

A Local-Ratio-Based Power Control Approach for Capacitated Access Points in Mobile Edge Computing

QINGHUI ZHANG, Yunnan University, China

WEIDONG LI, Yunnan University, China

QIAN SU, Yunnan University, China

XUEJIE ZHANG*, Yunnan University, China

Terminal devices (TDs) connect to networks through access points (APs) integrated into the edge server. This provides a prerequisite for TDs to upload tasks to cloud data centers or offload them to edge servers for execution. In this process, signal coverage, data transmission, and task execution consume energy, and the energy consumption of signal coverage increases sharply as the radius increases. Lower power leads to less energy consumption in a given time segment. Thus, power control for APs is essential for reducing energy consumption. Our objective is to determine the power assignment for each AP with same capacity constraints such that all TDs are covered, and the total power is minimized. We define this problem as a *minimum power capacitated cover* (MPCC) problem and present a *minimum local ratio* (MLR) power control approach for this problem to obtain accurate results in polynomial time. Power assignments are chosen in a sequence of rounds. In each round, we choose the power assignment that minimizes the ratio of its power to the number of currently uncovered TDs it contains. In the event of a tie, we pick an arbitrary power assignment that achieves the minimum ratio. We continue choosing power assignments until all TDs are covered. Finally, various experiments verify that this method can outperform another greedy-based way.

CCS Concepts: • **Networks** → **Wireless access networks**; • **Theory of computation** → *Design and analysis of algorithms*.

Additional Key Words and Phrases: Mobile Edge Computing, Minimum Power Cover, Local Ratio, Power Control

ACM Reference Format:

Qinghui Zhang, Weidong Li, Qian Su, and Xuejie Zhang. 2022. A Local-Ratio-Based Power Control Approach for Capacitated Access Points in Mobile Edge Computing. In *HP3C '22: International Conference on High Performance Compilation, Computing and Communication, June 23–25, 2022, Jilin, China*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3546000.3546027>

1 INTRODUCTION

With the rapid development of beyond 5G/6G and the Internet of Things, increasing number of terminal devices (TD) are being deployed at the edge of networks [22]. Even with advances in network technology, data centers cannot guarantee acceptable latency and network unobstructed. By placing the computing and storage resources closer to the TDs, mobile edge computing (MEC) can significantly increase performance in terms of low latency, reduced communications overhead, and high-quality user experience[1].

Nowadays, many tasks of TDs will be prioritized to migrate to servers close to TDs for computing or storage, unless these tasks can only be handled in cloud data centers. By the way, a TD usually establishes network connection with edge server through its access point (AP, it can be an integral part of the edge server itself). This provides a prerequisite

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

Manuscript submitted to ACM

for TDs to upload tasks to cloud data centers or offload them to edge servers for execution. During these processes, signal coverage, data transmission, task execution, etc., cost energy. Therefore, in recent years, considerable research on reducing energy consumption has been conducted, and power control is an important topic.

In typical wireless networks, the signal coverage of an AP, which is determined by its power, is a disk area centered on it. Because the signal intensity will attenuate with distance, a larger signal coverage disk needs more power[18]. Many power control methods have been proposed for different wireless networks. About power control of cellular network, [6] considered the sum spectral efficiency (SE) optimization problem in multi-cell Massive MIMO systems with a varying number of active users. The neural network they proposed, PowerNet, only uses the large-scale fading information to predict both the pilot and data powers. In [10], Dai *et al.* investigated the joint optimization of BS clustering and power control for NOMA-enabled CoMP transmission in dense cellular networks to maximize system sum-rate. In addition, the scope of [7] was to investigate how machine learning may be used to bring wireless nodes to the lowest possible transmission power level and, in turn, to respect the quality requirements of the overall network. Lowering transmission power has benefits in terms of both energy consumption and interference.

However, both upload bandwidth and server's computing resources are limited. This provides us a new perspective to power control for edge wireless network. In [20], Wu *et al.* first studied the joint multiuser offloading and transmission power control optimization problem in a multi-channel wireless interference scenario. They then proposed an efficient semi-distributed algorithm consisting of two subalgorithms of offloading scheme and transmission power control.

All the methods stated above guarantee QoS will not deteriorate due to interference. The same is true in this paper. We build a signal coverage model simplified communication process and paid attention to the location of facilities and the capacity of AP. Our objective is to determine the power assignment in MEC such that APs with limited capacity can cover all TDs and minimize the total power. We define this problem as the *minimum power capacitated cover* (MPCC) problem, which is an NP-hard problem. A local ratio approach is proposed to solve it. Finally, various experiments verify that this method can outperform another greedy-based way.

1.1 Related Works

The problem of resource allocation has been paid much attention in various fields, and it has many different optimization objectives. Furthermore, resource allocation problems are usually NP-hard problems, and how to get satisfactory results in a short time is a considerable challenge. Aiming at social welfare maximization, reference [21] proposed an auction mechanism for resource allocation in the Cloud Edge collaboration framework based on live video webcast services. Reference [16] addressed the problem of heterogeneous physical machines resource management (HPMRM) in heterogeneous clouds. Trade-offs between energy and performance are important for energy-aware scheduling. In reference [13], a polynomial-time algorithm was designed for energy-aware profit maximizing scheduling problem.

MPCC is a fundamental *minimum power coverage* (MPC) problem that has been extensively studied in the past two decades. Originally, the problem was introduced for modeling the power control problem in the communication of edge computing. Broadly, MPCC belongs to the family of *minimum weight set cover* (MWSC) problems. MPCC can be viewed as a special variant of the MWSC that has the following general formulation: Given a universe E of elements (points or terminal devices) and a family \mathcal{G} of ranges (or geometric objects like disks and polygons) where each range has a weight assignment $w : \mathcal{G} \mapsto \mathbb{R}^+$, determine a set $\mathcal{F} \subseteq \mathcal{G}$ of ranges such that every element in E is covered by at least one range in \mathcal{F} and that the total weight $\sum_{G \in \mathcal{F}} w(G)$ is minimized. Clearly, MPCC can be formulated as a special MWSC, where \mathcal{G} is the set of all possible disks centered at some APs having at least one TD on their boundaries. The difference is that the capacity of each $a \in A$ is limited to $k(a)$; therefore, the capacity of the disk centered at a in \mathcal{G} is also constrained.

The MWSC problem is, in general, challenging to solve optimally, even for some simple versions. For example, Alt *et al.* and Bilò *et al.* present a minimum cost covering problem without capacity constraint in [2, 3], which is still NP-hard for any $\alpha > 1$. Thus, finding polynomial time approximation algorithms is the main objective for MPCC problems.

For the *geometric minimum weight set cover* problem, it was studied from different aspects. For the minimum disk cover problem, in which disks may have different sizes, Mustafa and Ray designed a PTAS using a local search method [17]. Considering weight, Varadarajan [19] presented a clever quasi-uniform sampling technique that was improved by Chan *et al.* [4], yielding a constant approximation for the minimum weight disk cover problem. About partial cover problem, Liu *et al.* [15] introduced the *k-prize-collecting minimum power cover problem* (*k-PCPC*), and present a novel two-phase primal-dual algorithm and got an approximation ratio of at most 3^α . The author then introduced the minimum power cover problem with submodular and linear penalties in [14] and presented a combinatorial primal-dual $(3^\alpha + 1)$ -approximation algorithm.

In previous studies, the minimum power cover problem has rarely been considered for capacity constraints. However, for the vertex cover problem, another research topic of the set cover problem, many studies consider the minimum capacitated vertex cover problem. Consider a graph $G = (V, E)$ with weights and capacities on the vertices. All edges must be covered by vertices under the capacity constraint, and the objective is to minimize the weight. [11] give a primal-dual-based approximation algorithm with an approximation guarantee of 2 (every edge has two vertices). The study of the vertex cover problem with hard capacity constraints (VC-HC) was initiated in [8]. They established a surprising result that, while this setting admits constant factor approximations, it becomes set-cover hard when $\{0, 1\}$ -weighted vertices are considered. [12] considered VC-HC on hypergraphs. This literature, which considers a hypergraph G with a maximum edge size of f , provides an iterative partial rounding technique and obtains an f -approximation, improving upon the previous results of [5].

2 SYSTEM MODEL

In this section, we present the system model and a corresponding integer programming model. We focus on power control with capacitated APs and covering all TDs to minimize total power.

2.1 Model Description

Consider m APs with same capacity and n TDs randomly distributed in 2-dimensional space. Each TD accesses the edge network through its covering AP. Let (U, A, k) be an instance of the MPCC problem with $U = \{1, 2, \dots, n\}$ and $A = \{1, 2, \dots, m\}$, and let $k \in \mathbb{Z}^+$ denote the capacity of IP to each AP $a \in A$. Each AP a can adjust its own power, and the relationship between the power $p(a)$ and the radius $r(a)$ of its service area is

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

where c and α are constants (α is usually called the *attenuation factor*). The center and radius can be used to define a disk, so a and $u \in U$ can determine a unique disk D_{au} whose radius is precisely equal to the distance between a and u (u is on the boundary of D_{au} in the optimal case). Therefore, at most mn disks must be considered. We denote the set of such disks by \mathcal{D} . Each instance corresponds to a \mathcal{D} , and we use D to represent any disk in \mathcal{D} . The set of disks with a as the center is a subset of \mathcal{D} , denoted by $\mathcal{D}(a)$. To simplify the notation, we use D to represent both a disk in \mathcal{D} and the set of TDs contained in D and use $r(D)$ and $p(D)$ to denote the radius and power of disk D , where $p(D) = c \cdot r(D)^\alpha$. The mapping $k(\cdot) : \mathcal{D} \mapsto k$ can determine the capacity of $D \in \mathcal{D}$. Notably, u *contained* in D means that u is within the range of D ($u \in D$) and u is *covered* by a means that u gains an IP of a to access the edge network (the former does not

occupy the resources of the corresponding AP). Clearly, the capacity of $D \in \mathcal{D}(a)$ formed by different power strategies of one AP is equal and changes together. Table 1 summarizes the notation for reference.

Fig. 1 shows an instance with three APs and six TDs. The capacities of AP1, AP2 and AP3 are all 2, respectively (the first number in brackets indicates the AP index, the following represents its capacity). The disk formed by AP1 contains TDs 1, 2 and 4, TDs 3,5 is contained by the disk formed by AP2, and TDs 4, 5 and 6 are contained by AP3's disk. Notably, TD4 and TD5 are contained by two disks simultaneously. Since the capacity of all APs are only 2, TD4 is eventually covered by AP3 and TD5 is covered by AP2.

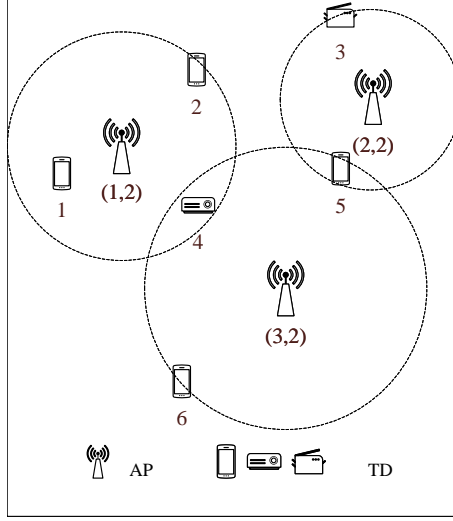


Fig. 1. An instance of the MPCC problem.

In practical situations, a TD may locate outside the maximum coverage that all APs can form, or the number of TDs exceeds the total capacity of all APs. We assume that an AP has no upper power limit and that the entire system has enough capacity to cover all the TDs. In this way, we can obtain a scheme covering all TDs through the technique in this paper, which can provide a reference for the later decision on whether to cover some distant TDs. We can also solve the instance where the demand exceeds the capacity by using the method of this paper many times. Therefore, the above two assumptions are reasonable.

2.2 Integer Program for the MPCC Problem

For a subset $\bar{\mathcal{D}}$ of \mathcal{D} , $\bar{\mathcal{D}}$ is a feasible solution to the MPCC problem and must meet the following conditions:

- (1) **Full coverage condition.** $C(\bar{\mathcal{D}}) = \bigcup_{D \in \bar{\mathcal{D}}} D$ denotes the TDs covered by $\bar{\mathcal{D}}$; then, $C(\bar{\mathcal{D}}) = U$ must be established.
- (2) **Limited capacity condition.** Since the IP capacity of each AP is limited, the number of TDs covered by an AP must be less than its capacity.
- (3) **Power uniqueness.** For any $a \in A$, there are n power strategies, but at most one strategy is selected; that is, there is at most one disk with a as the center in $\bar{\mathcal{D}}$.

Table 1. Notation

Notation	Description
z_D	integer variable indicating whether D is selected, $z_D \in \{0, 1\}$
y_{uD}	integer variable indicating whether u is covered by D , $y_{uD} \in \{0, 1\}$
$p(D)$	power of D
$k(D)$	capacity of D
$\mathcal{D}(a)$	set of disks centered on a
d_D	number of TDs contained in D
\hat{k}_D	capacity of D and varieties during the process
\hat{p}_D	power increment of D and varieties during the process
l_a	latest broken disk for a
C_a	set of TDs covered by a
e_D	local ratio of D
D^*	disk with minimum local ratio
a^*	central AP of D^*
D_{rmv}	set of disks that need to be removed

According to the above three conditions, we can formulate an integer programming model for the MPCC problem. Variable $z_D \in \{0, 1\}$ indicates whether $D \in \mathcal{D}$ is selected; that is, $z_D = 1$ if and only if D is selected. Variable $y_{uD} \in \{0, 1\}$ indicates whether $u \in U$ is covered by D , where $y_{uD} = 1$ if and only if u is covered by D . The following is the integer program:

$$\min \sum_{D \in \mathcal{D}} p(D) \cdot z_D \quad (2)$$

$$s.t. \sum_{D: u \in D} y_{uD} \geq 1, u \in U \quad (3)$$

$$k(D)z_D - \sum_{u: u \in D} y_{uD} \geq 0, D \in \mathcal{D} \quad (4)$$

$$z_D \geq y_{uD}, u \in D \in \mathcal{D} \quad (5)$$

$$\sum_{D \in \mathcal{D}(a)} z_D \leq 1, a \in A \quad (6)$$

$$z_D \in \{0, 1\}, y_{uD} \in \{0, 1\}, \forall u \in U, \forall D \in \mathcal{D} \quad (7)$$

Constraint (3) corresponds to the full coverage condition, (4) corresponds to the limited capacity condition, and (6) corresponds to the power uniqueness. Constraint (5) is an implicit condition in this integer program. If u is covered by D in the final result set, then D must be selected as the power strategy of its center AP.

3 A LOCAL RATIO APPROACH FOR MPCC

A local ratio approach is any algorithm that follows the problem-solving heuristic of making the locally optimal choice in each stage[9]. It is a greedy-based strategy. In many problems, the greedy strategy does not produce an optimal solution, but a greedy heuristic can yield locally optimal solutions that approximate a globally optimal solution in a reasonable amount of time. Since the MPCC problem is NP-hard, it is a considerable challenge to obtain an optimal

Algorithm 1 MLR

Input: An instance of MPCC (U, A, k) , a set of disks \mathcal{D} , a mapping of power $p : \mathcal{D} \mapsto \mathbb{R}^+$.

Output: A set of selected disks $\tilde{\mathcal{D}}$ and the set of coverage results for all APs C .

```

1:  $d_D \leftarrow |D|, \forall D \in \mathcal{D}$ .
2:  $\hat{k}_D \leftarrow k(D), \forall D \in \mathcal{D}$ .
3:  $\hat{p}_D \leftarrow p(D), \forall D \in \mathcal{D}$ .
4:  $l, C \leftarrow \text{ChooseDisks}(U, d, \hat{k}, \hat{p})$ .
5: for each  $a \in A$  do
6:   if  $l_a \neq \emptyset$  then
7:      $\tilde{\mathcal{D}} \leftarrow \tilde{\mathcal{D}} \cup l_a$ .
8:   end if
9: end for
10: return  $\tilde{\mathcal{D}}, C$ .

```

solution for integer programming (2). Therefore, we propose a local ratio approach named *minimum local ratio* (MLR), which can obtain an approximate solution for MPCC.

In a practical scenario, when an AP cannot serve two or more TDs that have the same distance from it simultaneously, the AP will determine which one to serve first based on factors such as the TD's request sequence or the urgency of the task. How to determine which TD to serve in this case is not the focus of this paper. Thus, we present definition 1 to define the power partial order relationship between two disks. After the partial order relationship between the two disks with the same radius determined by the two TDs is clarified, who the AP should serve first becomes evident. Map the positions of all TDs and APs to a two-dimensional coordinate system, and use $\cos(\vec{au})$ to represent the cosine of the angle between vector \vec{au} and the x -axis formed by a and u .

Definition 1. *The distance between TDs u and v from AP a is the same; that is, $r(D_{au}) = r(D_{av})$. If $\cos(\vec{au}) > \cos(\vec{av})$, then $r(D_{au}) > r(D_{av})$, and the same $p(D_{au}) > p(D_{av})$. Although $r(D_{au}) = r(D_{av})$, D_{au} contains v but D_{av} does not contain u .*

3.1 Minimum Local Ratio Algorithm

In Algorithm 1, we describe the overall logic of the MLR algorithm in the form of pseudocode. Its input is an MPCC problem instance (U, A, k) , a set of disks \mathcal{D} and a mapping of power p . \mathcal{D} is a candidate set from which a solution is created. After the input, we define some key variables. d_D records the number of TDs that D can currently contain (as the algorithm progresses, d_D changes). Let \hat{k}_D denote the capacity of D , which is initialized to $k(D)$. This capacity may change with the continuous allocation of TDs. The variable \hat{p}_D is the power increment of D , which is initialized to $p(D)$. We define l as the set of disks finally selected for all APs, where l_a is the disk finally selected for a , so $l = \bigcup_{a \in A} l_a$. Similarly, C_a records the set of TDs that are eventually covered by a , and $C = \bigcup_{a \in A} C_a$. After Algorithm 2, we can obtain the results of l and C , and the final result $\tilde{\mathcal{D}}$ is the set of non-null values in l .

In the main loop of Algorithm 2, each iteration calculates the local ratio e_D corresponding to $D \in \mathcal{D}$ in the current iteration state. On the basis of lemma 1, the disk D^* that has the minimum local ratio in each iteration satisfies the inequality $d_{D^*} \leq \hat{k}_{D^*}$. When D^* is selected, we update the value of l_{a^*} to D^* , where a^* is the center of D^* , and then assign the TDs contained by D^* in the current iteration to a^* for coverage.

Algorithm 2 ChooseDisks**Input:** U, d, \hat{k}, \hat{p} .**Output:** l, C

```

1:  $l_a \leftarrow \emptyset, \forall a \in A.$  ( $l_a$  is the latest broken disk for  $a$ )
2:  $C_a \leftarrow \emptyset, \forall a \in A.$  ( $C_a$  is the set of TDs covered by  $a$ )
3: while  $U \neq \emptyset$  do
4:    $e_D \leftarrow \hat{p}_D / \min(\hat{k}_D, d_D), \forall D \in \mathcal{D}.$ 
5:    $D^* \leftarrow \arg \min(e_D : D \in \mathcal{D})(d_D \leq \hat{k}_D \text{ is established}).$ 
6:    $a^* \leftarrow$  the central AP of  $D^*.$ 
7:    $l_{a^*} \leftarrow D^*.$ 
8:    $C_{a^*} \leftarrow C_{a^*} \cup D^*.$  (Make TDs in  $D^*$  covered by  $a^*$ )
9:   if  $d_{D^*} = \hat{k}_{D^*}$  then
10:     $\mathcal{D}_{rmv} \leftarrow \{D : D \in \mathcal{D}(a^*)\}.$ 
11:   elseif  $d_{D^*} < \hat{k}_{D^*}$ 
12:     $\mathcal{D}_{rmv} \leftarrow \{D : D \in \mathcal{D}(a^*) \text{ and } r(D) < r(D^*)\} \cup \{D^*\}.$ 
13:   end if
14:    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_{rmv}.$ 
15:    $\hat{p}_D \leftarrow \hat{p}_D - e_{D^*} \min(\hat{k}_D, d_D), \forall D \in \mathcal{D}.$ 
16:   for each  $u^* \in D^*$  do
17:      $U \leftarrow U \setminus \{u^*\}.$ 
18:      $D \leftarrow D \setminus \{u^*\}, \forall D \in \mathcal{D} : u^* \in D$ 
19:      $\hat{k}_D \leftarrow \hat{k}_D - 1, \forall D \in \mathcal{D}(a^*)$ 
20:     if  $d_D = 0$  or  $\hat{k}_D = 0, \forall D \in \mathcal{D}$  then
21:        $\mathcal{D} := \mathcal{D} \setminus \{D\}.$ 
22:     end if
23:   end for
24: end while
25: return  $l, C.$ 

```

LEMMA 1. *The disk D^* selected in each iteration in Algorithm 2 must satisfy $d_{D^*} \leq \hat{k}_{D^*}$.*

PROOF. Assuming that the D^* selected in an iteration satisfies $d_{D^*} > \hat{k}_{D^*}$, we know that $e_{D^*} = \hat{p}_{D^*} / \hat{k}_{D^*}$. Then, there must be a disk $D^-, p(D^-) < p(D^*)$, that happens to have $d_{D^-} = \hat{k}_{D^-}$. In the current iteration $\hat{p}_{D^-} < \hat{p}_{D^*}$ and $e_{D^-} < e_{D^*}$, which is contrary to the hypothesis. \square

In the second part of the main loop of Algorithm 2, all the relevant variables are updated. If $d_{D^*} = \hat{k}_{D^*}$ in the current iteration, all the disks in \mathcal{D} with a^* as the center are removed. If $d_{D^*} < \hat{k}_{D^*}$, then there are remaining IP resources for a^* ; we remove these disks in \mathcal{D} , with a^* as the center and a smaller radius, and D^* itself. For line 12 of Algorithm 2, the power increment of all remaining disks in \mathcal{D} is updated. We remove all the TDs allocated for the current iteration from the related set in the `for` loop. We also remove all disks from \mathcal{D} where the number of TDs contained in the disk is 0 or its capacity is 0. Finally, l can record the disk for each AP (an AP may not choose any disk).

Theorem 2. *MLR computes a feasible solution to ILP 2 in polynomial time.*

PROOF (CORRECTNESS). *MLR* outputs a feasible solution for ILP 2 because the `while` loop body in *ChooseDisks* guarantees all TDs can be covered by APs. At first, according to Lemma 1, the disk D^* selected in each iteration can

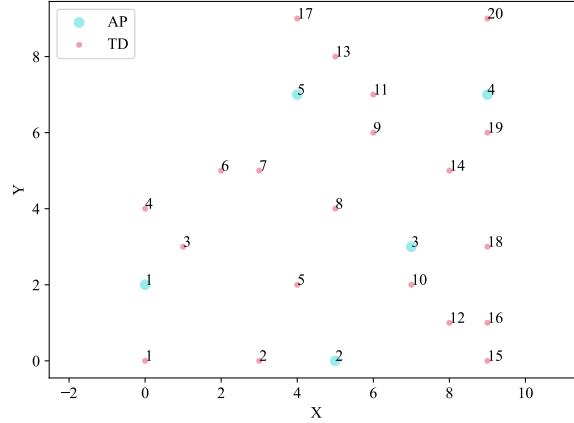


Fig. 2. An instance of using the MLR method to solve the MPCC problem

guarantee the capacity to cover the TDs contained in D^* . Second, because the capacities of the disks with the same center AP change together, the latest disk D^{last} of $a \in A$ can cover the TDs covered previously. In other words, when D^{last} is selected, it can also cover the TDs previously assigned to a . The former two points can satisfy constraints (4), (5) and (6). We assume that the total capacity of the system can cover all TDs, and \mathcal{D} is formed based on all TDs and APs. In theory, any TDs can be covered by any AP. This guarantees that MLR can satisfy constraint (3).

(Polynomial Running Time): The `while` loop iterates at most n times to select a disk for each AP in Algorithm *ChooseDisks*. Lines 4-8 take $O(mn)$ time to obtain a disk with the minimum local ratio. Lines 9-14 can be completed in $O(m)$ time. Then, line 15 updates the MLR of every $D \in \mathcal{D}$ in $O(mn)$ time. The body of the `for` statement (lines 16-23) takes $O(mn^2)$ time to update the variables of the system state and remove the incapacity disks from \mathcal{D} . In conclusion, MLR runs in polynomial time ($O(mn^3)$). \square

3.2 An Instance for MPCC

To further understand the MLR approach, we cite an instance to illustrate the process of it. In Fig. 2, we present an instance of the MPCC problem (U, A, k) , where $U = \{1, 2, \dots, 20\}$, $A = \{1, 2, 3, 4, 5\}$ and $k = 5$. For ease of presentation, we abstract all facilities as dots distributed in 2-dimensional coordinates. The larger blue dots represent APs, and the smaller red dots represent TDs, as shown in Fig. 2. The disk drawn as a solid line is the final disk result obtained by the MLR method. The specific power value that each AP should provide can be calculated by the equation $p(a) = c \cdot r(a)^\alpha$ (in this case $c = 1$, $\alpha = 2$). The disk drawn as the dotted line is the disk temporarily selected by the MLR method during processing (that is, the value assigned to the variable l_a during the algorithm's execution). There are nine disks in Fig. 2, indicating that the main loop of the MLR method has been executed nine times in total. Next, we will introduce step by step how MLR obtained the results in this instance.

After inputting the instance (U, A, k) and initializing the relevant variables, the algorithm enters the main loop for the first iteration. After calculating the local ratio e of all disks in \mathcal{D} , the disk $D_{3,10}$ with the minimum local ratio is selected. In this iteration, l_3 is temporarily assigned as $D_{3,10}$, and TD10 is allocated to AP3. We remove $D_{3,10}$ from \mathcal{D} and update local ratio of the remaining disks in \mathcal{D} with $e_{D_{3,10}}$. Because TD10 is covered, at the end of the

Table 2. Configuration of experimental parameters

Param	Description	Value
α	Power parameter in equation 1	[1,5]
c	Power parameter in equation 1	1
k	Capacity of AP	[25,100]
r	Side length of the area	[15,100]
$p_{iAP}^X (p_{jTD}^X)$	X coordinate of AP i (TD j)	[0,100]
$p_{iAP}^Y (p_{jTD}^Y)$	Y coordinate of AP i (TD j)	[0,100]

current iteration, the entire system must be updated. In the following iterations, this process is repeated continuously. According to the iteration sequence, the disks selected for each iteration are $D_{3,10}$, $D_{4,19}$, $D_{5,7}$, $D_{3,8}$, $D_{1,4}$, $D_{2,5}$, $D_{4,20}$, $D_{2,16}$ and $D_{1,6}$. After the algorithm's main loop ends, the result set of the algorithm is the union of all l_a , $a \in A$, that is, $\bar{\mathcal{D}} = \{D_{1,6}, D_{2,16}, D_{3,8}, D_{4,20}, D_{5,7}\}$. In addition, the coverage information of all APs is also obtained: $C_1 = \{1, 3, 4, 6\}$, $C_2 = \{2, 5, 15, 16\}$, $C_3 = \{8, 10, 12, 14, 18\}$, $C_4 = \{19, 20\}$ and $C_5 = \{7, 9, 11, 13, 17\}$. These TDs, which are simultaneously contained by some disks in $\bar{\mathcal{D}}$, will be covered by the disk selected first. For example, $D_{3,8}$ and $D_{2,16}$ contain TD8,10 and 12 in Fig. 2. During the algorithm's execution, $D_{3,8}$ is selected before $D_{2,16}$, so $D_{3,8}$ covers that TD8, 10, 12.

4 EXPERIMENTAL RESULTS

According to the MLR algorithm proposed above, we use synthetic data to conduct practical experiments to simulate the power control for APs in the wireless network. The experiments ignore the vertical distribution of two facilities but map their positions to a 2-dimensional coordinate system. The relevant parameters are shown in Table 2. The specific experimental settings are as follows:

- (1) The hardware configuration of the experimental environment is as follows: the CPU is an Intel i7-10700, configured with 8 cores and 16 threads at 2.9 GHz, with 16 GB memory, and a hard disk capacity of 1 TB.
- (2) The entire system includes two facilities, TD and AP, which are distributed randomly in a 2-dimensional space. The capabilities of each AP are limited, and all TDs must be covered by an AP. To ensure that the total system capacity meets the requirements of covering all TDs, we guarantee $m \cdot k \geq n$.
- (3) Each experiment is performed 50 times, and the final results are averaged to reduce the impact of randomness.
- (4) The IP utilization rate of an AP is the ratio of the number of TDs covered by it to its capacity.
- (5) The variance of IP utilization is defined by formula (8)

$$s^2 = \frac{\sum_{a \in A} (|C_a| - n/m)^2}{m}. \quad (8)$$

According to the property of variance, the smaller the value of s^2 is, the more balanced the number of TDs covered by each AP; on the contrary, a few APs cover all TDs.

- (6) In this section, we compare the MLR with the OPT and NCA (nearest capable access) approaches. OPT uses IBM's open source tool CPLEX to obtain the optimal solution of the MPCC problem. NCA is another greedy-based method used to solve the MPCC problem. When deciding which AP will cover which TD in each iteration, it selects the closest AP-TD point pair where the AP still has capacity.

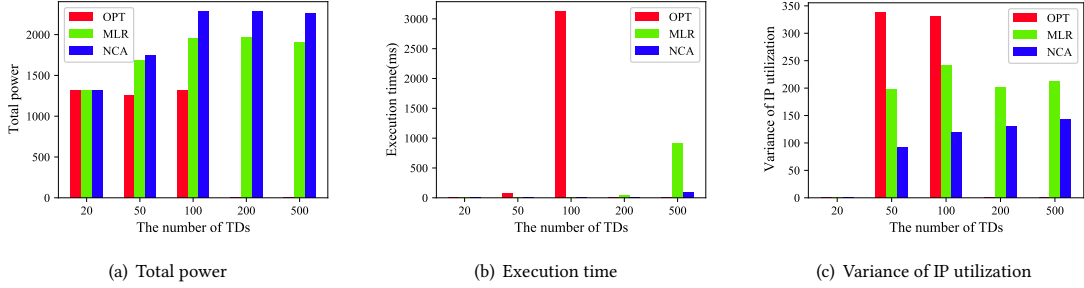


Fig. 3. System performance under different numbers of TDs.

4.1 Impact of the Number of TDs

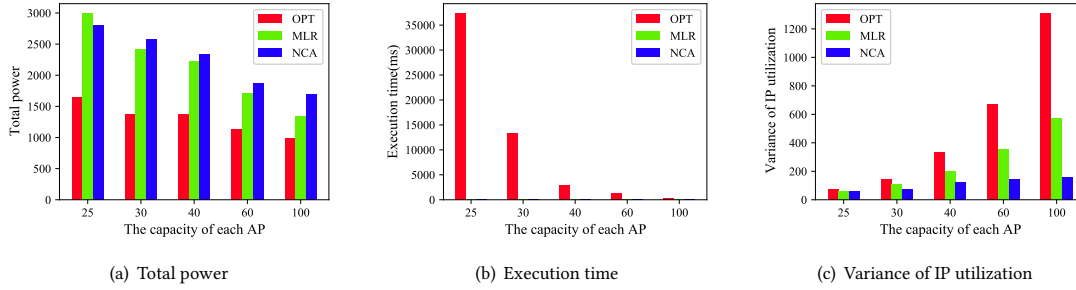
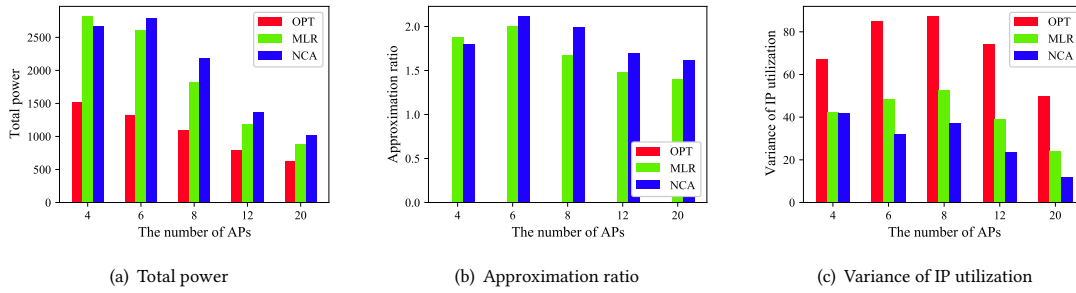
In this experiment, we analyzed the impact of changes in the number of TDs on the MPCC problem. The main foci are the total system power, algorithm execution time, and AP capacity utilization variance. The number of TDs gradually increases from 20 to 500, and all facilities are distributed in an area with a side length of 40. The ratio between the number of TDs and APs is maintained at 25:1, and the capacity of each AP is $k = 40$. For the two constants in equation (1), the values are $c = 1$ and $\alpha = 2$. When the number of TDs is greater than 200, the optimal solution of a single instance cannot be obtained within 10 minutes.

Fig. 3(a) shows the variation in the total power of the three algorithms as the number of TDs increases. Except that the power of the three methods is the same when the number of TDs is 20 (there is only one AP at this time), MLR is significantly better than NCA in all other cases. When the number of TDs changes in the range of 20 to 100, the total power of MLR and NCA increases significantly, while the change in OPT is modest because all facilities are randomly and evenly distributed in a fixed-size area. When the numbers of TDs and APs are small, the result of the optimal solution will not produce a large difference in the total power under different conditions. For MLR and NCA, in some cases, the pursuit of local optimization will produce worse results globally. When the number of TDs increases from 100 to 500, the area remains a fixed size. The total power of MLR and NCA changes only slightly, and the total power of MLR actually decreases. When the number of TDs is small (100), their distribution is sparse, and APs require a larger radius to ensure coverage. When the number of TDs is large (500), the distribution of them and the APs is dense. Although more APs are activated, their coverage radii are reduced. Therefore, under the condition of considering only power (not considering the cost of APs) and a dense distribution of TDs, even if fewer APs can cover all the TDs, more APs have reduced power.

Fig. 3(b) shows that the time required to obtain the optimal solution increases exponentially as the number of terminal devices increases. When the scale is small, both MLR and NCA can produce results in a very short time. Since MLR can still produce better results than NCA in 10 s at the maximum scale, this approach is acceptable in actual application scenarios. As shown in Fig. 3(c), in the results of OPT, AP coverage of TDs is more concentrated, while NCA is the most dispersed, and the result of MLR is somewhere in between.

4.2 Impact of k

In this experiment, we explored the impact of variations in the capacity of APs under the condition that the numbers of APs and TDs remain stable. Where $m = 4$ and $n = 100$, all facilities are randomly and evenly distributed in an area

Fig. 4. System performance under different k .Fig. 5. System performance under different numbers of APs ($k = 25$).

with a side length of 40. The capacity of AP k varies from 25 to 100. Clearly, when $k = 25$, all APs can just meet the requirements of covering all TDs.

Fig. 4 shows the total system power, execution time and the variance of IP utilization as k changes. MLR's performance is not as good as NCA when $k = 25$, as shown in Fig. 4(a). In highly tight capacity conditions, the MLR method will have more extreme situations, such as the last few TDs being covered by APs that are far away. However, as k continues to increase, the total power obtained by MLR decreases faster. In practical applications, the extreme situation of capacity shortage is a small probability event (if it occurs frequently, it means that additional APs need to be added). MLR shows better performance when there is surplus capacity. From Fig. 4(b), we know that the OPT method consumes a lot of time when capacity is tight. In contrast, the execution time of the MLR method and NCA is not related to k . The variance results shown in Fig. 4(c) indicate that the distribution of TDs tends to become more dense as k increases. Thus, when $r = 40$, selecting a disk with a larger radius that covers more TDS can save more power. Section 4.3 discusses the influence of the number of APs on the variance under the condition of constant k .

4.3 Impact of the number of APs with $k = 25$

In the third experiment, we kept the number of TDs n and the capacity of APs k unchanged, with $n = 100$ and $k = 25$. In this case, as m increases, the total capacity of the entire system also increases. m gradually grows from 4 to 20, and the corresponding total system capacity increases from 100 to 500.

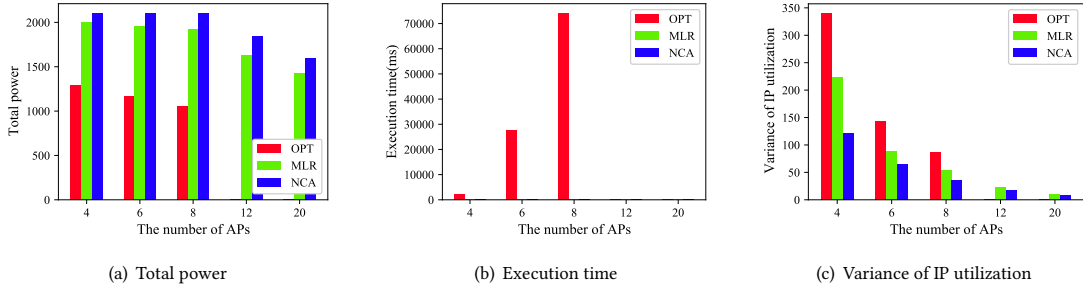


Fig. 6. System performance under different numbers of APs ($K = 160$).

When $m = 4$, the system capacity is just sufficient to cover all TDs. In this case, as shown in Fig. 5(a), the results obtained by MLR and NCP are worse than the optimal solution. Notably, the MLR method is not better than NCP because when capacity is incredibly tight, if some TDs are densely distributed around an AP, the MLR method will fill it more quickly and fall into a local optimum. For example, in Fig. 2, after AP3 covers TDs $\{8, 10, 12, 14, 18\}$, AP2 has to provide more power to cover the TDs $\{15, 16\}$ far away from it. As the number of APs and the total capacity of the system increase, the total power obtained by the three methods decreases. Fig. 5(b) shows that the performance improvement of the MLR method is the most obvious. Fig. 5(c) shows the variance changes of each method. The most obvious is that the coverage schemes of the OPT method are very intensive in all cases. It reaches the maximum value when $m=8$ (when $m=4$, $n=100$, the variance of OPT is still different from that of the other two methods because the total capacity K of the system may be greater than 100).

4.4 Impact of the number of APs with $K = 160$

In this experiment, we kept the number of TDs n and the total system capacity K constant at 100 and 160, respectively. The number of APs m gradually increases from 4 to 20, and k changes accordingly. All facilities are randomly and evenly distributed in an area with a side length of 40. When $m > 12$, OPT cannot produce results within 10 minutes.

As shown in Fig. 6(a), the results of MLR are better than those of NCA. As the number of APs increases, the distribution of APs in a fixed-size area becomes denser. Overall, the average radius of APs will be reduced, and the impact of the decreasing radius on the system is greater than that of the number of APs (more disks). As the number of APs increases and the capacity decreases, the variance of IP utilization among APs becomes more similar, as shown in Fig. 6(c). In Fig. 6(b), the variation in the number of APs also has a considerable impact on the execution time of the OPT method. On the contrary, the other two have almost no fluctuations.

5 CONCLUSION

Signal coverage consumes considerable energy in wireless networks, so this paper studies how to assign an appropriate power for APs to reduce energy consumption. We consider the signal coverage process in edge networks as a minimum power coverage problem and define this problem as an MPCC problem, which is a special case of MPC problem. Capacity constraints make it more difficult to solve. Modeling this problem in MEC is also a novel exploration. Then, we propose a local-ratio-based approach to solve the MPCC problem. Finally, various experiments show that our method can produce satisfactory results in most cases.

For the MPCC problem, we can also learn from the previous methods of the MPC problem to solve it (such as the primal-dual method). The MPCC problem is more complex but more valuable when considering heterogeneous APs and TDs' natural number requirements (not just 1). This is worthy of further study. In addition, this paper notes in Experiment 4.3 that more APs can decrease the total energy of the system. This does not take into account the cost of the AP itself, which can be considered in future studies.

6 ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China [Nos. 12071417, 61762091, 62062065], in part by the 13th Postgraduate Innovation Project of Yunnan University [No. 2021Z079] and Scientific research foundation of Yunnan Provincial Department of Education [No. 2022J0002].

REFERENCES

- [1] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. 2018. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal* 5 (2 2018), 450–465. Issue 1. <https://doi.org/10.1109/JIOT.2017.2750180>
- [2] Helmut Alt, Esther M. Arkin, Hervé Brönnimann, Jeff Erickson, Sándor P. Fekete, Christian Knauer, Jonathan Lenchner, Joseph S. B. Mitchell, and Kim Whittlesey. 2006. Minimum-cost coverage of point sets by disks. In *Proceedings of the twenty-second annual symposium on Computational geometry - SCG '06*, Vol. 2006. ACM Press, New York, New York, USA, 449. <https://doi.org/10.1145/1137856.1137922>
- [3] Vittorio Bilò, Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. 2005. Geometric Clustering to Minimize the Sum of Cluster Sizes. In *Lecture Notes in Computer Science*. Vol. 3669. 460–471. https://doi.org/10.1007/11561071_42
- [4] Timothy M. Chan, Elyot Grant, Jochen Könemann, and Malcolm Sharpe. 2012. Weighted Capacitated, Priority, and Geometric Set Cover via Improved Quasi-Uniform Sampling. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1576–1585. <https://doi.org/10.1137/1.9781611973099.125>
- [5] Wang Chi Cheung, Michel X. Goemans, and Sam Chiu-Wai Wong. 2014. Improved Algorithms for Vertex Cover with Hard Capacities on Multigraphs and Hypergraphs. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, Vol. 0. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1714–1726. <https://doi.org/10.1137/1.9781611973402.124> arXiv:1606.07861
- [6] Trinh Van Chien, Thuong Nguyen Canh, Emil Bjornson, and Erik G. Larsson. 2020. Power Control in Cellular Massive MIMO With Varying User Activity: A Deep Learning Solution. *IEEE Transactions on Wireless Communications* 19 (9 2020), 5732–5748. Issue 9. <https://doi.org/10.1109/TWC.2020.2996368>
- [7] Michele Chincoli and Antonio Liotta. 2018. Self-Learning Power Control in Wireless Sensor Networks. *Sensors* 18 (1 2018), 375. Issue 2. <https://doi.org/10.3390/s18020375>
- [8] Julia Chuzhoy and Joseph (Seffi) Naor. 2006. Covering Problems with Hard Capacities. *SIAM J. Comput.* 36, 2 (jan 2006), 498–515. <https://doi.org/10.1137/S0097539703422479>
- [9] Cormen, Thomas H. and Leiserson, Charles E. and Rivest, Ronald L. and Stein, Clifford. 1990. *Introduction to Algorithms*. MIT Press.
- [10] Yanpeng Dai, Junyu Liu, Min Sheng, Nan Cheng, and Xuemin Shen. 2021. Joint Optimization of BS Clustering and Power Control for NOMA-Enabled CoMP Transmission in Dense Cellular Networks. *IEEE Transactions on Vehicular Technology* 70 (2 2021), 1924–1937. Issue 2. <https://doi.org/10.1109/TVT.2021.3055769>
- [11] Sudipto Guha, Refael Hassin, Samir Khuller, and Einat Or. 2003. Capacitated vertex covering. *Journal of Algorithms* 48, 1 (aug 2003), 257–270. [https://doi.org/10.1016/S0196-6774\(03\)00053-1](https://doi.org/10.1016/S0196-6774(03)00053-1)
- [12] Mong-Jen Kao. 2021. Iterative Partial Rounding for Vertex Cover with Hard Capacities. *Algorithmica* 83, 1 (jan 2021), 45–71. <https://doi.org/10.1007/s00453-020-00749-9> arXiv:1606.08667
- [13] Weidong Li, Xi Liu, Xiaobo Cai, and Xuejie Zhang. 2019. Approximation algorithm for the energy-aware profit maximizing problem in heterogeneous computing systems. *J. Parallel and Distrib. Comput.* 124 (2019), 70–77. <https://doi.org/10.1016/j.jpdc.2018.10.013>
- [14] Xiaofei Liu, Weidong Li, and Han Dai. 2022. Approximation algorithms for the minimum power cover problem with submodular/linear penalties. *Theoretical Computer Science* (5 2022). <https://doi.org/10.1016/j.tcs.2022.05.012>
- [15] Xiaofei Liu, Weidong Li, and Runtao Xie. 2021. A primal-dual approximation algorithm for the k-prize-collecting minimum power cover problem. *Optimization Letters* (11 2021). <https://doi.org/10.1007/s11590-021-01831-z>
- [16] Xi Liu, Weidong Li, and Xuejie Zhang. 2018. Strategy-Proof Mechanism for Provisioning and Allocation Virtual Machines in Heterogeneous Clouds. *IEEE Transactions on Parallel and Distributed Systems* 29, 7 (2018), 1650–1663. <https://doi.org/10.1109/TPDS.2017.2785815>
- [17] Nabil H. Mustafa and Saurabh Ray. 2010. Improved Results on Geometric Hitting Set Problems. *Discrete and Computational Geometry* 44, 4 (dec 2010), 883–895. <https://doi.org/10.1007/s00454-010-9285-9>

- [18] Daniel Ronnow and Peter Handel. 2019. Nonlinear Distortion Noise and Linear Attenuation in MIMO Systems—Theory and Application to Multiband Transmitters. *IEEE Transactions on Signal Processing* 67 (10 2019), 5203–5212. Issue 20. <https://doi.org/10.1109/TSP.2019.2935896>
- [19] Kasturi Varadarajan. 2010. Weighted Geometric Set Cover via Quasi-Uniform Sampling. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (Cambridge, Massachusetts, USA) (*STOC '10*). Association for Computing Machinery, New York, NY, USA, 641–648. <https://doi.org/10.1145/1806689.1806777>
- [20] Fan Wu, Supeng Leng, Sabita Maharjan, Xiaoyan Huang, and Yan Zhang. 2021. Joint Power Control and Computation Offloading for Energy-efficient Mobile Edge Networks. *IEEE Transactions on Wireless Communications* (2021), 1–1. <https://doi.org/10.1109/TWC.2021.3130649>
- [21] Jixian Zhang, Laixin Chi, Ning Xie, Xutao Yang, Xuejie Zhang, and Weidong Li. 2022. Strategy-Proof Mechanism for Online Resource Allocation in Cloud and Edge Collaboration. *Computing* 104, 2 (feb 2022), 383–412. <https://doi.org/10.1007/s00607-021-00962-6>
- [22] Yan Zhang. 2022. *Mobile Edge Computing*. Vol. 9. Springer International Publishing. <https://doi.org/10.1007/978-3-030-83944-4>