{2, 3, \dots\}$, $p(I) \leq (t - 1)OPT$, where OPT is the optimal value of the SPMPC problem.

Proof. According to Algorithm 1, for any $D \in I \subseteq \mathcal{D}^{tight}$, there is $\sum_{u:u \in D} y_u = c \cdot r(D)^\alpha$, then

$$\begin{aligned} p(I) &= \sum_{D:D \in I} c \cdot r(D)^\alpha \\ &= \sum_{D:D \in I} \sum_{u:u \in D} y_u \\ &\leq (t-1) \sum_{u:u \in U(I)} y_u \\ &\leq (t-1) \sum_{u:u \in U} y_u \\ &\leq (t-1)OPT \end{aligned}$$

where the first inequality is due to Lemma 1. \square

Lemma 6. For any integer $t \in \{2, 3, \dots\}$, $p(F) \leq (t-1)(1 + 2\sin\frac{\pi}{t})^\alpha OPT$, where OPT is the optimal value of the SPMPC problem.

Proof. According to Algorithm 1, we know $F = \{Disk(s, (1 + 2\sin\frac{\pi}{t})r(D)) | D \in I\}$. Therefore, there is

$$\begin{aligned} p(F) &= \sum_{D:D \in F} c \cdot r(D)^\alpha \\ &= (1 + 2\sin\frac{\pi}{t})^\alpha p(I) \\ &\leq (t-1)(1 + 2\sin\frac{\pi}{t})^\alpha OPT \end{aligned}$$

where the last inequality is based on the Lemma 5. \square

Theorem 1. Algorithm 1 obtains an $O(\alpha)$ -approximate solution to the SPMPC problem.

Proof. Since the output of the Algorithm 1 is (F, R) , then

$$\begin{aligned} p(F) + \pi(R) &\leq (t-1)(1 + 2\sin\frac{\pi}{t})^\alpha OPT + \pi(R) \\ &\leq (t-1)(1 + 2\sin\frac{\pi}{t})^\alpha OPT + \pi(R^{temp}) \\ &= (t-1)(1 + 2\sin\frac{\pi}{t})^\alpha OPT + \sum_{u:u \in R^{temp}} y_u \\ &\leq ((t-1)(1 + 2\sin\frac{\pi}{t})^\alpha + 1)OPT \end{aligned}$$

where the second inequality is due to $R \subseteq R^{temp}$.

For each given α , let $t^* = \arg \min_{t \in \{2, 3, 4, \dots\}} ((t-1)(1 + 2\sin\frac{\pi}{t})^\alpha + 1)$, then there is a minimum value $(t^* - 1)(1 + 2\sin\frac{\pi}{t^*})^\alpha + 1$, which is always optimal compared with $3^\alpha + 1$ in [14]. The comparison of the approximate ratio of our algorithm and the approximate ratio $3^\alpha + 1$ of the algorithm in [14] is depicted in Figure 2.

Let $g(t) = (t-1)(1 + 2\sin\frac{\pi}{t})^\alpha + 1$, similar to the reference [7], we have

$$\lim_{\alpha \rightarrow +\infty} \frac{g(\lfloor \alpha \rfloor)}{\alpha} = \lim_{\alpha \rightarrow +\infty} \frac{\lfloor \alpha \rfloor - 1}{\alpha} (1 + 2\sin\frac{\pi}{t})^\alpha + \lim_{\alpha \rightarrow +\infty} \frac{1}{\alpha} = e^{2\pi}.$$

when α is large enough, $g(t)$ is almost linear in the α value, and we have the minimum value $(t^* - 1)(1 + 2\sin\frac{\pi}{t^*})^\alpha + 1 \approx e^{2\pi} \alpha = O(\alpha)$.

\square

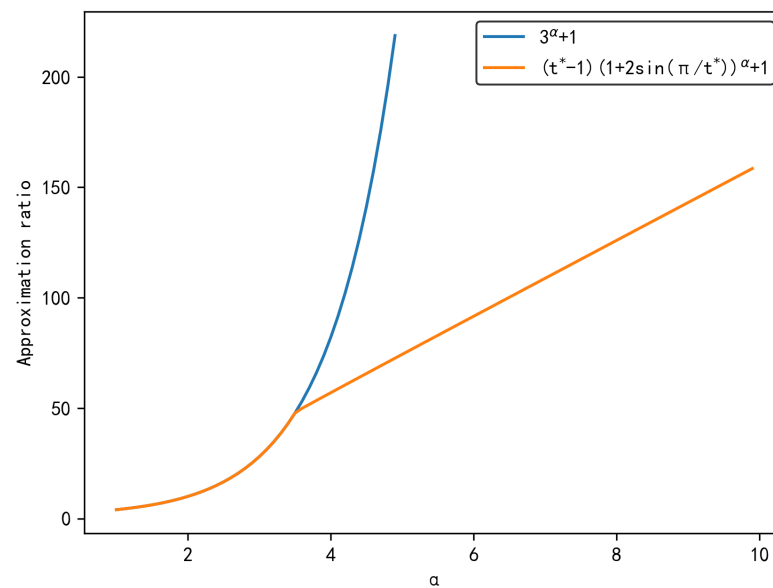


Figure 2. Comparison of two approximate ratios.

5. Conclusions

In this paper, we study minimum power cover problem with submodular penalty. Using the concept of relaxed independent sets of [7], based on the primal dual technique, we design an $O(\alpha)$ -approximation algorithm. Our result is an improvement on the minimum power cover problem with submodular penalty of [14].

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Lev-Tov, N.; Peleg, D. Polynomial time approximation schemes for base station coverage with minimum total radii. *Comput. Netw.* **2005**, *47*, 489–501. [\[CrossRef\]](#)
- Charikar, M.; Panigrahy, R. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.* **2004**, *68*, 417–441. [\[CrossRef\]](#)
- Zhang, Q.H.; Li, W.D.; Su, Q.; Zhang, X.J. A local-ratio-based power control approach for capacitated access points in mobile edge computing. In Proceedings of the 6th International Conference on High Performance Compilation, Computing and Communications, Jilin, China, 23–25 June 2022; pp. 175–182.
- Zhang, Q.H.; Li, W.D.; Su, Q.; Zhang, X.J. A primal-dual based power control approach for capacitated edge servers. *Sensors* **2022**, *22*, 7582. [\[CrossRef\]](#) [\[PubMed\]](#)
- Freund, A.; Rawitz, D. Combinatorial interpretations of dual fitting and primal fitting. In Proceedings of the International Workshop on Approximation and Online Algorithms, Lisbon, Portugal, 6–10 September 2003; Springer, Berlin/Heidelberg, Germany, 2003; pp. 137–150.
- Li, M.H.; Ran, Y.L.; Zhang, Z. A primal dual algorithm for the minimum power partial cover problem. *J. Comb. Optim.* **2022**, *44*, 1913–1923. [\[CrossRef\]](#)
- Dai, H.; Deng, B.; Li, W.D.; Liu, X.F. A note on the minimum power partial cover problem on the plane. *J. Comb. Optim.* **2022**, *44*, 970–978. [\[CrossRef\]](#)
- Abu-Affash, A.K.; Carmi, P.; Katz, M.J.; Morgenstern, G. Multi cover of a polygon minimizing the sum of areas. In Proceedings of the International Workshop on Algorithms and Computation, New Delhi, India, 18–20 February 2011; pp. 685–698.
- Bar-Yehuda, R.; Rawitz, D. A note on multicovering with disks. *Comput. Geom.* **2013**, *46*, 394–399. [\[CrossRef\]](#)
- Liang, W.; Li, M.H.; Zhang, Z.; Huang, X.H. Minimum power partial multi-cover on a line. *Theor. Comput. Sci.* **2021**, *864*, 118–128. [\[CrossRef\]](#)
- Ran, Y.L.; Huang, X.H.; Zhang, Z.; Du, D.Z. Approximation algorithm for minimum power partial multi-cover in wireless sensor networks. *J. Glob. Optim.* **2021**, *80*, 661–677. [\[CrossRef\]](#)
- Liu, X.F.; Li, W.D.; Xie, R.T. A primal dual approximation algorithm for the k -prize-collecting minimum power cover problem. *Optim. Lett.* **2022**, *16*, 2373–2385. [\[CrossRef\]](#)

13. Liu, X.F.; Dai, H.; Li, S.Z.; Li, W.D. k -prize-collecting minimum power cover problem with submodular penalties on a plane. *Chin. Sci. Inf. Sci.* **2022**, *52*, 947–959.
14. Liu, X.F.; Li, W.D.; Dai, H. Approximation algorithms for the minimum power cover problem with submodular/linear penalties. *Theor. Comput. Sci.* **2022**, *923*, 256–270. [[CrossRef](#)]
15. Fujishige, S. *Submodular Functions and Optimization*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2005; Volume 58, pp. 1–395.