Hindawi
Mobile Information Systems
Volume 2020, Article ID 1501403, 12 pages
https://doi.org/10.1155/2020/1501403

Research Article

Task Allocation Optimization Scheme Based on Queuing Theory for Mobile Edge Computing in 5G Heterogeneous Networks

Jianbin Xue, Zesen Wang, Yonggang Zhang, and Lu Wang

School of Computer and Communication, Lanzhou University of Technology, LanZhou, GanSu 730050, China

Correspondence should be addressed to Zesen Wang; wang_zesen@hotmail.com

Received 25 November 2019; Revised 28 April 2020; Accepted 8 May 2020; Published 29 May 2020

Academic Editor: Raul Montoliu

As an indispensable key technology in 5G Internet of Things (IoT), mobile edge computing (MEC) provides a variety of computing and services at the edge of the network for energy-limited and computation-constrained mobile devices (MDs). In this paper, we use the multiaccess characteristics of 5G heterogeneous networks and queuing theory. By considering the heterogeneity of base stations, we establish the waiting and transmission consumption model when tasks are offloaded. Then, the problem of jointly optimizing the energy and delay consumption of MDs is proposed. We adopt an optimization scheme based on task assignment probability; moreover, the task assignment algorithm based on quasi-Newton interior point (TA-QNIP) method is developed to solve the optimization issue. Compared with the Newton interior point algorithm, the proposed algorithm accelerates the convergence speed and reduces the complexity of the algorithm and is closer to the objective function optimal solution. The simulation results demonstrate that the proposed method can reduce the total consumption of MDs effectively; furthermore, the performance of the algorithm is proved.

1. Introduction

With the widespread deployment of Internet of Things (IoT) in 5G era [1], mobile applications such as natural language processing, virtual reality, and interactive games have greatly enriched our lives [2]. However, mobile devices (MDs) with constrained computing power and battery capacity could not handle the huge amount of data generated by mobile applications [3, 4]. To avoid this mismatch of resources, researchers have come up with various cloud-based solutions [5]. By utilizing the abundant resources in the center cloud, the computing intensive tasks of mobile applications can be offloaded, thus reducing the workload of IoT devices (smart furniture, smart glasses, and industrial sensor, etc. [6]) and shortening the computing delay [7]. However, due to the multihop structure of the core network, the delay between the MDs and the center cloud is too long. If a lot of IoT devices request cloud services from the same node at the same time, the backhaul link will be heavily burdened [8].

In order to alleviate the burden of the core network, mobile edge computing (MEC) is gradually proposed and brings cloud computing functions to the edge of the network [9]. With the help of MEC, MDs can offload tasks to the edge of the network, instead of using servers located in the center of the network which is far away from MDs [10]. This greatly improves the offloading efficiency of the device, while reducing the energy consumption of the device and shortening the backhaul delay [11].

In recent years, with the progress of 5G communication, MEC based on 5G architecture has been studied by many scholars [12–16]. In the 5G network, a heterogeneous network composed of a macro base station and a small base station is a common form of 5G architecture [17]. Since the macro base station and the small base station are located at different locations and the configured hardware levels are different, they have different effects in the small cell network. Therefore, our goal is to improve the offloading efficiency of MEC in 5G environments by considering the performance differences of the base stations.

The MD has the offloading decision right. The question of whether, how much, and what to offload is determined by monitoring various parameters through the terminal system parser, such as the size of the data to be offloaded, the time delay caused by the offloading task, or the amount of energy required to execute locally. Compared with the deterministic task model, the average energy consumption and execution latency of the stochastic task model system have a stronger correlation, so designing an efficient computation offloading scheme is more challenging. Therefore, compared with the computation offloading optimization scheme of the deterministic task model, the design of MEC systems with random task arrival is a less explored field.

Aiming at the problem of MDs' consumption in MEC, this paper designs a task offloading scheme with base station collaboration based on 5G heterogeneous networks. The purpose of this scheme is to improve the low efficiency of task offloading caused by congestion. Different from the existing literature which only optimizes the delay or energy consumption, this paper reduces the overall consumption of MDs by optimizing the energy and delay consumption of MDs jointly. The contributions and innovations of this paper are as follows:

(i) In this paper, a base station cooperative task offloading scheme based on 5G heterogeneous networks is designed. By using queuing theory, the waiting energy and delay consumption of tasks to be offloaded are considered jointly; in addition, the offloading decision problem is transformed into the task assignment probability problem.

(ii) The optimization goal of the MDs-centered energy and delay consumption minimization is established. Then, the joint optimization of the total consumption of MDs with different demands on delay and energy consumption is accomplished by allocating the task assignment probability.

(iii) In order to solve the problem of consumption minimization, the task assignment algorithm based on quasi-Newton interior point (TA-QNIP) method is proposed. In addition, the complexity of the proposed algorithm is discussed and the convergence performance is verified.

The rest of this paper is arranged as follows: we summarize the related work in Section 2. Section 3 establishes a complete system model. In Section 4, we formulate the optimization problem of minimizing the energy and delay consumption of MDs. In Section 5, the Newton algorithm is briefly introduced; accordingly, the TA-QNIP method is proposed, and then we analyze the algorithm complexity. Simulation results are discussed in Section 6. Finally, we summarize the work of the full text in Section 7.

2. Related Work

At present, there are some related works focusing on MEC under 5G architecture. Wang et al. [12] improved system revenue by jointly optimizing computation offloading, resource allocation, and content caching. Zhang et al. studied how to decrease the computation offloading delay of the MEC system in 5G architecture [13]. In order to meet the key requirements of 5G networks for low latency and high reliability, the authors in [18, 19] proposed a joint optimization problem for computation offloading of MEC systems based on delay and reliability. However, the above works did not take into account the basic characteristics of the multiaccess feature of the 5G architecture. Combining the multiaccess characteristics of 5G heterogeneous networks, Zhang et al. [10] proposed the MEC energy-efficient computing offload mechanism in 5G heterogeneous networks, which effectively reduced energy consumption through joint optimization of offloading strategies and cellular network resource allocation. Considering the constraints of computing ability and service delay requirements, Yang et al. [4] developed an energy optimization scheme based on artificial fish swarm algorithm to minimize the entire energy consumption of the system. The above works have achieved good results in the optimization of system energy consumption when using the 5G multiaccess feature to design scheme, but the optimization of task processing delay is not considered at the same time.

Recent survey [20] has shown that there are two types of computation offloading: binary offloading and partial offloading. Computing tasks cannot be divided into subtasks in binary offloading. The entire task must be executed on the local or MEC servers [21, 22], thus reducing the flexibility of the task processing in practical application environments. However, in partial offloading, subtasks can choose different offloading ways based on different processing requirements and optimal system efficiency [20]. In view of task separability, Guan et al. [23] designed an efficient task offloading scheme for IoT based on cooperative communication in the mobile cloud computing system. Pang et al. [24] studied the problem of delay-driven collaborative task calculation in fog wireless access network. Although previous research studies make good use of the separability of the task to establish a model, but did not fully utilize the small cell heterogeneous network characteristics under the 5G architecture.

Applying queuing theory to MEC is the focus of scholars in recent years. In [25, 26], the energy consumption, execution delay, and price cost of the offloading process in the MEC system are studied in depth by using queuing theory. The authors in [27] based on Lyapunov optimization developed an online algorithm and the theoretical boundary of the algorithm in terms of average power consumption and average queue length was proved. In [28], different queue models were applied to study the energy cost and delay performance, and the optimal solution was solved by the semismooth Newton method of Armijo line search. Yang et al. [29] used a probabilistic optimization scheme to jointly optimize energy costs and packet congestion and effectively controlled congestion of edge servers by grouping with different priorities. Li [30] established a queuing model for one MD and multiple heterogeneous edge servers, and in the past work, the heterogeneity of the edge server was introduced for the first time to study the optimization of the computational offload strategy. The authors in [31]

established three different queue models based on MD, cloudlet, and central cloud and then conducted in-depth research on the optimization of energy consumption and execution delay of cloudlet-assisted task offloading. However, the previous works did not consider the impact of congestion caused by the bearer capacity of the base station on the MEC system.

Different from previous studies, under the 5G MEC heterogeneous networks, we proposed the object that jointly optimizing the energy and delay consumption of MDs. Through queuing theory, we comprehensively considered the differences of heterogeneous base stations and established the waiting consumption and transmission consumption model during task offloading. A task assignment algorithm based on quasi-Newton interior point method is proposed. MDs can reasonably assign offloading tasks according to the congestion degree of the system to minimize the total consumption. Finally, the complexity of the proposed algorithm is discussed and the convergence is verified.

3. System Model

In this part, we first establish the system model, including network model, local model, transmission model, and edge cloud model, and then, the problem model to be optimized is established.

3.1. Network Model. In the edge cloud network, the processing tasks generated by each MD can be executed locally or offloaded to the MEC server for computing. In order to save energy consumption and shorten time delay, we design an uncertain task offloading model based on queuing theory. As shown in Figure 1, we consider a set of MDs in the system, which is denoted by MD_i ($i = 1, 2, 3 \ldots N$), a macro base station (MBS) equipped with MEC servers and a small base station (SBS). The MBS and SBS are connected by a fiber link. Due to the different types of tasks generated by each MD, the generated task requests are random. We assume that a task consists of multiple subtasks. In general, the computing tasks randomly generated by MDs can be processed locally, or some tasks can be offloaded to MEC servers through MBS for processing. In this model, MDs can also offload some tasks to MEC servers through SBS, thus reducing the processing pressure of MBS. Based on the queuing theory [32], we consider that the processing model of the local is the M/M/1 queue, and the model of the task transmission is the M/M/c queue. Figure 2 shows the task queuing process. Suppose the task generation rate of the MD_i is λ_i (measured by the number of generation tasks on per unit of time, e.g., second), the size of request data is θ_i. The probability that the task generated by the MD_i is locally executed is p_i^l, the probability that the task is processed by the edge cloud is p_i^c, and p_i^m and p_i^s are the probability that the MD_i offloads the task through the macro base station and the probability that the task is offloaded through the small base station, respectively, where $p_i^c = p_i^s + p_i^m$. Due to the nature of Poisson distribution, we assume that the service request offloaded to the MEC servers follows the Poisson process with an average rate of $p_i^c\lambda_i$, and the locally processed service request follows the Poisson process with an average rate of $p_i^l\lambda_i$.

3.2. Local Model. The consumption of MDs performing tasks locally is divided into two parts: computation and task response consumption. In order to simplification, we only consider the task response consumption. u_i^D represents the execution capability of MD_i, and l_i^D represents the proportion of CPU occupied by MD_i. Since the generation of tasks is distributed negatively exponentially, the task processing model is considered to be M/M/1 queue on the MD side. By Little's Law [33], the local task response time is $T = (1/u)/(1-\eta)$, and the queuing efficiency is $\eta = \lambda/u$, where λ and u are task arrival rate and device service rate, respectively. Therefore, the average response time and energy consumption of locally executed tasks are as follows:

$$T_i^D = \frac{1}{u_i^D\left(1 - l_i^D\right) - p_i^l\lambda_i}, \tag{1}$$

$$E_i^D = \xi_i T_i^D = \xi_i \frac{1}{u_i^D\left(1 - l_i^D\right) - p_i^l\lambda_i}, \tag{2}$$

where ξ_i represents the response energy consumption coefficient of MD_i.

3.3. Transmission Model. In the edge heterogeneous network, in addition to MBS, the SBS is regarded as the cooperative base station within the MBS coverage. In order to effectively utilize the spectrum resources, both MBS and SBS work in the same frequency band [10]. It is assumed that the bandwidth of the channels in the system is the same, which is denoted by B. Since this paper mainly researches task assignment problems and alleviates system congestion, in order to simplify the model, we assume that the interference can be negligible because channels allocated to MDs for computation offloading are all orthogonal [34, 35]. Therefore, we can calculate the uplink transmission rate of the MD_i offloading tasks to the MBS:

$$R_i^m = B\log_2\left(1 + \frac{P_i^m H_i^m}{\sigma^2}\right). \tag{3}$$

Similarly, the uplink transmission rate of the MD_i offloading tasks to the SBS is given by

$$R_i^s = B\log_2\left(1 + \frac{P_i^s H_i^s}{\sigma^2}\right), \tag{4}$$

where σ^2 is the Gaussian white noise power and P_i^m and P_i^s denote the transmission power of MDs to MBS and SBS, respectively. The transmission power can be determined by the power control mechanism of MBS and SBS [36]. In addition, $P_i^{\max}$ is the maximum transmission power of the MD_i, $H_i$is the channel gain between the MD_i and base stations, $H_i = 127 + 30 \times \log d_i$, and d_i is the distance between the MD_i and base stations [37].

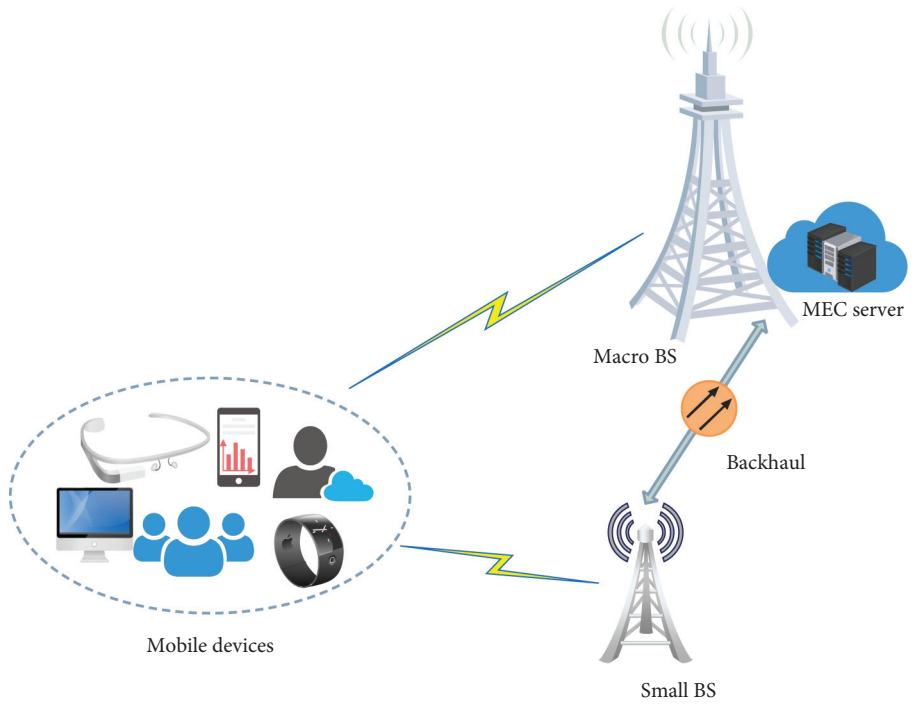

Figure 1: Base station cooperative task offloading model based on 5G heterogeneous networks.

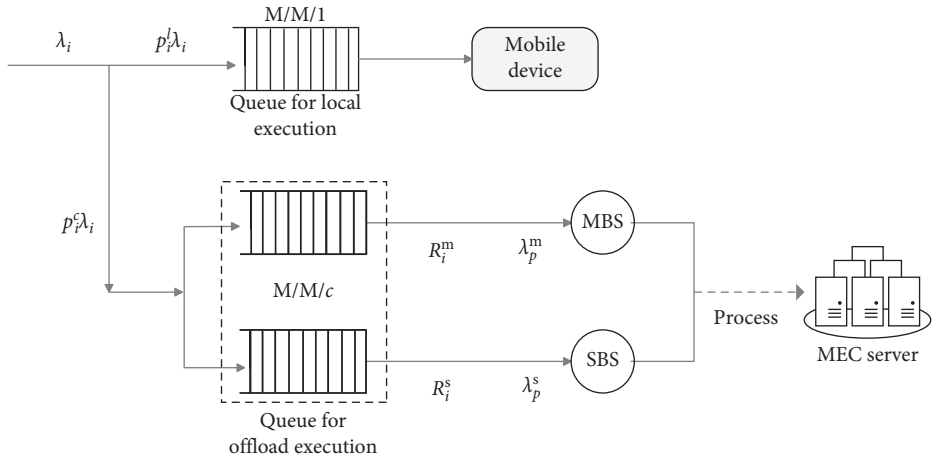

Figure 2: Queuing model of task request processing.

For MBS and SBS, the maximum acceptable task arrival rate is λ_{max}^m and λ_{max}^s, respectively, and the sum of all task request rates from different MDs is expressed as follows:

$$\lambda_{total}^m = \sum_{i=1}^{N} \lambda_i p_i^m, \tag{5}$$

$$\lambda_{total}^s = \sum_{i=1}^{N} \lambda_i p_i^s. \tag{6}$$

Then, the actual task arrival rate on the MBS is $\lambda_p^m = \min[\lambda_{total}^m, \lambda_{max}^m]$, and the actual task arrival rate on the SBS is $\lambda_p^s = \min[\lambda_{total}^s, \lambda_{max}^s]$. Assume that the service rate of MBS is u^m and the service rate of SBS is u^s. According to the M/M/c queuing model definition, the queue strengths of the tasks to the MBS and the SBS are as follows:

$$\rho^m = \frac{\lambda_p^m}{cu^m}, \tag{7}$$

$$\rho^s = \frac{\lambda_p^s}{cu^s}. \tag{8}$$

Queue strength is a parameter to measure the stability of the system. When $\rho^m < 1$ and $\rho^s < 1$, the average amount of tasks arriving at the system is less than the average amount of tasks leaving the system; therefore, the task waiting time will not be too long caused by the lengthy queue, and at this time, the system is stable. In order to offload computing tasks to the MEC servers, wireless uplink transmissions generate extra energy and delay overhead. The total transmission time includes the transmission time of the uplink and the waiting time

for the task to be offloaded. Therefore, when the MD_i offloads its tasks to the MEC servers through MBS, the transmission delay and energy consumption can be calculated as follows:

$$T_i^m\left(P_i^m\right) = \frac{p_i^m \lambda_i \theta_i}{R_i^m} + W_i^m = \frac{p_i^m \lambda_i \theta_i}{B\log_2\left(1 + \left(P_i^m H_i^m/\sigma^2\right)\right)} + W_i^m, \tag{9}$$

$$\begin{aligned} E_i^m\left(P_i^m\right) &= P_i^m T_i^m\left(P_i^m\right) = P_i^m \frac{p_i^m \lambda_i \theta_i}{R_i^m} + \gamma_i W_i^m \\ &= P_i^m \left(\frac{p_i^m \lambda_i \theta_i}{B\log_2\left(1 + \left(P_i^m H_i^m/\sigma^2\right)\right)}\right) + \gamma_i W_i^m, \end{aligned} \tag{10}$$

where $\gamma_i$is the waiting energy coefficient of MD_i. Similarly, when MD_i offloads its tasks to MEC servers through SBS, the transmission delay and energy consumption can be calculated as follows:

$$T_i^s\left(P_i^s\right) = \frac{p_i^s \lambda_i \theta_i}{R_i^s} + W_i^s = \frac{p_i^s \lambda_i \theta_i}{B\log_2\left(1 + \left(P_i^s H_i^s/\sigma^2\right)\right)} + W_i^s, \tag{11}$$

$$\begin{aligned} E_i^s\left(P_i^s\right) &= P_i^s T_i^s\left(P_i^s\right) = P_i^s \frac{p_i^s \lambda_i \theta_i}{R_i^s} + \gamma_i W_i^s \\ &= P_i^s \left(\frac{p_i^s \lambda_i \theta_i}{B\log_2\left(1 + \left(P_i^s H_i^s/\sigma^2\right)\right)}\right) + \gamma_i W_i^s, \end{aligned} \tag{12}$$

where W_i^m is the waiting time for the task generated by the MD_i to be offloaded to MBS. According to the Little formula, the queuing system with an average arrival rate of λ, in the average sense, the waiting time under the M/M/c queuing system is as follows:

$$W_i^m = \frac{L_q^m}{u^m} = \frac{\left((c\rho^m)^{c-1}\rho^m/c!\left(1-\rho^m\right)^2\right)p_0^m}{u^m}, \tag{13}$$

where the idle probability of MBS is as follows:

$$p_0^m = \left[\sum_{k=0}^{c-1} \frac{\left(c\rho^m\right)^k}{k!} + \frac{\left(c\rho^m\right)^c}{c!\left(1-\rho^m\right)}\right]^{-1}. \tag{14}$$

Similarly, the waiting time for the task generated by MD_i to be offloaded to SBS is given by

$$W_i^s = \frac{L_q^s}{u^s} = \frac{\left((c\rho^s)^{c-1}\rho^s/c!\left(1-\rho^s\right)^2\right)p_0^s}{u^s}, \tag{15}$$

where the idle probability of SBS is as follows:

$$p_0^s = \left[\sum_{k=0}^{c-1} \frac{\left(c\rho^s\right)^k}{k!} + \frac{\left(c\rho^s\right)^c}{c!\left(1-\rho^s\right)}\right]^{-1}, \tag{16}$$

where L_q^m and L_q^s are the average waiting queue length of the task. In addition, the backhaul link rate between SBS and MBS is much higher than that of the wireless link, so the rest of the paper simply omits the backhaul delay [29].

3.4. Edge Cloud Model. After receiving the offloaded task, the MEC server performs the calculation immediately. The maximum workload of the MEC system is limited to the maximum receiving rate, which is expressed as λ_{max}^c. In the system, the total request rate from different MDs is expressed as follows:

$$\lambda_{total}^D = \sum_{i=1}^{N} \lambda_i p_i^c = \sum_{i=1}^{N} \left[\lambda_p^m p_i^m + \lambda_p^s p_i^s\right]. \tag{17}$$

When the MEC servers perform the offloading task completely, the calculated result will be returned to the MD. We omit the time and energy consumption of MDs to receive and process the result, which is similar to [38].

4. Problem Formulation

In the 5G MEC network environment, under the conditions of meeting the maximum task arrival rate limit and task assignment probability constraints, and comprehensively considering the waiting consumption of MDs, we propose the problem of minimizing the delay and energy consumption of MDs based on multi-base station cooperation. Similar to the work in reference [39], the total delay consumption of the user's task processing can be obtained:

$$T_i\left(p_i^l, p_i^m, p_i^s\right) = T_i^D\left(p_i^l\right) + T_i^m\left(p_i^m\right) + T_i^s\left(p_i^s\right), \tag{18}$$

and the total energy consumption of task processing can be obtained:

$$E_i\left(p_i^l, p_i^m, p_i^s\right) = E_i^D\left(p_i^l\right) + E_i^m\left(p_i^m\right) + E_i^s\left(p_i^s\right). \tag{19}$$

Therefore, in the system, the average execution delay and energy consumption of MDs are expressed as follows:

$$T\left(p_i^l, p_i^m, p_i^s\right) = \sum_{i=1}^{N} T_i\left(p_i^l, p_i^m, p_i^s\right), \tag{20}$$

$$E\left(p_i^l, p_i^m, p_i^s\right) = \sum_{i=1}^{N} E_i\left(p_i^l, p_i^m, p_i^s\right). \tag{21}$$

Since this paper considers the multiobjective optimization of MDs' energy and delay consumption, the transmission consumption between base stations is ignored. Considering that MEC servers have powerful computing ability, the computing energy and delay consumption of the MEC are ignored in this paper. Therefore, the objective function and the restriction conditions are as follows:

$$\begin{aligned} &P1: \min_{\left\{p_i^l, p_i^m, p_i^s\right\}} \left\{T\left(p_i^l, p_i^m, p_i^s\right), E\left(p_i^l, p_i^m, p_i^s\right)\right\}, \\ &C1: p_i^l \lambda_i < u_i^D\left(1 - l_i^D\right) \quad (i = 1, 2, 3 \ldots N), \\ &C2: \lambda_{total}^D < \lambda_{max}^c, \\ &C3: 0 \le P_i^m + P_i^s \le P_i^{max} \quad (i = 1, 2, 3 \ldots N), \\ &C4: p_i^s + p_i^m + p_i^l = 1 \quad (i = 1, 2, 3 \ldots N), \\ &C5: p_i^l > 0, p_i^m > 0, p_i^s > 0 \quad (i = 1, 2, 3 \ldots N), \\ &C6: \sum_{i=1}^{N} \lambda_i p_i^m < \lambda_{max}^m \quad (i = 1, 2, 3 \ldots N). \\ &C7: \sum_{i=1}^{N} \lambda_i p_i^s < \lambda_{max}^s (\mathrm{i} = 1, 2, 3 \ldots N). \end{aligned} \tag{22}$$

Here, $C1$ indicates that the local arrival rate of the task should be less than the remaining execution capacity of the

local device. $C2$ ensures that the total task arrival rate offloaded to the edge cloud system does not exceed the maximum acceptable rate of the servers. When MDs offload the task, the transmit power strength should satisfy $C3$. The probability of offloading in different ways for different tasks should meet $C4$ and $C5$. $C6$ and $C7$ prevent the base station from being overloaded and maintain the stability of the system.

Notice that P1 is a multiobjective nonlinear optimization problem with multiple constraints. In order to satisfy the different demands of MD_i in various application environments, we introduce the delay weight factor α, and then the energy consumption weight factor is $(1-\alpha)$, where $0 \le \alpha \le 1$, so $P1$ can be transformed into the following form:

$$P2: \min_{\{p_i^l, p_i^m, p_i^s\}} \alpha T\left(p_i^l, p_i^m, p_i^s\right) + (1-\alpha)E\left(p_i^l, p_i^m, p_i^s\right), \tag{23}$$

subject to $C1 \sim C7$, where

$$\begin{aligned} T &= \sum_{i=1}^{N} \frac{1}{u_i^D\left(1-l_i^D\right) - p_i^l\lambda_i} + \sum_{i=1}^{N}\left[\frac{p_i^m\lambda_i\theta_i}{B\log_2\left(1+\left(P_i^m H_i^m/\sigma^2\right)\right)}\right] + W_i^m + \sum_{i=1}^{N}\left[\frac{p_i^s\lambda_i\theta_i}{B\log_2\left(1+\left(P_i^s H_i^s/\sigma^2\right)\right)}\right] + W_i^s, \\ E &= \sum_{i=1}^{N} \xi_i \frac{1}{u_i^D\left(1-l_i^D\right) - p_i^l\lambda_i} + \sum_{i=1}^{N}\left[P_i^m\left(\frac{p_i^m\lambda_i\theta_i}{B\log_2\left(1+\left(P_i^m H_i^m/\sigma^2\right)\right)}\right)\right] + \gamma_i W_i^m + \sum_{i=1}^{N}\left[P_i^s\left(\frac{p_i^s\lambda_i\theta_i}{B\log_2\left(1+\left(P_i^s H_i^s/\sigma^2\right)\right)}\right)\right] + \gamma_i W_i^s. \end{aligned} \tag{24}$$

5. Problem Solution through TA-QNIP Method

The interior point method is an optimization algorithm for solving the constraint problem. The basic idea is to convert the constraint optimization problem into the nonconstrained problem by introducing a penalty function method and then use the nonconstrained optimization method to iteratively solve the target value and continuously update the penalty function, then approach the optimal solution of the objective function. Since the Newton algorithm has the advantages of fast convergence, etc., when the interior point method is applied in the past work, most of the optimization iterative processes adopt the Newton method. The flow of the Newton algorithm is shown in Algorithm 1.

In this paper, the quasi-Newton method is used to solve the optimization problem, and different objective functions can be solved by different quasi-Newton methods [40]. In order to solve this nonlinear optimization problem better, we adopt quasi-Newton algorithm based on Broyden–Fletcher–Goldfarb–Shanno optimization algorithm (BFGS) with the best performance to design the TA-QNIP method. Through the interior point method, the constraint problem is transformed into an unconstrained problem at first. By adopting the BFGS quasi-Newton optimization algorithm to approximate the optimal value and using the gradient vector information, a positive definite symmetric matrix that approximates the Hessian matrix is constructed. Because it is not necessary to solve the second partial derivative of the objective function, the difficulty in the calculation is greatly reduced. Therefore, the $P2$ can be transformed into a nonconstraint problem of minimizing penalty function:

$$P3: \min_{\{p_i^l, p_i^m, p_i^s, \nabla^{(k)}\}} \phi\left(p_i^l, p_i^m, p_i^s, \nabla^{(k)}\right), \tag{25}$$

where the penalty function can be expressed as follows:

$$\begin{aligned} \phi\left(p_i^l, p_i^m, p_i^s, \nabla^{(k)}\right) &= \left[\alpha T\left(p_i^l, p_i^m, p_i^s\right) + (1-\alpha)E\left(p_i^l, p_i^m, p_i^s\right)\right] - \nabla^{(k)} ln \prod_{i=1}^{N}\left[u_i^D\left(1-l_i^D\right) - p_i^l\lambda_i\right] \\ &\quad - \nabla^{(k)} ln\left[\lambda_{max}^c - \prod_{i=1}^{N}\lambda_i\left(p_i^m + p_i^s\right)\right] - \nabla^{(k)} ln \prod_{i=1}^{N}\left(P_i^{max} - P_i^m - P_i^s\right) \\ &\quad - \nabla^{(k)} ln \prod_{i=1}^{N} p_i^l - \nabla^{(k)} ln \prod_{i=1}^{N} p_i^m - \nabla^{(k)} ln \prod_{i=1}^{N} p_i^s \\ &\quad - \nabla^{(k)} ln \prod_{i=1}^{N}\left(\lambda_{max}^m - \sum_{i=1}^{N}\lambda_i p_i^m\right) - \nabla^{(k)} ln \prod_{i=1}^{N}\left(\lambda_{max}^s - \sum_{i=1}^{N}\lambda_i p_i^s\right) \\ &\quad + \frac{1}{\sqrt{\nabla^{(k)}}} \prod_{i=1}^{N}\left[1 - p_i^l - p_i^m - p_i^s\right]^2. \end{aligned} \tag{26}$$

(1) Initial feasible point x_0, define ε as a sufficiently small positive real number, $k = 0$.
(2) Calculate $\mathbf{g}_k$ and $\mathbf{H}_k$.
(3) **If** $\|\mathbf{g}_k\| < \varepsilon$,
it should stop iterating;
else
determine the search direction $q_k = -\mathbf{H}_k^{-1} \cdot \mathbf{g}_k$
(4) Calculate next iteration points: $x_{k+1} = x_k + q_k$
(5) $k = k + 1$ and turn to step 2.
where $\mathbf{g}_k$ is the gradient vector of the objective function, $\mathbf{H}_k^{-1}$ is the inverse of the Hessian matrix, and the Newton iteration direction is $q_k = -\mathbf{H}_k^{-1} \cdot \mathbf{g}_k$. Each iteration of the Newton algorithm needs to solve the inverse of the Hessian matrix of the objective function, so that the calculation is complicated.

Algorithm 1: Newton algorithm

In the penalty function, when the arbitrary solution $((p_i^l)^0, (p_i^m)^0, (p_i^s)^0)_{i=1}^N$ approaches the constraint boundary, the function value will increase rapidly, forcing the optimal value to be solved within the feasible domain. $\nabla^{(k)} > 0$ ($k = 0, 1, 2 \ldots$) is penalty factor, it is a decreasing coefficient, and the reduction factor is set to Γ. Then, the penalty factor can be denoted as $\nabla^{(k+1)} = \Gamma\nabla^{(k)}$ ($k = 0, 1, 2 \ldots$), where $(p_i^l(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N$ is the extreme point obtained by the penalty function under the TA-QNIP method. We express $\mathbf{g}_k$ as the gradient vector of the objective function and $\mathbf{D}_k$ as the approximate matrix of the inverse of the Hessian matrix of the objective function, so that the search direction is $q_k = -\mathbf{D}_k \cdot \mathbf{g}_k$. When solving $\mathbf{D}_k$, we first need to derive the quasi-Newton conditions that the approximate matrix of the Hessian matrix needs to satisfy. Let the objective function be $f(\mathbb{P})$, $\mathbb{P}$ is the set of solutions, and then expand the Taylor series of $f(\mathbb{P})$ at $\mathbb{P} = \mathbb{P}_{k+1}$, that is:

$$\begin{aligned} f(\mathbb{P}) &= f(\mathbb{P}_{k+1}) + f'(\mathbb{P}_{k+1})(\mathbb{P} - \mathbb{P}_{k+1}) \\ &\quad + \frac{1}{2}(\mathbb{P} - \mathbb{P}_{k+1})^T f''(\mathbb{P}_{k+1})(\mathbb{P} - \mathbb{P}_{k+1}) + R_n(\mathbb{P}) \\ &\approx f(\mathbb{P}_{k+1}) + f'(\mathbb{P}_{k+1})(\mathbb{P} - \mathbb{P}_{k+1}) \\ &\quad + \frac{1}{2}(\mathbb{P} - \mathbb{P}_{k+1})^T f''(\mathbb{P}_{k+1})(\mathbb{P} - \mathbb{P}_{k+1}). \end{aligned} \tag{27}$$

Take the first-order partial derivative of $f(\mathbb{P})$:

$$f'(\mathbb{P}) \approx f'(\mathbb{P}_{k+1}) + f''(\mathbb{P}_{k+1})(\mathbb{P} - \mathbb{P}_{k+1}), \tag{28}$$

When $\mathbb{P} = \mathbb{P}_k$, we can obtain: $f'(\mathbb{P}_k) = f'(\mathbb{P}_{k+1}) + f''(\mathbb{P}_{k+1})(\mathbb{P}_k - \mathbb{P}_{k+1})$. Through conversion, we can get: $\mathbf{g}_k = \mathbf{g}_{k+1} + \mathbf{H}_{k+1}(\mathbb{P}_k - \mathbb{P}_{k+1})$, that is, $\mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{H}_{k+1}(\mathbb{P}_{k+1} - \mathbb{P}_k)$. To facilitate the definition, we set

$$\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k, \tag{29}$$

$$\mathbf{s}_k = \mathbb{P}_{k+1} - \mathbb{P}_k, \tag{30}$$

$$\mathbf{B}_{k+1} \approx \mathbf{H}_{k+1}, \tag{31}$$

$$\mathbf{D}_{k+1} \approx \mathbf{B}_{k+1}^{-1}, \tag{32}$$

where $\mathbf{B}_{k+1}$ is the approximation of the Hessian matrix and $\mathbf{D}_{k+1}$ is the approximation of the inverse matrix $\mathbf{H}_{k+1}^{-1}$ of the Hessian matrix, then

$$\mathbf{y}_k = \mathbf{B}_{k+1}\mathbf{s}_k, \tag{33}$$

$$\mathbf{s}_k = \mathbf{D}_{k+1}\mathbf{y}_k. \tag{34}$$

The above formula is the quasi-Newton condition, which constrains the approximation of the Hessian matrix in the iteration. Then, we construct an approximation matrix that satisfies the quasi-Newton condition by the BFGS method instead of the original Hessian matrix. Let the iterative formula of the approximate Hessian matrix be

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \Delta\mathbf{B}_k. \tag{35}$$

Let $\Delta\mathbf{B}_k = \alpha\mathbf{u}\mathbf{u}^T + \beta\mathbf{v}\mathbf{v}^T$, where vectors $\mathbf{u}$ and $\mathbf{v}$ are undetermined vectors, and their dimensions are $n \times 1$ (n is the dimension of $\mathbb{P}$). The variable quantity of matrix obtained by this way must be symmetric matrix; then

$$\begin{aligned} \mathbf{y}_k = \mathbf{B}_{k+1}\mathbf{s}_k &= \left(\mathbf{B}_k + \alpha\mathbf{u}\mathbf{u}^T + \beta\mathbf{v}\mathbf{v}^T\right)\mathbf{s}_k \\ &= \mathbf{B}_k\mathbf{s}_k + \left(\alpha\mathbf{u}^T\mathbf{s}_k\right)\mathbf{u} + \left(\beta\mathbf{v}^T\mathbf{s}_k\right)\mathbf{v}. \end{aligned} \tag{36}$$

Let $\alpha\mathbf{u}^T\mathbf{s}_k = 1$, $\beta\mathbf{v}^T\mathbf{s}_k = -1$, then we have $\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k = \mathbf{u} - \mathbf{v}$; let $\mathbf{u} = \mathbf{y}_k$ and $\mathbf{v} = \mathbf{B}_k\mathbf{s}_k$, we get $\alpha = (1/\mathbf{y}_k^T\mathbf{s}_k)$, $\beta = -(1/\mathbf{s}_k^T\mathbf{B}_k^T\mathbf{s}_k)$. Finally, the correction matrix is obtained:

$$\begin{aligned} \Delta\mathbf{B}_k &= \alpha\mathbf{u}\mathbf{u}^T + \beta\mathbf{v}\mathbf{v}^T \\ &= \frac{1}{\mathbf{y}_k^T\mathbf{s}_k}\mathbf{u}\mathbf{u}^T - \frac{1}{\mathbf{s}_k^T\mathbf{B}_k^T\mathbf{s}_k}\mathbf{v}\mathbf{v}^T \\ &= \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\mathbf{B}_k\mathbf{s}_k\mathbf{s}_k{}^T\mathbf{B}_k{}^T}{\mathbf{s}_k^T\mathbf{B}_k^T\mathbf{s}_k}. \end{aligned} \tag{37}$$

The above formula can be replaced by $\mathbf{B}_{k+1} = \mathbf{B}_k + (\mathbf{y}_k\mathbf{y}_k^T/\mathbf{y}_k^T\mathbf{s}_k) - (\mathbf{B}_k\mathbf{s}_k\mathbf{s}_k{}^T\mathbf{B}_k{}^T/\mathbf{s}_k^T\mathbf{B}_k^T\mathbf{s}_k)$. We introduce the identity matrix $\mathbf{I}$, by using Sherman–Morrison formula, and the above equation can be converted into $\mathbf{B}_{k+1}^{-1} = (\mathbf{I} - (\mathbf{s}_k\mathbf{y}_k^T/\mathbf{y}_k^T\mathbf{s}_k))\mathbf{B}_k^{-1}(\mathbf{I} - (\mathbf{y}_k\mathbf{s}_k^T/\mathbf{y}_k^T\mathbf{s}_k)) + (\mathbf{s}_k\mathbf{s}_k^T/\mathbf{y}_k^T\mathbf{s}_k)$, that is:

$$\mathbf{D}_{k+1} = \left(\mathbf{I} - \frac{\mathbf{s}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k}\right)\mathbf{D}_k\left(\mathbf{I} - \frac{\mathbf{y}_k\mathbf{s}_k^T}{\mathbf{y}_k^T\mathbf{s}_k}\right) + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{y}_k^T\mathbf{s}_k}. \tag{38}$$

(1) **Input:**
Initialize the feasible point $((1-p_i^m-p_i^s)^0, (p_i^m)^0, (p_i^s)^0)_{i=1}^N$, initialize the penalty coefficient $\nabla^{(0)}$, set the dropping factor Γ, $k=0$, $\mathbf{D}_0=\mathbf{I}$.
(2) Define ε_1 and ε_2 as a sufficiently small positive real number, where $\varepsilon_1 > \varepsilon_2$.
(3) Determine search direction $q_k = -\mathbf{D}_k \cdot \mathbf{g}_k$.
(4) Find the optimal step factor:
$\ell_k = \arg \min_{\ell \in \mathbf{R}} \phi(((1-p_i^m-p_i^s)(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N + \ell q_k)$
$\mathbf{s}_k = \ell_k q_k$
$((1-p_i^m-p_i^s)(\nabla^{(k+1)}), p_i^m(\nabla^{(k+1)}), p_i^s(\nabla^{(k+1)}))_{i=1}^N = ((1-p_i^m-p_i^s)(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N + \mathbf{s}_k$
(5) Iteration:
While $\|\mathbf{g}_{k+1}\| > \varepsilon_1$
do $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$
$\mathbf{D}_{k+1} = (\mathbf{I} - (\mathbf{s}_k\mathbf{y}_k^T/\mathbf{y}_k^T\mathbf{s}_k))\mathbf{D}_k(\mathbf{I} - (\mathbf{y}_k\mathbf{s}_k^T/\mathbf{y}_k^T\mathbf{s}_k)) + (\mathbf{s}_k\mathbf{s}_k^T/\mathbf{y}_k^T\mathbf{s}_k)$
$k = k+1$
Go to step 4
end while
output $((1-p_i^m-p_i^s)(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N$
(6) Set the algorithm termination condition:
while $\|((1-p_i^m-p_i^s)(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N - ((1-p_i^m-p_i^s)^0, (p_i^m)^0, (p_i^s)^0)_{i=1}^N\| > \varepsilon_2$
do
Iteration: $\nabla^{(k+1)} = \Gamma\nabla^{(k)}$ $(k=0, 1, 2, \ldots)$
$((1-p_i^m-p_i^s)^0, (p_i^m)^0, (p_i^s)^0)_{i=1}^N = ((1-p_i^m-p_i^s)(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N$, $k = k+1$
end while
(7) **Return:** $((1-p_i^m-p_i^s)(\nabla^{(k)}), p_i^m(\nabla^{(k)}), p_i^s(\nabla^{(k)}))_{i=1}^N$
(8) **Output** $((p_i^l)^*, (p_i^m)^*, (p_i^s)^*)_{i=1}^N$ is the approximate optimal solution of the objective function.

Algorithm 2: Task assignment algorithm based on quasi-Newton interior point method

Thus, the inverse of the Hessian matrix is avoided in every iteration, and the difficulty in the calculation is greatly reduced. By iterating the correction matrix many times, the optimal search direction is changed continuously, and the approximate optimal solution $((p_i^l)^*, (p_i^m)^*, (p_i^s)^*)_{i=1}^N$ is obtained. The task assignment algorithm based on quasi-Newton interior point method is shown in Algorithm 2.

*5.1. *Algorithm Complexity Analysis.* The service difference between base stations will affect the offload probability, and the number of MDs and base stations will affect the algorithm complexity. In the two comparison algorithms in this paper, they both use the interior point method to set the penalty function and transform the constraint problem into a nonconstraint problem. When the task assignment algorithm based on Newton interior point (TA-NIP) method solves the optimal value in a nonconstrained problem, in order to find the optimal search direction, the Hessian matrix of the objective function must be solved first, and the inverse of the Hessian matrix of the objective function is calculated. The calculation complexity is exponential order. However, the TA-QNIP method proposed in this paper only needs to construct an approximate matrix to represent the inverse of the Hessian matrix, thereby reducing the complexity of the algorithm. In the complexity analysis, the first-order operation of matrix is ignored and the second-order operation of matrix is considered. Let N be the number of users, the number of base stations is 2, and k is the number of iterations. Then, the complexity of the TA-NIP method is $O((2N)^3 * k)$, and the complexity of the TA-QNIP method is $O(2N * k)$.

6. Simulation Results

In this section, we evaluate the performance of the proposed TA-QNIP method through simulation results. At the same time, according to the simulation results, the advantages of the cooperative base station model are also proved. We consider that the distance d^m between MBS and MDs is 1000 m, and the distance d^s between SBS and MDs is 50 m [34]. The task generation rate of the device λ_i satisfies [0.1, 1.1] MB/s and the task size randomly generated by each device is $\theta_i = [2.5, 5]$ MB [41]. Local device execution capability is $u_i^D = 0.5$ GHz [10], and the CPU occupied proportion l_i^D of MD$_i$ is randomly selected in [0, 1]. The response power coefficient of MD$_i$ is set at $\xi_i = 0.1$, and MD$_i$'s waiting power coefficient is $\gamma_i = 0.01$ [42]. The channel bandwidth $B = 5$ MHz [4], Gaussian white noise power $\sigma^2 = -127$ dbm, and the transmission power P_i^m and P_i^s of the MD are randomly selected in [0.2, 0.3] w. In the following simulation analysis, we use "total consumption" to represent the sum of energy and delay consumption when MDs process the task under different energy and delay consumption demands. Because the total consumption reflects the cost of delay and energy consumption when MDs process the task, there is no specific unit, and it is only expressed in the simulation environment.

Figures 3 and 4 reflect the convergence of the TA-QNIP method and the TA-NIP algorithm. To facilitate the research, we discuss the convergence of the algorithm at $\rho < 0.8$

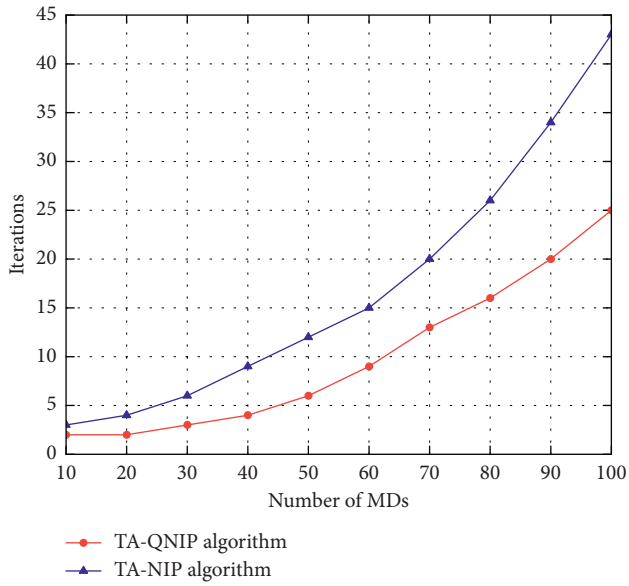

Figure 3: The influence of the number of MDs on the iterations.

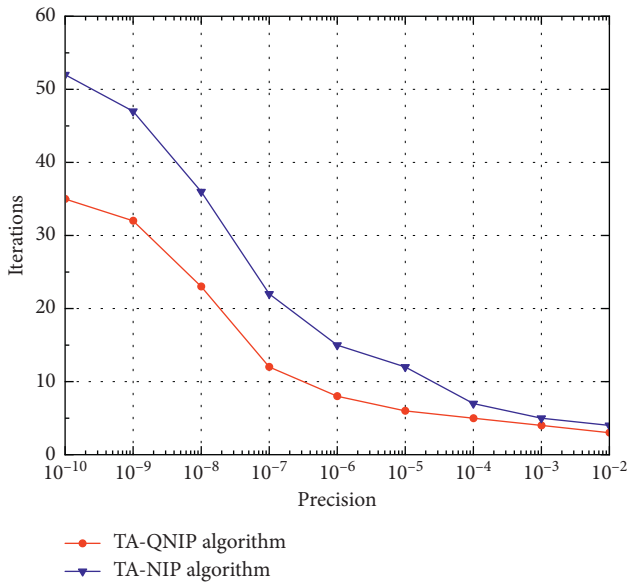

Figure 4: The effect of algorithm precision on the iterations.

and $\alpha = 0.5$. Figure 3 shows the influence of the number of MDs on the iterations; when the algorithm precision is 10^{-5}, the iterations of the two algorithms increase with the increase of the number of MDs, but the TA-QNIP algorithm shows better convergence performance, especially when the number of MDs increases significantly. Figure 4 shows the effect of algorithm precision on the iterations when $N = 50$, and it can be seen that, under different algorithm precision, the iterations of the proposed TA-QNIP algorithm are lower than those of the TA-NIP algorithm. Based on the verification of the above simulation results, the TA-QNIP algorithm has better convergence performance.

As shown in Figure 5, different schemes are applied to optimize the total consumption of MDs. In order to reflect

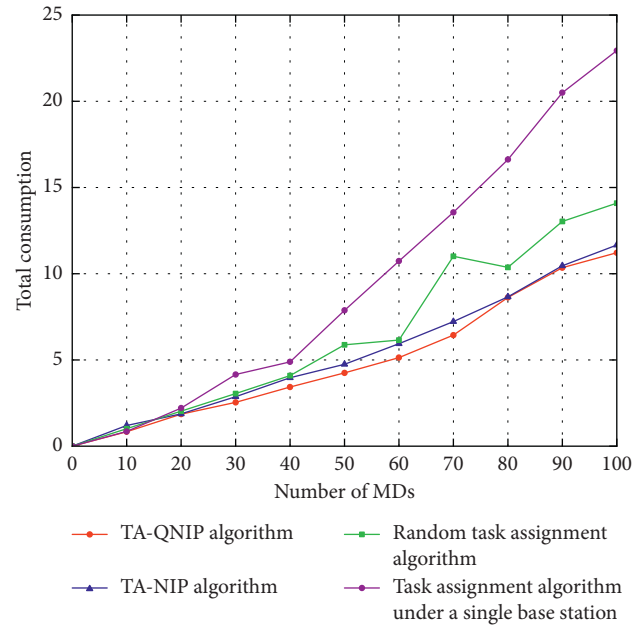

Figure 5: The impact of the number of MDs.

the advantages of the proposed scheme in this paper, based on the TA-QNIP algorithm, the TA-NIP algorithm, random task assignment algorithm [4], and task assignment algorithm under a single base station were also proposed for analysis and comparison. It is noted that as the quantity of MDs increases, the trend of MDs' total consumption will increase under different allocation algorithms. It shows that the total consumption of MDs optimized by the TA-NIP algorithm is slightly higher than that of the proposed algorithm because every step of the algorithm needs to solve the inverse matrix of the Hessian matrix of the objective function. When Hessian matrix is not positive, the correctness of the descent direction could not be guaranteed, so it could not converge at the approximate optimal solution. When using random task assignment algorithm, with the increase of MDs, the total consumption of MDs does not show stable optimization results. The reason is that random task assignment algorithm cannot produce a good allocation mechanism to ensure system performance, so the algorithm is the worst in the cooperative base station model. When applying the task assignment algorithm based on single base station, it can be seen that when the number of MDs is small, the single base station can meet the task requirements of fewer MDs at the same time, so the total consumption of MDs under the cooperative model is not much different. However, when the number of MDs increases, it is difficult for a single base station to meet the demands of multi-MD and multitask processing at the same time; therefore, compared with the cooperative model, the total consumption of MDs is more.

Figure 6 shows the effect of total consumption on different queue strength. We discuss the situation when the delay weight coefficient $\alpha = 0.5$. It can be seen that the total consumption increases as the quantity of MDs increases. When the number of tasks is small, the allowable queue length does not reach saturation, and the tasks that need to

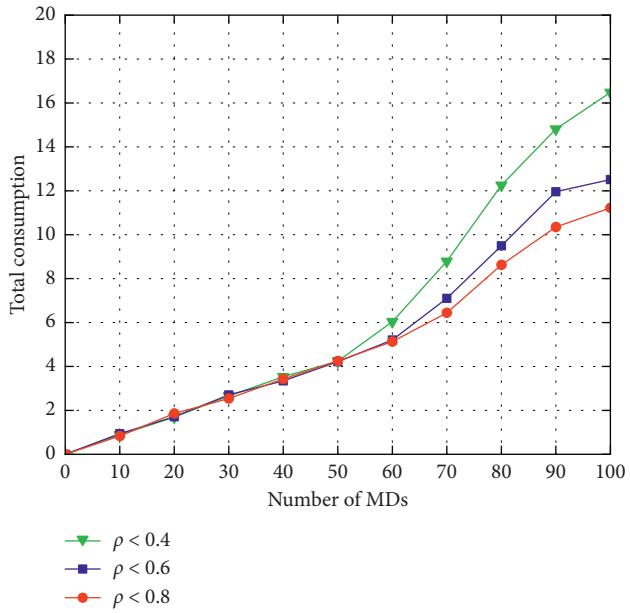

Figure 6: Effect of queue strength on the total consumption.

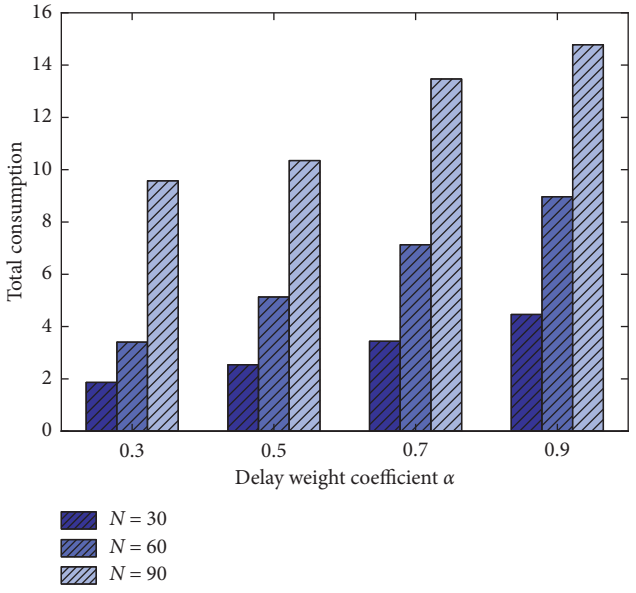

Figure 7: The influence of different weighting factors on the total consumption.

be offloaded are successfully entered into the queuing sequence. Therefore, under the constraint of different allowable queue strength, the growth trend of total consumption is almost the same. It should be noted that, with the increase of the number of MDs, when the queue strength meets the limits of $\rho < 0.4$ and $\rho < 0.6$, respectively, the growth trend of the total MDs' consumption is significantly faster than that of $\rho < 0.8$, because the allowable queue length of the offloading task is reduced, and the tasks to be offloaded cannot successfully enter the queuing sequence, resulting in additional consumption due to system congestion.

Figure 7 shows the influence of different delay weight coefficients α on the total consumption of MDs. It can be seen that when the number of MDs is constant, with the increase of delay weight and the decrease of energy consumption weight, the types of tasks that MDs need to process tend to be more sensitive to delay. Therefore, most tasks are executed on local devices, which will increase the overall cost of MDs. On the contrary, with the increase of energy consumption weight and the decrease of delay weight, it means that the target MDs pay more attention to the demand of energy consumption, so most tasks of MDs choose to be offloaded to the MEC for execution, thus reducing the total consumption of MDs.

7. Conclusion

In this paper, a base station collaborative task offloading scheme in 5G MEC networks is established. First, we use queuing theory to model the process of task processing, and then the problem of minimizing the total consumption of MDs is formulated. We establish the probability-based optimization scheme. In order to solve the objective equation effectively, we propose the TA-QNIP method with lower computational complexity. Simulation results show that compared with the TA-NIP algorithm, the proposed algorithm can accelerate the convergence speed and reduce the total consumption of MDs more effectively. When the number of MDs and the task processing demands is massive, the proposed scheme is more effective. In addition, considering user task-intensive scenarios, this work can be extended to large-scale heterogeneous network for future research to greatly improve the offloading experience of user groups.

Data Availability

The simulation data supporting the system performance analysis are from previously reported studies and datasets, which have been cited.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (grant nos. 61841107 and 61461026).

References

[1] S. Zhao, J. Wen, S. Mumtaz et al., "Spatially coupled codes via partial and recursive superposition for industrial IoT with high trustworthiness," *IEEE Transactions on Industrial Informatics*, vol. 34, 2020.

[2] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2d big data: content deliveries over wireless device-to-device sharing in realistic large scale mobile networks," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 1–10, 2018.

[3] Y. Liu, Z. Zeng, X. Liu, X. Zhu, and M. Z. A. Bhuiyan, "A novel load balancing and low response delay framework for edge-

cloud network based on SDN," *IEEE Internet of Things Journal*, vol. 1, 2019.

[4] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6398–6409, 2018.

[5] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[6] X. Liu, A. Liu, T. Wang et al., "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *Journal of Parallel and Distributed Computing*, vol. 135, pp. 140–155, 2020.

[7] T.-D. Nguyen, E.-N. Huh, and M. Jo, "Decentralized and revised content-centric networking-based service deployment and discovery platform in mobile edge computing for IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4162–4175, 2019.

[8] L. Wang, L. Jiao, J. Li, J. Gedeon, and M. Muhlhauser, "MOERA: mobility-agnostic online resource allocation for edge computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1843–1856, 2019.

[9] M. Merluzzi, P. Di Lorenzo, and S. Barbarossa, "Latency-constrained dynamic computation offloading with energy harvesting IoT devices," in *Proceedigs of the 2019 IEEE Conference on Computer Communications Workshops*, pp. 750–755, Paris, France, 2019.

[10] K. Zhang, Y. Mao, S. Leng et al., "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[11] X. Zheng, Y. Chen, M. Alam, and J. Guo, "Multi-task scheduling based on classification in mobile edge computing," *Electronics*, vol. 8, no. 9, p. 938, 2019.

[12] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[13] J. Zhang, W. Xie, F. Yang, and Q. Bi, "Mobile edge computing and field trial results for 5g low latency scenario," *China Communications*, vol. 13, no. 2, pp. 174–182, 2017.

[14] I. Ketykó, L. Kecskés, C. Nemes, and L. Farkas, "Multi-user computation offloading as multiple knapsack problem for 5g mobile edge computing," in *Proceedigs of the 2016 European Conference on Networks and Communications*, pp. 225–229, Athens, Greece, 2016.

[15] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: the role of MEC in the Internet of Things," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, 2016.

[16] R. Al-Zaidi, J. Woods, M. Al-Khalidi, K. M. Alheeti, and K. McDonald-Maier, "Next generation marine data networks in an IoT environment," in *Proceedings of the 2017 Second International Conference on Fog and Mobile Edge Computing*, pp. 50–55, Valencia, Spain, 2017.

[17] Z. Yan, W. Zhou, S. Chen, and H. Liu, "Modeling and analysis of two-tier HetNets with cognitive small cells," *IEEE Access*, vol. 5, pp. 2904–2912, 2017.

[18] C. F. Liu, M. Bennis, and H. V. Poor, *Latency and Reliability-Aware Task Offloading and Resource Allocation for Mobile Edge Computing*, IEEE Globecom Workshops (GC Wkshps), New York, NY, USA, 2017.

[19] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[20] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[21] Q. Liu, T. Han, and N. Ansari, "Joint radio and computation resource management for low latency mobile edge computing," in *Proceedings of the 2018 IEEE Global Communications Conference*, pp. 1–7, Abu Dhabi, United Arab Emirates, 2018.

[22] S. Sardellitti, M. Merluzzi, and S. Barbarossa, "Optimal association of mobile users to multi-access edge computing resources," in *Proceedigs of the 2018 IEEE International Conference on Communications Workshops*, pp. 1–6, Kansas City, MO, USA, 2018.

[23] M. Guan, B. Bai, L. Wang, S. Jin, and Z. Han, "Joint optimization for computation offloading and resource allocation in Internet of Things," in *Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Toronto, Canada, 2017.

[24] A. C. Pang, W. H. Chung, T. C. Chiu, and J. Zhang, "Latency-driven cooperative task computing in multi-user fog-radio access networks," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 615–624, Atlanta, GA, USA, 2017.

[25] L. Q. Liu, Z. Chang, X. J. Guo, and T. Ristaniemi, "Multi-objective optimization for computation offloading in mobile-edge computing," in *Proceedings of the 2017 IEEE Symposium on Computers and Communications*, pp. 832–837, Heraklion, Greece, 2017.

[26] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2018.

[27] Z. Jiang and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306–2316, 2015.

[28] L. Liu, Z. Chang, and X. Guo, "Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1869–1879, 2018.

[29] Y. Yang, Y. Ma, W. Xiang, X. Gu, and H. Zhao, "Joint optimization of energy consumption and packet scheduling for mobile edge computing in cyber-physical networks," *IEEE Access*, vol. 6, pp. 15576–15586, 2018.

[30] K. Li, "Computation offloading strategy optimization with multiple heterogeneous servers in mobile edge computing," *IEEE Transactions on Sustainable Computing*, vol. 1, 2019.

[31] X. Guo, L. Liu, Z. Chang, and T. Ristaniemi, "Joint optimization of energy and delay for computation offloading in cloudlet-assisted mobile cloud computing," *Wireless Networks*, vol. 25, no. 4, pp. 2027–2040, 2019.

[32] J. P. Zhou, *Communication Networks Theory*, The People's Posts and Telecommunications Press, China, Beijing, 2nd edition, 2009.

[33] S. M. Ross, *Introduction to Probability Models*, pp. 481–558, Academic Press, Boston, MA, USA, 11th edition, 2014.

[34] H. Zhang, J. Guo, L. Yang et al., "Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC," in *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 115–120, Atlanta, GA, USA, 2017.

[35] H. Liao, Z. Zhou, X. Zhao et al., "Learning-based context-aware resource allocation for edge computing-empowered industrial IoT," *IEEE Internet of Things Journal*, vol. 34, 2019.

[36] M. B. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power-control scheme in wireless cellular systems," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 210–221, 2003.

[37] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11365–11373, 2018.

[38] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[39] M. Amoretti, L. Consolini, A. Grazioli et al., "Impact of different auto-scaling strategies on adaptive Mobile Cloud Computing systems," in *Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 589–596, Messina, Italy, 2016.

[40] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Mathematics of Computation*, vol. 35, no. 151, p. 773, 1980.

[41] W. Fan, Y. Liu, B. Tang et al., "Computation offloading based on cooperations of mobile edge computing-enabled base stations," *IEEE Access*, vol. 6, pp. 22622–22633, 2017.

[42] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2018.