

Article

A Primal–Dual-Based Power Control Approach for Capacitated Edge Servers

Qinghui Zhang ¹, Weidong Li ², Qian Su ¹ and Xuejie Zhang ^{1,*}¹ School of Information Science and Engineering, Yunnan University, Kunming 650091, China² School of Mathematics and Statistics, Yunnan University, Kunming 650091, China

* Correspondence: xjzhang@ynu.edu.cn

Abstract: The intensity of radio waves decays rapidly with increasing propagation distance, and an edge server’s antenna needs more power to form a larger signal coverage area. Therefore, the power of the edge server should be controlled to reduce energy consumption. In addition, edge servers with capacitated resources provide services for only a limited number of users to ensure the quality of service (QoS). We set the signal transmission power for the antenna of each edge server and formed a signal disk, ensuring that all users were covered by the edge server signal and minimizing the total power of the system. This scenario is a typical geometric set covering problem, and even simple cases without capacity limits are NP-hard problems. In this paper, we propose a primal–dual-based algorithm and obtain an m -approximation result. We compare our algorithm with two other algorithms through simulation experiments. The results show that our algorithm obtains a result close to the optimal value in polynomial time.

Keywords: power control; primal–dual; approximation algorithm; edge computing; power cover



Citation: Zhang, Q.; Li, W.; Su, Q.; Zhang, X. A Primal–Dual-Based Power Control Approach for Capacitated Edge Servers. *Sensors* **2022**, *22*, 7582. <https://doi.org/10.3390/s22197582>

Academic Editor: Christian Vollaire

Received: 17 August 2022

Accepted: 1 October 2022

Published: 6 October 2022

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



Copyright: © 2022 by the authors. 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

1.1. Background

In recent years, edge computing has been proposed as a timely and resource-efficient alternative to address data computation issues [1]. Edge computing brings the service and utilities of cloud computing closer to the user. The response time that users perceive is effectively reduced, and the data processing in the cloud center is alleviated. In addition to reduced service delays, we must consider another issue. From 2010 to 2020, global Internet traffic expanded 15-fold, and the energy consumed in transmitting data has increased at the same rate [2]. Industry and academia have focused on reducing energy consumption in wireless communication processes, in which the power of the antenna is an important consideration. Various different power control methods have been proposed for wireless networks. In terms of controlling the power of cellular networks, [3] interpreted cellular and cell-free massive MIMO networks as max–min utility optimization problems with affine interference mappings and polyhedral constraints. In [4], Dai et al. investigated the joint optimization of base station (BS) clustering and power control for non-orthogonal multiple access (NOMA)-enabled coordinated multipoint (CoMP) transmission in dense cellular networks, maximizing the sum rate of the system. In addition, in terms of wireless sensor networks (WSNs), Ref. [5] investigated how machine learning could be used to reduce the possible transmission power level of wireless nodes and, in turn, satisfy the quality requirements of the overall network. Reducing the transmission power has benefits in terms of both energy consumption and interference. In [6], Moltafet et al. developed a dynamic control algorithm using the Lyapunov drift-plus-penalty method. They used this approach to jointly optimize the sampling action of each sensor, the transmit power allocation, and the subchannel assignment, minimizing the average total transmit power of all sensors, subject to a maximum average Age of Information (AoI) constraint for each sensor.

The above researchers optimized the antenna power by considering the influence of each user on the system and ensuring the user's quality of service (QoS). QoS may include the data transmission rate, response time, task execution time, etc. However, in edge networks, some computing resources, such as CPUs, are often scarcer than channel resources. Edge networks often have no remaining CPU resources before they run out of channel resources. Therefore, wireless channel allocation and signal interference are not considered in this paper. In addition, the directional signal of an antenna is very commonly used in today's cellular communication. In this paper, we abstract the MIMO-based base station as an omni-directional antenna. So, the signal coverage of the antenna can be seen as a disk.

In such edge networks based on the above assumptions, the signal coverage area of an antenna, which is determined by the antenna's power, is a disk centered on the it. Because the signal intensity decreases with increasing distance, a larger signal coverage disk needs more power [7]. In an edge computing system composed of edge servers and users, the edge servers communicate with users through wireless signals transmitted by antennas. This article assumes that the edge server has an omni-directional antenna and is abbreviated as the *server*. When a server serves a user, its signal should cover the user. In addition, each edge server can provide only limited resources for the users it serves. In summary, the problem studied in this article is how to appropriately assign power in a capacitated edge server to ensure that the server covers all users while minimizing the total power. We define this problem as the *capacitated minimum power cover (CMPC)* problem. The signal-to-noise ratio (SNR) between the server and the user is usually the factor affecting the quality of communication. We know that SNR is inversely proportional to the signal distance. We make SNR_{min} to be the SNR between a server i and j user, where the distance between them is the farthest in the system. We assume that as long as the SNR between any server and user is greater than SNR_{min} , the server could serve this user. In addition, the overall system has the capacity to cover all users. As a result, we can obtain an approximate scheme that covers all users with the technique developed in this paper, and we explain that the extreme case, where a server serves the furthest user, rarely occurs. We can also address the situation in which the demand exceeds the capacity by repeatedly using the method proposed in this paper. Therefore, the above two assumptions are reasonable.

Our detailed contributions in this work can be summarized as follows:

- (*CMPC and Resource Allocation Model*) The CMPC problem is a fundamental *minimum power coverage (MPC)* problem. The MPC problem is NP-hard even in the absence of capacity constraints [8,9]. Based on the CMPC problem, we establish a minimum power control resource allocation model in a capacitated edge network.
- (*Primal–Dual Algorithms*) To address the above challenges, we propose a primal–dual-based approximation algorithm to solve the CMPC problem. After the theory proof, we obtain an approximation guarantee of m (m edge servers considered) in polynomial time.
- (*Performance Evaluation*) Numerical results are presented to validate the effectiveness and efficiency of our proposed algorithms.

1.2. Previous Research

The primal–dual-based approximation algorithm is a generalization of the primal–dual method used for linear programming and combinatorial optimization problems [10]. This algorithm provides effective solutions for many optimization problems in mobile edge computing (MEC). In [11], attack-resilient distributed algorithms based on primal–dual optimization were proposed for situations when Byzantine attackers are present in a system. In [12], Wang et al. modeled the system in an online manner and formulated the underlying optimization problem, maximizing the total profit according to constraints on the computational resources on the edge clouds and job migration. Furthermore, a $(1 - 1/e)$ -competitive primal–dual-based online algorithm was proposed. An efficient so-

cial welfare approximation algorithm that used a classic primal–dual framework was developed in [13]. In this paper, the cloud market for computing jobs with completion deadlines was studied, and efficient online auctions for cloud resource provisioning were designed.

Broadly, the CMPC problem belongs to the family of *minimum weight set cover* (MWSC) problems. In general, the MWSC problem is challenging to solve optimally, even for some simple versions. For example, Alt et al. and Bilò et al. presented a minimum cost-covering problem without capacity constraints that is still NP-hard for any $\alpha > 1$ in [8,9]. Thus, the development of polynomial-time approximation algorithms is the main objective for CMPC problems. In [14], Zhang et al. proposed a local-ratio-based power control approach for the access point in mobile edge computing. As for the theoretical study of the minimum power cover (MPC) problem, Liu et al. introduced the k -prize-collecting minimum power cover problem (k -PCPC) where k is the number of users that need be covered in [15]. They presented a novel two-phase primal–dual algorithm for the k -PCPC with an approximation ratio of at most 3^α . In recent work [16], Liu et al. considered an MPC problem with submodular and linear penalties. For the minimum power partial cover problem, Dai et al. present an $O(\alpha)$ -approximation algorithm in [17].

As for the disk cover problem, some researchers focused on minimizing the cardinality of disks. In [18], Lyu et al. considered UAV-Mounted mobile base stations (UAV-MBS) with the same radii to provide wireless coverage for a group of distributed ground terminals and minimize the number of UAV-MBS. In [19], a rounding-based mechanism for capacitated covering problems that minimized the cardinality was proposed and obtained a constant approximation to address this problem. In the past, there have been some researchers who investigated the minimum weight disk cover problem. Varadarajan considered the weight [20] and presented a clever quasi-uniform sampling technique that was improved by Chan et al. [21], yielding a constant approximation for the minimum weight disk cover problem. This constant approximation was generalized by Bansal and Pruhs [22] for the minimum weight disk multicover problem in which every point must be covered multiple times.

Energy-efficient optimization has attracted extensive attention in mobile computing. By endowing edge servers with multiple power states, e.g., active state and sleep state, it is promising to improve the total energy consumption of edge servers through switching under-utilized servers into sleep state in [23]. Ali et al. in [24] proposed a novel energy-efficient deep learning-based offloading scheme (EEDOS) to select an optimal set of computation components to offload to ESs, aiming at minimizing the energy consumption of MDs. Li et al. in [25] studied the channel selection for task offloading. The effect of multi-channel interference on the energy efficiency of task offloading was taken into account. Gu et al. [26] studied the problem of how to efficiently assign computing tasks to reduce energy consumption in the edge computing system under the constraints of the computing capacity of both MDs and ENs, wireless channel conditions, and delay.

2. System Model and Preliminaries

System Model

We consider that all the facilities in the edge networks are distributed in a fixed dimensional \mathbb{R}^d space, where the facilities are edge servers and users. Let S denote the set of m edge servers and U denote the set of n users. For each user $j \in U$ to connect to an edge server $i \in S$, j must be contained in the signal disk formed by i and obtain an IP from this disk. The IP capacity of edge server i is k_i . If i forms a signal disk with radii r_i , the power we should provide is

$$p_i = c \cdot (r_i)^\alpha, \quad (1)$$

where c and α are constants (α is usually called the *attenuation factor*).

Although r_i can be selected arbitrarily, there will be precisely one terminal device located on the boundary of the disk with radius $d(i, j)$ in the optimal solution, where $d(i, j)$ denotes the distance between i and j . Therefore, at most, mn disks need to be considered. We denote the set of such disks as \mathcal{D} . i and j can form a disk $D_{ij} \in \mathcal{D}$ with radius $r_{ij} = d(i, j)$

and center i . The disk set with i as the center in \mathcal{D} is represented by \mathcal{D}_i . Therefore, we have $\mathcal{D}_i = n$. A set of disks $\widehat{\mathcal{D}} \in \mathcal{D}$ is called a cover for a set of users $\widehat{U} \in U$ if each user $j \in \widehat{U}$ is covered by some disk in $\widehat{\mathcal{D}}$. The problem is to find a cover $\widehat{\mathcal{D}} \in \mathcal{D}$ for S with minimum power $p(\widehat{\mathcal{D}}) = \sum_{D_{ij} \in \widehat{\mathcal{D}}} p_{ij}$.

In the following, we denote the instance of the CMPC problem as (U, \mathcal{D}, k) and the optimal power for the instance (U, \mathcal{D}, k) as OPT . To simplify the notation, we use D_{ij} to represent both a disk in \mathcal{D} and the set of users contained in D_{ij} and p_{ij} to denote the power of disk D_{ij} , where $p_{ij} = c \cdot (r_{ij})^\alpha$. We note that D_{ij} containing a user u means that u is within the range of D_{ij} ($u \in D_{ij}$); thus, D_{ij} covering u means that server i serves u , who then obtains resources. In addition, we use SNR_{min} to denote the SNR between a server–user pair (s, u) where $(s, u) = \arg \max d(i, j)$. Therefore, the SNR between any other server–user pair will not be less than SNR_{min} and a server could serve any users. Obviously, the power of s occurred by serving $i \in U \setminus \{u\}$ would not more than the power by serving u . We give the mentioned notations of this paper in Table 1.

In actual scenarios, there are situations in which multiple users are located at the same distance from a server. Suppose that a server cannot provide services for all users simultaneously. In that case, the server provides the users with services in an orderly manner according to the urgency or bid level of the users. Because this article does not focus on task scheduling, we use Definition 1 to determine the order of the users.

Definition 1. Map the positions of all users and servers to a coordinate system and use $\cos(\vec{s}\vec{u})$ to represent the cosine of the angle between vector $\vec{s}\vec{u}$ and the x-axis formed by server s and user u . The distances between users u and d and server s are r_{su} and r_{sd} , respectively. If $\cos(\vec{s}\vec{u}) > \cos(\vec{s}\vec{d})$, then $r_{su} > r_{sd}$, and the same is true if $p_{su} > p_{sd}$. Because $r_{su} = r_{sd}$, D_{su} contains d but D_{sd} does not contain u .

Table 1. Summary of Notations.

Notations	Description
m	# of servers
n	# of users
S	Set of servers
U	Set of users
U'	Set of uncovered users
SNR_{min}	SNR between a server and user with the farthest distance
\mathcal{D}'	Set of unselected disks
k_i	Server i 's capacity
$d(i, j)$	Distance between server i and user j
D_{ij}	Disk formed by i and j or the set of users the disk contains
r_i (r_{ij})	Radius of server i 's disk (disk D_{ij})
p_i (p_{ij})	Power of server i 's disk (disk D_{ij})
\mathcal{D}	Set of disks formed by m servers and n users
\mathcal{D}_i	Set of disks centered on server i in \mathcal{D}
x_{ij}	Disk D_{ij} is selected (1) or not (0)
$y_{h,ij}$	User h is covered by disk D_{ij} (1) or not (0)
θ_h	Costs charged due to user h
μ_i	Extra cost of server i selecting multiple disks
β_{ij}	Minimum cost that disk D_{ij} charges each user it contains
$\gamma_{h,ij}$	Cost that user h is willing to pay for disk D_{ij}
\mathcal{D}'_{ij}	Set of uncovered users in D_{ij}
$\delta(D_{ij})$	Set of users charged by D_{ij}

3. A Primal–Dual Approach for Capacitated Servers

In this section, we present a primal–dual algorithm for the CMPC problem on the instance (U, \mathcal{D}, k) . Then, we show how to use this algorithm to determine the power assignment of each server.

3.1. Capacitated Minimum Power Cover Problem

Under the assumption of sufficient capacity ($\sum_{i \in S} k_i \geq n$), the CMPC problem can be formulated as an integer program. Variable x_{ij} indicates whether disk $D_{ij} \in \mathcal{D}$ is chosen; that is, $x_{ij} = 1$ if and only if D_{ij} is selected. Variable $y_{h,ij}$ indicates whether user $h \in U$ is covered by D_{ij} ; here, $y_{h,ij} = 1$ if and only if h is covered by D_{ij} . The integral linear programming (ILP) problem can be formulated as follows:

$$\min \sum_{D_{ij} \in \mathcal{D}} p_{ij} \cdot x_{ij} \quad (2)$$

$$\text{s.t.} \quad \sum_{D_{ij}: h \in D_{ij}} y_{h,ij} \geq 1, h \in U \text{ and } D_{ij} \in \mathcal{D} \quad (2a)$$

$$k_i x_{ij} - \sum_{h: h \in D_{ij}} y_{h,ij} \geq 0, D_{ij} \in \mathcal{D} \quad (2b)$$

$$x_{ij} \geq y_{h,ij}, h \in D_{ij} \in \mathcal{D} \quad (2c)$$

$$\sum_{D_{ij} \in \mathcal{D}_i} x_{ij} \leq 1, \mathcal{D}_i \in \mathcal{D} \quad (2d)$$

$$x_{ij} \in \{0, 1\}, y_{h,ij} \in \{0, 1\}, \forall h \in U, \forall D_{ij} \in \mathcal{D} \quad (2e)$$

Note that constraint (2a) ensures that user h is covered by at least one disk. The capacity limit of each disk is expressed in constraint (2b). Constraint (2c) guarantees that disk D_{ij} can not cover user h until the disk is selected. Constraint (2d) implies that each server $i \in S$ can select only a disk as its power assignment.

The ILP without constraints (2b) and (2d) is still an NP-hard combinatorial optimization problem that is equivalent to the classic set cover problem. The challenge escalates when we consider the capacity of the servers and the uniqueness of the power assignments. To address these challenges, we utilize the primal–dual algorithm design technique. We relax the ILP constraints of x_{ij} and $y_{h,ij}$ to $x_{ij} \geq 0$ and $y_{h,ij} \geq 0$ to formulate the dual problem. Note that we do not need to add the constraints $x_{ij} \leq 1$ and $y_{h,ij} \leq 1$, since they are automatically satisfied in an optimal solution of Equation (2). By introducing dual variables $\theta_h, \beta_{ij}, \gamma_{h,ij}$ and μ_i to constraints (2a)–(2d), respectively, the dual LP of the relaxed (2) becomes:

$$\max \sum_{h \in C} \theta_h - \sum_{i \in S} \mu_i, \quad (3)$$

$$\text{s.t.} \theta_h - \beta_{ij} - \gamma_{h,ij} \leq 0, h \in D_{ij} \in \mathcal{D}, \quad (3a)$$

$$k_i \beta_{ij} + \sum_{h: h \in D_{ij}} \gamma_{h,ij} \leq p_{ij} + \mu_i, D_{ij} \in \mathcal{D}_i \in \mathcal{D}, \quad (3b)$$

$$\theta_h \geq 0, \beta_{ij} \geq 0, h \in U, D_{ij} \in \mathcal{D}, \quad (3c)$$

$$\gamma_{h,ij} \geq 0, h \in D_{ij} \in \mathcal{D}. \quad (3d)$$

These dual variables also have economic benefits. Servers charge a fee of θ_h to user $h \in U$ to provide services. The dual variable μ_i represents the additional cost when server $i \in S$ selects multiple disks. Users are not charged more than they are willing to pay. For a user h contained in D_{ij} , h is willing to pay β_{ij} when D 's capacity is insufficient. Otherwise, the fee paid by h is $\gamma_{h,ij}$. Furthermore, all users contained in disk D_{ij} are willing to pay no more than the sum of p_{ij} and μ_i . Therefore, the objective function (3) maximizes the profit of the servers.

We next design an efficient primal–dual covering scheme that simultaneously increases the dual variables by a polynomial number of times, which we use to solve optimization problems (2) and (3).

Algorithm 1 PD.

Input: A set of users U , a set of servers S , a power function $p : D_{ij} \mapsto \mathbb{R}^+$, and a capacity constraint k .

Output: A subset of disks $\widehat{\mathcal{D}}$ covering all users in U .

- 1: Initialize $k'_i \leftarrow k_i, L_i \leftarrow \emptyset, \mu_i \leftarrow 0, \theta_h \leftarrow 0, \beta_{ij} \leftarrow 0, \gamma_{h,ij} = 0, \forall j, h \in U, \forall i \in S$.
 - 2: $\widehat{\mathcal{D}} \leftarrow \emptyset, \mathcal{D}' \leftarrow \mathcal{D}, U' \leftarrow U$.
 - 3: **while** $U' \neq \emptyset$ **do**
 - 4: Increase $\{\theta_h\}_{h \in U'}$ and $\{\beta_{ij} + \gamma_{h,ij}\}_{D'_{ij}:h \in D'_{ij}}$ simultaneously until some disk D_{su} becomes tight. (If $\left| \{h : h \in D'_{ij} \cap U'\} \right| > k_i$, we increase β_{ij} ; otherwise, we increase $\gamma_{h,ij}, h \in D'_{ij}$)
 - 5: $L_s \leftarrow D_{su}$.
 - 6: $U' \leftarrow U' \setminus D'_{su}, \mathcal{D}' \leftarrow \mathcal{D}' \setminus \mathcal{D}'_{<r_{su}}$.
 - 7: $D'_{ij} \leftarrow D'_{ij} \setminus \{D'_{ij} \cap D'_{su}\}, D_{ij} \in \mathcal{D}' \setminus \mathcal{D}'_s$.
 - 8: **end while**
 - 9: $\widehat{\mathcal{D}} \leftarrow \{L\}$.
 - 10: **return** $\widehat{\mathcal{D}}$.
-

3.2. Primal–Dual Algorithm Design

The first use of the primal–dual-based approximation algorithm is based on the work of Bar-Yehuda and Even [27]. The procedure *PD* follows the classic primal–dual method: starting from the trivial dual feasible solution of zero, the method increases the dual variables simultaneously until some disk becomes tight. Then, a tight disk is chosen and iterated until a feasible solution is obtained. Next, we introduce how the algorithm works in detail. The pseudo-code of followed process is shown in Algorithm 1 named PD (Primal–Dual).

Initially, the dual variables $\{\theta\}$ and $\{\alpha\}$ are 0, resulting in a dual feasible solution (with all $\beta_{ij} = 0$ and $\gamma_{h,ij} = 0$). We use U' to denote the set of uncovered users and \mathcal{D}' to denote the set of unselected disks. We use D'_{ij} to denote the uncovered users currently in D_{ij} . The disks are selected by increasing the dual variables θ_h for the uncovered users $h \in U'$ simultaneously. The dual program has two kinds of constraints: user constraints and server constraints.

To maintain the dual feasibility of the user constraints (3a), as we increase θ_h , we must increase β_{ij} or $\gamma_{h,ij}$, where $h \in D_{ij} \in \mathcal{D}'$. If the disk contains a large number of uncovered users, we increase β_{ij} ; otherwise, we increase $\gamma_{h,ij}$. Formally, if $k_i < \left| D'_{ij} \right|$, we increase β_{ij} ; otherwise, we increase $\gamma_{h,ij}$.

For each disk constraint, $k_i \beta_{ij} + \sum_{h:h \in D_{ij}} \gamma_{h,ij} \leq p_{ij} + \mu_i$; initially, the left-hand side of this equation is 0, and the right-hand side is equal to the cost of the disk. This algorithm ensures that each server selects at most one disk; thus, $\mu_i = 0$ for all $i \in S$. When the dual variables of the unassigned edges are increased, we stop the procedure as soon as a disk constraint is met with equality. (In Algorithm 1, this is represented by disk D_{su} in the main loop.) We can confirm only that the users in D'_{su} are served by s , not that s chooses disk D_{su} . We temporarily select this disk as the current disk for s and record the value with $L_s = D_{su}$. Through L_s , we can ensure that each edge server selects at most one disk as its power strategy, so $\mu_i = 0, i \in S$ in the whole algorithm process. In lines 5–6 of Algorithm 1, we update the related variables and sets. First, we delete the users in D'_{su} from U' and the concentric disk with a radius less than $d(s, u)$ from \mathcal{D}' ($\mathcal{D}'_{<r_{su}} = \{D_{ij} : D_{ij} \in \mathcal{D}' \text{ and } r_{ij} < r_{su}\} \cup D_{su}$). Then, we update $\{\mathcal{D}'\}$ in the disks of the other servers. After these variables and sets are updated, the dual variables corresponding

to the removed users and disks stop increasing in subsequent iterations. The above steps are iterated until all users are covered by a disk. The disk that covers the user are the last disk selected by each server, that is, $\{L\}$.

The above process maintains dual feasibility. According to Lemma 1, the capacity constraint is maintained throughout the algorithm.

3.3. A PD Instance

To further understand the PD algorithm, we illustrate an instance of the PD algorithm. In Figure 1, we present an instance of the CMPC problem (U, \mathcal{D}, k) , where $U = \{1, 2, \dots, 20\}$, $S = \{1, 2, 3, 4, 5\}$ and $k = 5$. For ease of representation, we use triangles and circles to represent servers and users, respectively, which are distributed in a two-dimensional coordinate system. In Figure 1, the disk drawn as a solid line is the final disk obtained by Algorithm 1. The specific power value that each server should provide can be calculated with the equation $p(a) = c \cdot r(a)^\alpha$ (in this case, $c = 1$ and $\alpha = 2$). The disk drawn with the dot line is the disk that is temporarily selected by the PD algorithm during processing (that is, the value assigned to the variable l_a during the algorithm's execution). There are nine disks in Figure 1, indicating that the main loop of the PD algorithm produces nine tight disks in total. Next, we introduce how the PD algorithm obtained the results in this instance.

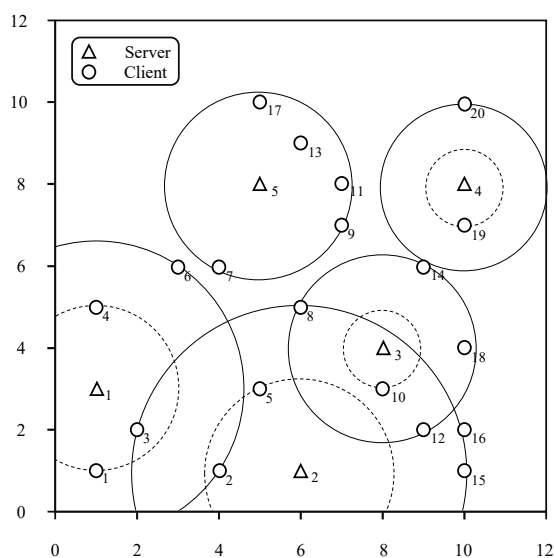


Figure 1. Total power of the PD algorithm with different numbers of servers and values of k .

After the instance (U, \mathcal{D}, k) is input and all dual variables are set to zero, the algorithm enters the while loop. By continuously increasing the dual variables, disks $D_{3,10}$ become tight first. 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 $D_{1,6}$. In this process, each user determines which server serves it. For example, although users u_8 and u_{12} are contained in disk $D_{3,8}$ and disk $D_{2,5}$, respectively, the first selected disk, that is, $D_{3,8}$, covers them. Finally, the users served by servers s_1 to s_5 are $\{u_1, u_4, u_6\}_{s_1}, \{u_2, u_3, u_5, u_{15}, u_{16}\}_{s_2}, \{u_8, u_{10}, u_{12}, u_{14}, u_{18}\}_{s_3}, \{u_{19}, u_{20}\}_{s_4}$, and $\{u_7, u_9, u_{11}, u_{13}, u_{17}\}_{s_5}$.

3.4. Theoretical Analysis

Lemma 1. D_{su} must have the capacity to cover all users in D'_{su} when it is selected. Formally, $|D'_{su}| \leq k_s$ when D_{su} is selected.

Proof. Assume that the D_{su} selected in an iteration satisfies $|D'_{su}| > k_s$. At this point, $k_s \beta_{su} = p_{su}$, with $\gamma_{h,su} = 0, h \in D_{su}$. Then, there must be an unselected disk $D_{sd}, r_{sd} < r_{su}$ with $|D'_{sd}| = k_s$. We have that

$$\begin{aligned} p_{sd} &\geq k_s \beta_{sd} + \sum_{h:h \in D_{sd}} \gamma_{h,sd} \\ &= \sum_{h:h \in D'_{sd}} (\beta_{sd} + \gamma_{h,sd}) \end{aligned} \quad (4)$$

$$\begin{aligned} &= \sum_{h:h \in D'_{sd}} \theta_h \\ &= k_s \beta_{su} \end{aligned} \quad (5)$$

$$\begin{aligned} &= p_{su} \\ &> p_{sd}. \end{aligned} \quad (6)$$

Equation (4) holds because $\gamma_{h,sd} = 0, h \in D_{sd} \setminus D'_{sd}$. Based on the former assumption and Definition 1, $D'_{sd} \subseteq D'_{su}$. Then, $\theta_h = \beta_{su} = (\beta_{sd} + \gamma_{h,sd}), h \in D'_{sd}$. Therefore, Equation (5) holds. In summary, Inequality (6) breaks Constraint (3b), and the above assumption is not tenable. \square

Lemma 2. We can charge the power of each selected disk D_{ij} to users in D'_{ij} such that each user h obtains a charge of at most $m \cdot \theta_h$.

Proof. Define a disk to be a *low-degree* disk if $|D_{ij}| \leq k_i$ when it is selected; otherwise, define the disk as a *high-degree* disk. We discuss the charging mechanism for both low-degree and high-degree disks. We use $\delta(D_{ij})$ to denote the set of users charged by i .

Consider a low-degree disk D_{ij} with $\beta_{ij} = 0$. When D_{ij} is selected, we charge all the users contained in this disk; thus, $\delta(D_{ij}) = D_{ij}$. Since the disk constraint is tight, we have $p_{ij} = \sum_{h:h \in D_{ij}} \gamma_{h,ij} = \sum_{h:h \in \delta(D_{ij})} \theta_h$. Thus, we charge the cost of disk D_{ij} to all users contained in this disk by charging θ_h to each $h \in \delta(D_{ij})$.

Now, consider a high-degree disk D_{ij} ; then, at some point in time, we have $|D'_{ij}| = k_i$. At this point, we fix the value of β_{ij} and subsequently increase the $\gamma_{h,ij}$ variables and make $\delta(D_{ij}) = D'_{ij}$. When this disk is declared open $|D'_{ij}| \leq k_i$, we have that $p_{ij} = k_i \beta_{ij} + \sum_{h:h \in D_{ij}} \gamma_{h,ij}$. For the users not in $\delta(D_{ij})$, note that $\gamma_{h,ij} = 0$. Hence, $p_{ij} = k_i \beta_{ij} + \sum_{h:h \in \delta(D_{ij})} \gamma_{h,ij}$. Since there are exactly k_i users in $\delta(D_{ij})$, we have $p_{ij} = \sum_{h:h \in \delta(D_{ij})} (\beta_{ij} + \gamma_{h,ij}) = \sum_{h:h \in \delta(D_{ij})} \theta_h$. Thus, the cost of disk D_{ij} is charged to all users in $\delta(D_{ij})$.

Finally, Algorithm 1 selects up to m disks. Therefore, each user in C can be charged up to m times. Formally, $|\{D_{ij} : h \in \delta(D_{ij}), D_{ij} \in \hat{\mathcal{D}}\}| \leq m, h \in U$. \square

Theorem 1. The CMPC-PD algorithm returns an m -approximation for the capacitated minimum power cover problem in polynomial time.

Proof. (Approximation ratio): For the cover $\hat{\mathcal{D}}$ constructed by the algorithm, we show that $\sum_{D_{ij} \in \hat{\mathcal{D}}} p_{ij} \leq f \cdot \text{OPT}$, where OPT is the value of an optimum solution to the CMPC problem. Let Z_{LP}^* be the optimal value of the linear programming relaxation of (2). It is sufficient to show that $\sum_{D_{ij} \in \hat{\mathcal{D}}} p_{ij} \leq f \cdot (\sum_{h \in U} \theta_h - \sum_{i \in S} \mu_i)$ for the final dual solution θ and α , since by weak duality, we know that for any dual feasible solution θ and α , $\sum_{h \in U} \theta_h - \sum_{i \in S} \mu_i \leq Z_{LP}^*$. Thus, since the LP is a relaxation, $Z_{LP}^* \leq \text{OPT}$.

According to Lemma 2, we can charge the cost of each chosen disk D_{ij} to the users in $\delta(D_{ij})$ at most m times. Thus, we have that

$$\begin{aligned} \sum_{D_{ij} \in \widehat{\mathcal{D}}} p_{ij} &= \sum_{D_{ij} \in \widehat{\mathcal{D}}} (k_i \beta_{ij} + \sum_{h: h \in D_{ij}} \gamma_{h,ij} - \mu_i) \\ &= \sum_{D_{ij} \in \widehat{\mathcal{D}}} \sum_{h: h \in \delta(D_{ij})} \theta_h \\ &= \sum_{h \in \mathcal{U}} \theta_h \cdot |\{D_{ij} : h \in \delta(D_{ij}), D_{ij} \in \widehat{\mathcal{D}}\}| \\ &\leq m \cdot \sum_{h \in \mathcal{U}} \theta_h \\ &\leq m \cdot \text{OPT} \end{aligned}$$

where the second equality is derived from Lemmas 2 and $\mu_i = 0, i \in S$.

(Polynomial Running Time): The **while** loop iterates at most n times to cover all users. Line 4 takes $O(mn^2)$ time to increase the dual variables. Lines 5–7 update several sets in $O(n)$ time. Thus, the CMPC-PD algorithm runs in polynomial time $O(mn^3)$. \square

4. Experimental Results

We use the PD algorithm proposed above and synthetic data to conduct practical experiments to simulate the power control for servers in edge networks. The experiments ignore the vertical distribution of the two facilities, mapping their positions to a two-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 with eight cores and 16 threads at 2.9 GHz, 16 GB memory, and a hard disk capacity of 1 TB.
2. The entire system includes two facilities, namely, servers and users, which are distributed randomly in a two-dimensional space. The capabilities of each server are limited, and all users must be covered by some server. In this experiment, we randomly set the capacity of each server according to its average capacity \bar{k} ; thus, the capacity of each server varies around \bar{k} . To ensure that the total system capacity satisfies the requirement of covering all users, if $\sum_{i \in S} k_i < n$, we artificially increase the gap so that $\sum_{i \in S} k_i = n$; if $\sum_{i \in S} k_i \geq n$, no operation is performed.
3. The experimental data are generated in an average distribution over a given range, so each experiment is repeated 50 times. The final results are averaged to reduce the impact of randomness.
4. The IP utilization rate of a server is the ratio of the number of users covered by it to its capacity.
5. The variance in the IP utilization is defined by Equation (7):

$$s^2 = \frac{\sum_{i \in S} (|L_i| - n/m)^2}{m}. \quad (7)$$

According to the property of variance, the smaller the value of s^2 is, the more balanced the number of users covered by each server.

6. In this section, we compare the PD algorithm with the optimal and nearest capable server (NCS) approaches. The OPT approach uses IBM's open source tool CPLEX to obtain the optimal solution to the CMPC problem. If we do not obtain the optimal solution within 10 min, we stop CPLEX. The NCS approach is a greedy-based method that can be used to solve the CMPC problem. To determine which server covers which user during each iteration, the algorithm selects the closest server–user point pair for which the server still has capacity.

Table 2. Configuration of experimental parameters.

Param	Description	Value
α	Power parameter in Equation (1)	[1, 2]
c	Power parameter in Equation (1)	1
m	# of servers	[1, 10]
n	# of users	[20, 500]
\bar{k}	Average capacity of all servers	[0, 200]
k_i	Capacity of server i	[0, 200]
K	Total capacity of all servers	$m \cdot \bar{k}$
l	Side length of facility distribution area	100
λ	Ratio of the side length of the server distribution area to l	[0, 1]
$p_i^X (p_i^Y)$	X (Y) coordinate of server i	[0, 100]
$p_j^X (p_j^Y)$	X (Y) coordinate of user j	[0, 100]

4.1. Impact of the Number of Users

In this experiment, we analyzed the impact of changes in the number of users on the CMPC problem. The main foci are the total system power, algorithm execution time, and variance in the server's capacity utilization. n is gradually increased from 20 to 200. All facilities are distributed in an area with a side length of 100. The average capacity of each server is $\bar{k} = 50$. Therefore, the total capacity of all servers is $K = 500$. For the two constants in Equation (1), $c = 1$ and $\alpha = 2$. When the number of users is greater than 200, the optimal solution of a single instance cannot be obtained within 10 min. So, the power and execution time of CPLEX seems to be 0 when $n = 200$ in Figure 2.

Figure 2a shows the variation in the total power as the number of users increases for the three approaches. Overall, the total power obtained by the three methods increases as the number of users increases. This is consistent with our intuition. For example, consider several servers installed around a shopping mall. It is certain that the number and capacity of these servers do not change in one day. However, users in the mall change over time. When the number of consumers increases, the probability that the server serves more distant customers increases. The statistics of the total power should then increase. In addition, the DP result is closer to OPT than the NCS result. This finding indicates the superiority of our method. Although our results are not as good as the optimal solution, Figure 2b shows the considerable time cost required to arrive at the optimal solution.

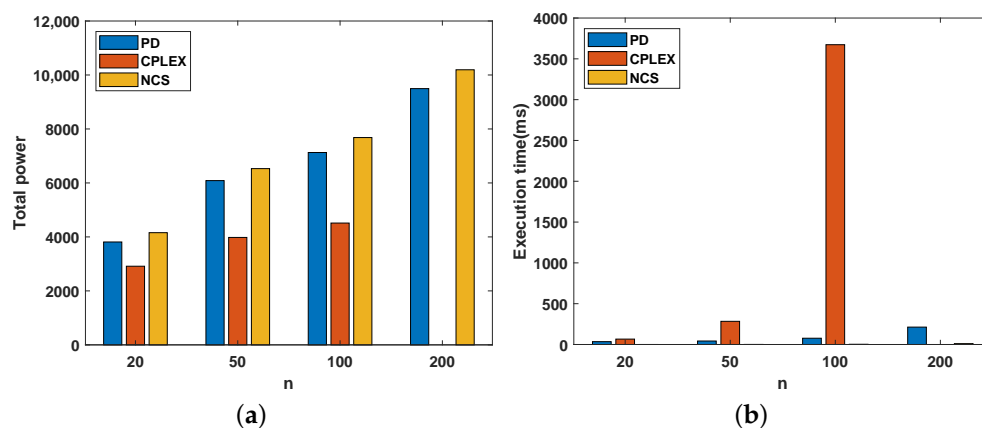


Figure 2. System performance with different numbers of users. (a) Total power. (b) Execution time (ms).

4.2. Impact of the Number of Servers and K

In this section, we study the impact of the number of servers and the total capacity K on the algorithm. This investigation helps us to determine whether to reduce the total

power of the system by increasing the number or capacity of the servers when the number of users is stable. In this experiment, we assume that there are 100 users, and m and K increase from 1 to 8 and 100 to 200, respectively. The values of the other variables are the same as those in Section 4.1.

Figure 3a clearly shows that when the total capacity of the system is sufficient ($K \geq 150$), the total power does not change significantly with increasing m . This result indicates that sufficient capacity generates less power. When the system capacity is tight, increasing the number of servers causes the PD algorithm to encounter the issue of local optimization. Combined with the results in Figure 3a,b, this finding leads to a rapid increase in the total power. However, when considering the cost of the server itself, with the assumption of a stable number of users, we can use fewer servers. In this case, the PD algorithm shows good performance regardless of resource constraints, as shown in Figure 3b. In addition, Figure 3b shows that as the number of servers increases, the approximate ratio of the PD algorithm increases. This result confirms the conclusion of Theorem 1.

In the introduction, we assume that a server can serve every user. In an optimal solution, a server serves the relatively close user and rarely the user who is farthest away. The approach proposed in this paper leads to servers serving more costly users due to local optimality, which is a challenge for all approximation algorithms. In Figure 3b, we can see that the approximation ratio will only exceed 2 in a few cases where $m \in \{4, 5\}$ and $k \in \{100, 110, 120\}$. When the approximation ratio is less than 2, the average radius of the disk obtained by our approach does not exceed $2^{\frac{1}{\alpha}}$ times the optimal solution.

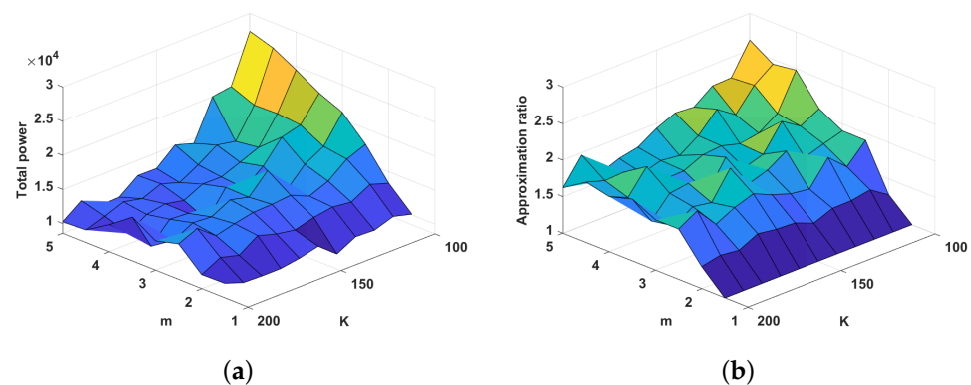


Figure 3. System performance with different numbers of servers and values of K . (a) Total power. (b) Approximation ratio.

4.3. Impact of the Number of Servers and λ

Server location problems are also an important research topic. Although this paper does not discuss how to determine the location of the server, we can explore the impact of the server distribution on the PD algorithm by controlling the concentration of the servers. We assume that the two facilities are distributed in a square with a side length of $l = 100$ and use λ to control the distribution area of the server. The side length of this area is λl , and the center of this region is located at $(l/2, l/2)$. The number of servers m is increased from 1 to 8, with $K = 150$ and $n = 100$. The values of the other variables are the same as those in Section 4.1.

Figure 4a shows that when the number of servers is less than or equal to 3, the more concentrated the servers are, the lower the total power is. However, as the number of servers gradually increases, different concentrations lead to power changes. When the number of servers is increased to eight, the total power ranking result is opposite to the previous result. Thus, when we need to place servers in an area, to meet the requirement of the system, we need to place servers only near the center of the area; however, if the QoS of users is considered, the servers should be evenly placed in this area rather than only close to the center.

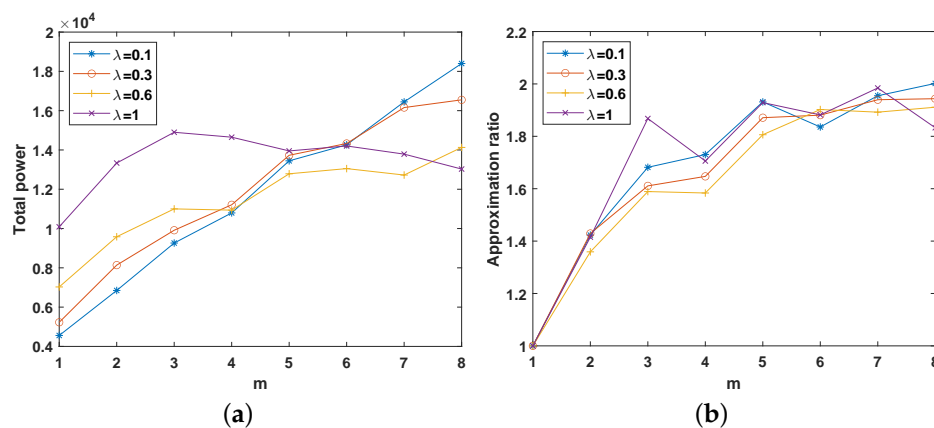


Figure 4. System performance with different numbers of servers and values of λ . (a) Total power. (b) Approximation ratio.

4.4. Impact of Different Values of α

In Equation (1), α is an important parameter that represents the signal attenuation coefficient. Variations in α inevitably affect the total power. Therefore, in this section, we explore the impact of α on the total power and the performance of the PD algorithm. In this experiment, the same dataset was used for different values of α . This dataset was developed according to the case when $m = 6$ and $K = 150$, as described in Section 4.2.

Figure 5a shows that the total power of the PD and CPLEX algorithms increases exponentially with increasing α . The reason for this result is clear. A larger value of α increases the cost of the PD algorithm choosing a disk that differs from the optimal solution. Therefore, as shown in Figure 5b, the approximation ratio of the PD algorithm does not increase with small increases in α . Although the influence of α on the approximation ratio is not obvious, the variance in the IP utilization decreases with increasing α . This result shows that users are increasingly evenly served by the server.

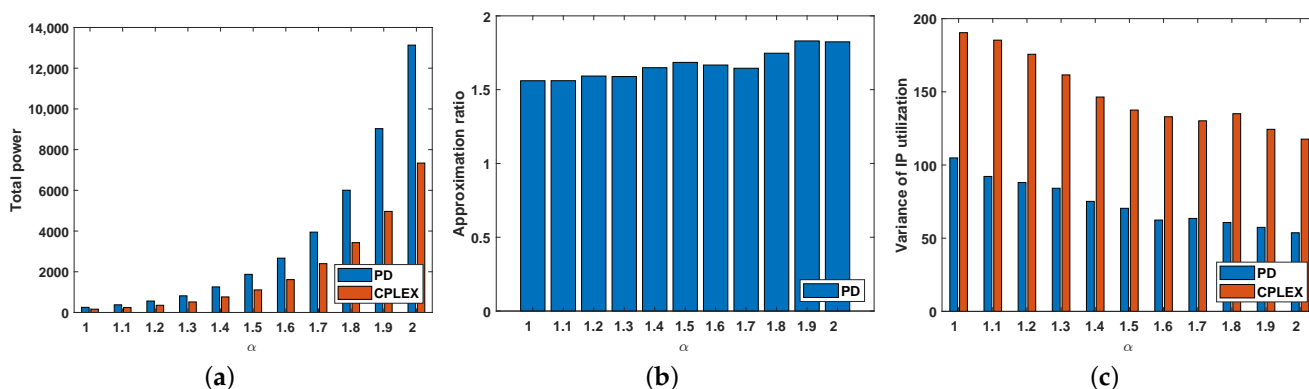


Figure 5. System performance with different values of α . (a) Total power. (b) Approximation ratio. (c) Variance in IP utilization.

4.5. Evaluating the Gap between Assumptions and Practical Scenarios

We propose a variable named SNR_{min} in the introduction section. When the SNR between server s and user u is greater than SNR_{min} , s can serve u . However, there is still some gap between this assumption and the practical scenarios. We need a further evaluation of this gap that is not confined to the analysis at the end of Section 4.2.

We assume the default coverage radius r^* is the same for all servers. The radius r^* is determined by ρ , which is defined as the proportion of users that can be covered when the radius of servers is r^* . For example, when $\rho = 0.5$, r^* is equal to the radius of the server signal disk when it can just cover 50% of the users. In this experiment, $m = 4$, and K and ρ

increase from 100 to 200 and 0.5 to 1, respectively. In addition, we use \bar{E} to evaluate the gap between assumptions and practical scenarios, which is defined as follows:

$$\bar{E} = \frac{\sum_{i \in S} (r_i - r^*)}{m}, \quad (8)$$

where r_i is the radius of server i from the solution of Algorithm 1. As defined in Equation (8), \bar{E} indicates the average expansion of the radius of the solution obtained by our approach compared to r^* .

As we can see in Figure 6, if $\rho \geq 0.7$ such that $\bar{E} \leq 1$, it indicates that the server signal disk does not need to be increased by more than a factor of 1 radius when by default, the server can cover more than 70% of the users. Under the line-of-sight channel model, we think this expansion is acceptable. However, \bar{E} becomes impractical when r^* is close to 0.5 or the total capacity is very constrained. We can obtain recommendations on whether additional servers should be established from this result.

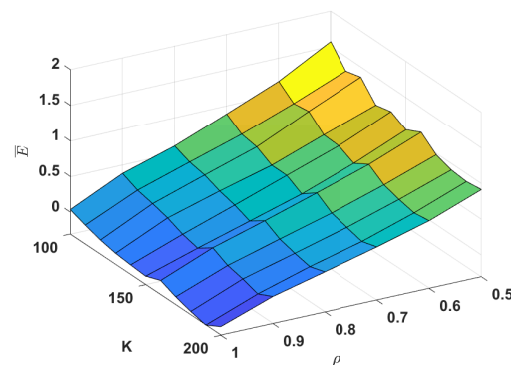


Figure 6. Average expansion \bar{E} with different values of K and ρ .

5. Conclusions

Signal coverage consumes considerable energy in wireless networks; thus, in this paper, we studied how to assign the appropriate power to servers to reduce energy consumption. We built a signal coverage model based on the CMPC problem in edge networks and developed an m -approximation primal–dual-based algorithm. The numerical results show satisfactory performance.

In this paper, we considered that all users need to be covered by servers. However, servers need not serve users because of high energy costs, poor communication quality, etc. Therefore, server blocking probability, packet loss, and throughput metrics should focus on future work. These considerations can affect the decision to choose the optimal connecting station for the wireless user. In addition, the power allocation problem for each user after power control for servers would be a further work. Thus, the signal interference between users has to be considered.

Author Contributions: Conceptualization, Q.Z., W.L. and Q.S.; Formal analysis, Q.Z. and W.L.; Funding acquisition, Q.Z. and X.Z.; Methodology, Q.Z. and W.L.; Software, Q.Z.; Supervision, W.L. and X.Z.; Validation, Q.Z., W.L. and X.Z.; Visualization, Q.Z.; Writing—original draft, Q.Z.; Writing—review and editing, Q.Z., W.L. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China [Nos. 12071417, 61762091, and 62062065], the 13th Postgraduate Innovation Project of Yunnan University [No. 2021Z079] and the Scientific Research Fund Project of Yunnan Provincial Department of Education [No. 2022J0002].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset link: <https://github.com/zqh-ynu/PD-based-data> (accessed on 1 August 2022).

Acknowledgments: The authors would like to thank all of the cited authors and the anonymous reviewers in this article for their helpful suggestions.

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

References

1. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile Edge Computing: A Survey. *IEEE Internet Things J.* **2018**, *5*, 450–465. [[CrossRef](#)]
2. IEA. Data Centres and Data Transmission Networks. 2021. Available online: <https://www.iea.org/reports/data-centres-and-data-transmission-networks> (accessed on 1 September 2022).
3. Miretti, L.; Cavalcante, R.L.G.; Stanczak, S.; Schubert, M.; Boehnke, R.; Xu, W. Closed-form max-min power control for some cellular and cell-free massive MIMO networks. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022.
4. Dai, Y.; Liu, J.; Sheng, M.; Cheng, N.; Shen, X. Joint Optimization of BS Clustering and Power Control for NOMA-Enabled CoMP Transmission in Dense Cellular Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1924–1937. [[CrossRef](#)]
5. Chincoli, M.; Liotta, A. Self-Learning Power Control in Wireless Sensor Networks. *Sensors* **2018**, *18*, 375. [[CrossRef](#)] [[PubMed](#)]
6. Moltafet, M.; Leinonen, M.; Codreanu, M.; Pappas, N. Power Minimization for Age of Information Constrained Dynamic Control in Wireless Sensor Networks. *IEEE Trans. Commun.* **2022**, *70*, 419–432. [[CrossRef](#)]
7. Ronnow, D.; Handel, P. Nonlinear Distortion Noise and Linear Attenuation in MIMO Systems—Theory and Application to Multiband Transmitters. *IEEE Trans. Signal Process.* **2019**, *67*, 5203–5212. [[CrossRef](#)]
8. Alt, H.; Arkin, E.M.; Brönnimann, H.; Erickson, J.; Fekete, S.P.; Knauer, C.; Lenchner, J.; Mitchell, J.S.B.; Whittlesey, K. Minimum-cost coverage of point sets by disks. In Proceedings of the Twenty-Second Annual Symposium on Computational Geometry—SCG '06, Sedona, AZ, USA, 5–7 June 2006; ACM Press: New York, NY, USA, 2006; Volume 2006, p. 449.
9. Bilò, V.; Caragiannis, I.; Kaklamani, C.; Kanellopoulos, P. Geometric Clustering to Minimize the Sum of Cluster Sizes. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3669, pp. 460–471.
10. David, P.W.; David, B.S. *The Design of Approximation Algorithms*; Cambridge University Press: New York, NY, USA, 2011.
11. Turan, B.; Uribe, C.A.; Wai, H.T.; Alizadeh, M. Resilient Primal–Dual Optimization Algorithms for Distributed Resource Allocation. *IEEE Trans. Control Netw. Syst.* **2021**, *8*, 282–294. [[CrossRef](#)]
12. Wang, Z.; Xu, D. Online Job Scheduling in Mobile Edge Computing based on Primal-Dual Method. In Proceedings of the 2021 Computing, Communications and IoT Applications (ComComAp), Shenzhen, China, 26–28 November 2021; pp. 47–52.
13. Zhou, R.; Li, Z.; Wu, C.; Huang, Z. An Efficient Cloud Market Mechanism for Computing Jobs with Soft Deadlines. *IEEE/ACM Trans. Netw.* **2017**, *25*, 793–805. [[CrossRef](#)]
14. Zhang, Q.; Li, W.; Su, Q.; Zhang, X. 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. [[CrossRef](#)]
15. Liu, X.; Li, W.; Xie, R. A primal-dual approximation algorithm for the k-prize-collecting minimum power cover problem. *Optim. Lett.* **2021**. [[CrossRef](#)]
16. Liu, X.; Li, W.; Dai, H. Approximation algorithms for the minimum power cover problem with submodular/linear penalties. *Theor. Comput. Sci.* **2022**, *923*, 256–270. [[CrossRef](#)]
17. Dai, H.; Deng, B.; Li, W.; Liu, X. A note on the minimum power partial cover problem on the plane. *J. Comb. Optim.* **2022**, *44*, 970–978. [[CrossRef](#)]
18. Lyu, J.; Zeng, Y.; Zhang, R.; Lim, T.J. Placement Optimization of UAV-Mounted Mobile Base Stations. *IEEE Commun. Lett.* **2017**, *21*, 604–607. [[CrossRef](#)]
19. Bandyapadhyay, S.; Bhowmick, S.; Inamdar, T.; Varadarajan, K. Capacitated Covering Problems in Geometric Spaces. *Discret. Comput. Geom.* **2020**, *63*, 768–798. [[CrossRef](#)]
20. Varadarajan, K. Weighted Geometric Set Cover via Quasi-Uniform Sampling. In Proceedings of the Forty-Second ACM Symposium on Theory of Computing, Cambridge, MA, USA, 6–8 June 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 641–648.
21. Chan, T.M.; Grant, E.; Könemann, J.; Sharpe, M. 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, Kyoto, Japan, 17–19 January 2012; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2012; pp. 1576–1585.
22. Bansal, N.; Pruhs, K. Weighted Geometric Set Multi-cover via Quasi-uniform Sampling. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7501 LNCS, pp. 145–156.
23. Wang, S.; Zhang, X.; Yan, Z.; Wenbo, W. Cooperative Edge Computing with Sleep Control Under Nonuniform Traffic in Mobile Edge Networks. *IEEE Internet Things J.* **2019**, *6*, 4295–4306. [[CrossRef](#)]

24. Ali, Z.; Jiao, L.; Baker, T.; Abbas, G.; Abbas, Z.H.; Khaf, S. A Deep Learning Approach for Energy Efficient Computational Offloading in Mobile Edge Computing. *IEEE Access* **2019**, *7*, 149623–149633. [[CrossRef](#)]
25. Li, L.; Zhang, X.; Liu, K.; Jiang, F.; Peng, J. An Energy-Aware Task Offloading Mechanism in Multiuser Mobile-Edge Cloud Computing. *Mob. Inf. Syst.* **2018**, *2018*, 7646705:1–7646705:12. [[CrossRef](#)]
26. Gu, B.; Zhou, Z.; Mumtaz, S.; Frascolla, V.; Bashir, A.K. Context-Aware Task Offloading for Multi-Access Edge Computing: Matching with Externalities. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [[CrossRef](#)]
27. Bar-Yehuda, R.; Even, S. A linear-time approximation algorithm for the weighted vertex cover problem. *J. Algorithms* **1981**, *2*, 198–203. [[CrossRef](#)]