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](mailto:wang_zesen@hotmail.com)

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

Academic Editor: Raul Montoliu

Copyright © 2020 Jianbin Xue et al. is an open access article distributed under the Creative [Commons Attribution License,](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an indispensable key technology in 5G Internet of (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. 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. 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

(IoT) in 5G era [[1],](#page10) mobile applications such as natural language processing, virtual reality, and interactive games have greatly enriched our lives [[2].](#page10) However, mobile de-vices (MDs) with constrained computing power and bat-tery capacity could not handle the huge amount of data generated by mobile applications [[3,](#page10) [4].](#page11) To avoid this mismatch of resources, researchers have come up with various cloud-based solutions [[5].](#page11) By utilizing the abun-dant 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])](#page11) and shortening the com-puting delay [[7].](#page11) 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].](#page11)

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

1. 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].](#page11) greatly improves the oﬄoading eﬃciency of the device, while re-ducing the energy consumption of the device and shortening the backhaul delay [[11].](#page11)

In recent years, with the progress of 5G communication, MEC based on 5G architecture has been studied by many scholars [[12–16].](#page11) In the 5G network, a heterogeneous net-work composed of a macro base station and a small base station is a common form of 5G architecture [[17].](#page11) 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. our goal is to improve the oﬄoading eﬃciency of

MEC in 5G environments by considering the performance diﬀerences of the base stations.

2

MD has the oﬄoading decision right. 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. compared with the computation oﬄoading optimization scheme of the deter-ministic task model, the design of MEC systems with ran-dom 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. 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. contributions and innovations of this paper are as follows:

1. In this paper, a base station cooperative task oﬀ-loading scheme based on 5G heterogeneous net-works 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.
2. optimization goal of the MDs-centered energy and delay consumption minimization is established.

the joint optimization of the total con-sumption of MDs with diﬀerent demands on delay and energy consumption is accomplished by allo-cating the task assignment probability.

1. 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 con-vergence performance is verified.

rest of this paper is arranged as follows: we sum-marize the related work in Section [2.](#page2) Section [3](#page3) establishes a complete system model. In Section [4,](#page5) we formulate the optimization problem of minimizing the energy and delay consumption of MDs. In Section [5,](#page6) 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.](#page8) Finally, we summarize the work of the full text in Section [7.](#page10)

**2. Related Work**

At present, there are some related works focusing on MEC under 5G architecture. Wang et al. [[12]](#page11) improved system revenue by jointly optimizing computation oﬄoading,

Mobile Information Systems

resource allocation, and content caching. Zhang et al. studied how to decrease the computation oﬄoading delay of the MEC system in 5G architecture [[13].](#page11) In order to meet the key requirements of 5G networks for low latency and high reliability, the authors in [[18, 19]](#page11) proposed a joint opti-mization 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 net-works, Zhang et al. [[10]](#page11) proposed the MEC energy-eﬃcient computing oﬄoad mechanism in 5G heterogeneous net-works, which eﬀectively reduced energy consumption through joint optimization of oﬄoading strategies and cellular network resource allocation. Considering the con-straints of computing ability and service delay requirements, Yang et al. [[4]](#page11) developed an energy optimization scheme based on artificial fish swarm algorithm to minimize the entire energy consumption of the system. 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]](#page11) has shown that there are two types of computation oﬄoading: binary oﬄoading and partial oﬀ-loading. Computing tasks cannot be divided into subtasks in binary oﬄoading. entire task must be executed on the local or MEC servers [[21, 22],](#page11) 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].](#page11) In view of task separa-bility, Guan et al. [[23]](#page11) designed an eﬃcient task oﬄoading scheme for IoT based on cooperative communication in the mobile cloud computing system. Pang et al. [[24]](#page11) 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],](#page11) the energy consumption, exe-cution delay, and price cost of the oﬄoading process in the MEC system are studied in depth by using queuing theory. authors in [27] based on Lyapunov optimization de-veloped an online algorithm and the theoretical boundary of the algorithm in terms of average power consumption and average queue length was proved. In [[28],](#page11) 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]](#page11) 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]](#page11) established a queuing model for one MD and multiple heterogeneous edge servers, and in the past work, the heterogeneity of the edge server was intro-duced for the first time to study the optimization of the computational oﬄoad strategy. authors in [31]

Mobile Information Systems

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. How-ever, 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. queuing theory, we comprehensively considered

the diﬀerences of heterogeneous base stations and estab-lished the waiting consumption and transmission con-sumption 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 mini-mize 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 pro-cessing 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,](#page4) we consider a set of MDs in the system, which is denoted by MD*i* (*i* � 1, 2, 3 . . . *N*), a macro base station (MBS) equipped with MEC servers and a small base station (SBS). 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 pro-cessed 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 re-ducing the processing pressure of MBS. Based on the queuing theory [[32],](#page11) 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](#page4) shows the task queuing process. Suppose the task generation rate of the MD*i* is *λi* (measured by the number of generation tasks on per unit of time, e.g., second), the size of request data is *θi*. probability that the task generated by the MD *i* is locally executed is *pli*, the probability that the task is processed by the edge cloud is *pci*, and *pmi* and *psi* 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 *pci* � *psi* + *pmi*. Due to the nature of Poisson distribution, we assume that the service

3

request oﬄoaded to the MEC servers follows the Poisson process with an average rate of *pciλi*, and the locally pro-cessed service request follows the Poisson process with an average rate of *pliλi*.

*3.2. Local Model.* consumption of MDs performingtasks locally is divided into two parts: computation and task response consumption. In order to simplification, we only consider the task response consumption. *uDi* represents the execution capability of MD*i*, and *lDi* represents the pro-portion of CPU occupied by MD*i*. Since the generation of tasks is distributed negatively exponentially, the task pro-cessing model is considered to be M/M/1 queue on the MD side. By Little’s Law [[33],](#page11) the local task response time is *T* �(1/*u*)/(1− *η*), and the queuing eﬃciency is *η* � *λ*/*u*,where *λ* and *u* are task arrival rate and device service rate, respectively. the average response time and en-

ergy consumption of locally executed tasks are as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *TiD* � | |  |  | 1 | |  | *,* |  | ( | 1 | ) |  |
|  | *uiD* 1− *liD* − *pil* | | |  |  |  |
|  |  |  | *λi* | |  |  |  |  |
| *D* |  |  | *D* |  |  | 1 | |  | (2) | | |  |
| *Ei* | � | *ξiTi* | | � *ξi* |  | | | *,* |  |
| *uiD* 1− *liD* − *pilλi* | | |  |

where *ξi* represents the response energy consumption co-eﬃcient of MD*i*.

*3.3. Transmission Model.* In the edge heterogeneous net-work, in addition to MBS, the SBS is regarded as the co-operative 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].](#page11) 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 as-signment 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].](#page11)

fore, we can calculate the uplink transmission rate of the

MD*i* oﬄoading tasks to the MBS:

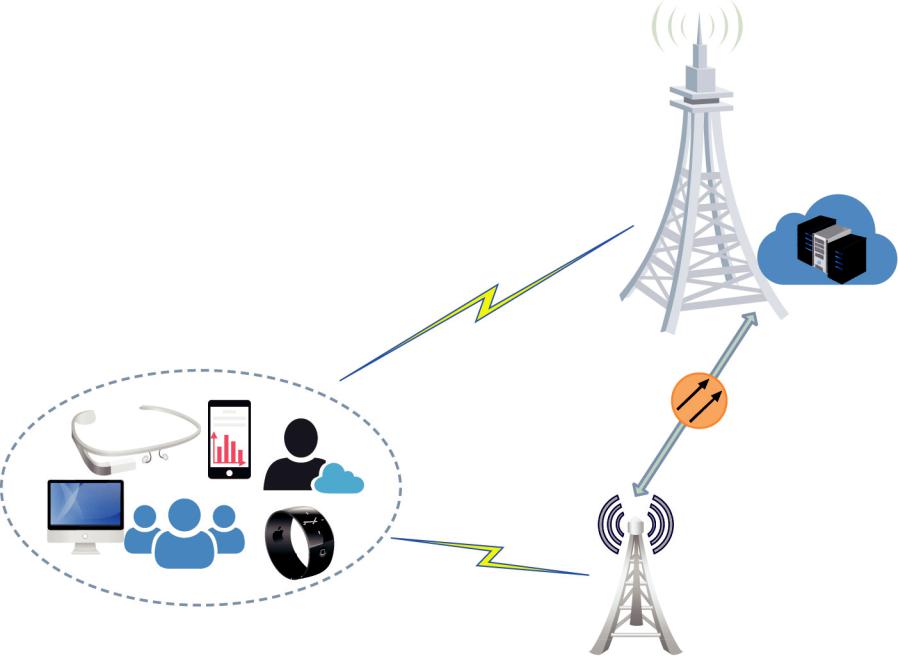
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Rim* | � *B*log21 | *PmHm* | (3) |  |
| + *i* *σ*2 *i* !*.* |  |

Similarly, the uplink transmission rate of the MD*i* oﬀ-loading tasks to the SBS is given by

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *s* | � *B*log2 | 1 + | *PsHs* | |  | (4) |  |
| *i* | *i* | !*,* |  |
| *Ri* |  |  |  |
| *σ*2 |  |  |

where *σ*2 is the Gaussian white noise power and *Pmi* and *Psi* denote the transmission power of MDs to MBS and SBS, respectively. transmission power can be determined by the power control mechanism of MBS and SBS [[36].](#page12) In addition, *P*max*i* is the maximum transmission power of the MD*i*, *Hi*is the channel gain between the MD*i* and base stations, *Hi* � 127 + 30 × log*di*, and *di* is the distance be-tween the MD*i* and base stations [[37].](#page12)

4 Mobile Information Systems



MEC server

Macro BS

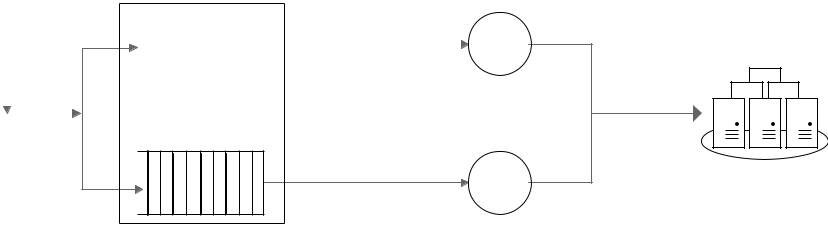
Backhaul

Mobile devices

Small BS

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *λi* | *pilλi* | | |  |  |  |  |  | M/M/1 | | | | | | | | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Mobile | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | device | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | Queue for local | | | | | | | | | | | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  | execution | | | | | | | | | | | | | | |  |  |  |
| *pciλi* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MBS |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | *Ri*m | *λp*m | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | M/M/*c* | | | | | | | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



Process

|  |  |
| --- | --- |
| SBS | MEC server |

*R*s*i* *λ*s*p*

Queue for

offload execution

FIGURE 2: Queuing model of task request processing.

For MBS and SBS, the maximum acceptable task arrival

rate is *λmmax* and *λsmax*, respectively, and the sum of all task request rates from diﬀerent MDs is expressed as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *N* |  |  |  | 5 |  |  |
| *m* |  | X | *pm,* | | ( | ) |  |
| *λtotal* | � | *i*�1 *λi* | *i* |  |  |  |  |
|  |  | *N* |  |  |  | 6 |  |  |
| *λtotals* | � X *λips.* | | | | ( | ) |  |
| the actual task arrival*i*�1rate *i* | | | | on the MBS is |  |  |  |

*λmp* �min[*λmtotal , λmmax*], and the actual task arrival rate on theSBS is *λsp* � min[*λstotal,* *λsmax*]. Assume that the service rate of MBS is *um* and the service rate of SBS is *us*. According to the

M/M/*c* queuing model definition, the queue strengths of the tasks to the MBS and the SBS are as follows:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | *λm* | | |  | 7 |  |  |
| *ρ* | *m* | |  | *p* | |  | ( | ) |  |
|  |  |  | *λs* | | |  |
|  |  |  |  |  |  |
|  |  | � *cum,* | | | | (8) | | |  |
| *ρ* | | *s* |  | *p* | | |  |
|  | � |  | *.* | |  |
|  | *cus* |  |

Queue strength is a parameter to measure the stability of the system. When *ρm* <1 and *ρs* <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. total transmission time in-cludes the transmission time of the uplink and the waiting time

Mobile Information Systems

for the task to be oﬄoaded. when the MD *i* oﬄoads

its tasks to the MEC servers through MBS, the transmission delay and energy consumption can be calculated as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *m* | *m* |  | *pimλi* | | | | *θi* |  | *m* | | |  | *pimλiθi* | | | *m* | |  |  |  |
| *Ti* | *Pi* | � |  |  | |  |  | + *Wi* � | |  |  |  | |  |  | + *Wi* | | *,* |  |  |
|  | *Rim* | | |  | *B*log2 | | 1 + *PimHim*/*σ*2 | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ( | | 9 | ) |  |
| *mm* | |  | *m* | | *m* | | | *m* | *mpimλi* | | | *θi* |  | *m* | |  |  |  |  |
| *Ei* | *Pi* | � *Pi* | |  | *Ti* | |  | *Pi* | � *Pi* | |  |  | + *ciWi* | | |  | 10 | |  |  |
|  |  | *Rm* |  |  |  |  |
|  |  | � *Pi* | |  | *B*log2 | | | |  |  | *i* |  |  |  |  | ( | ) |  |
|  |  |  | 1+ *PimHim*/*σ*2!+ *ciWi ,* | | | | | | |  |
|  |  |  |  |  |  |  |  |  | *pimλiθi* | | |  |  |  |  |  |  |  |
|  |  |  | *m* | |  |  |  |  |  |  | *m* | |  |  |  |  |  |

where *ci*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 calcu-lated as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *T* | | *s* | *s* |  | *pisλi* | | | | *θi* |  |  | *s* |  |  |  | *pisλiθi* | | | | *s* |  | 11 |  |  |
|  | *P* | � |  |  |  |  |  | + *W* | |  | � | |  |  |  |  |  | + *W* | *,* |  |  |
|  |  |  | *Ris* | |  |  | *B*log21+ *PisHis*/*σ*2 | | | | |  |  |
|  | *s* | *i* | *i* |  |  |  |  | *s* |  | *i* |  |  | *i* |  |  |  |
|  |  | *s* |  | *s s* | | | |  |  |  | *spisλiθi* | | |  |  |  | *s* |  | ( | ) |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *E* |  |  | *P* | � *P* |  | *T P* | | | | | � *P* | |  |  |  | + *c* | *W* | |  |  |  | 12 |  |  |
| *i* |  | *i* | *i Rs* | | | *i* |  |  |  |  |
|  |  | *i* |  |  | *i* | |  | *i* |  |  | *i* |  |  |  |  |  |  |
|  |  |  |  | � *Pi* | |  | *B*log2 | | | |  |  |  |  | *i* |  |  |  |  |  | ( | ) |  |
|  |  |  |  |  | 1+ *PisHis*/*σ*2!+ *ciWi ,* | | | | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  | *pisλiθi* | | | | |  |  |  |  |  |  |  |
|  |  |  |  |  | *s* | |  |  |  |  |  |  |  | *s* |  |  |  |  |  |

where *Wmi* 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:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *Lqm* | *cρm c*−1*ρm*/*c*!1− *ρm* | | 2 *pm* | | (13) |  |
| *Wim* | � *um* � |  |  | 0 | *,* |  |
|  | *um* |  |  |



where the idle probability of MBS is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *c*−1 | *m* | *k* |  | *m c* | | − 1 |  | 14 |  |  |
| *m* | � | 24X | *cρ* | |  | *cρ* | 35 | *.* | ( | ) |  |
| *p*0 |  |  |  |  |  |
| *k*�0 | *k*! |  | + *c*!1− *ρm* | | |  |  |  |

Similarly, the waiting time for the task generated by MD*i* to be oﬄoaded to SBS is given by

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *s* | *Lqs* | |  |  | *cρs* | *c*−1*ρs*/*c*!1− *ρs* | | | | | 2 *ps* |  | ( ) | | |  |
|  |  |  |  |  |  |  |  |  |  | 0 |  |  |
| *Wi* � | *us* | | � |  |  |  |  | *us* |  |  |  | *,* |  | 15 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| where the idle probability of SBS is as follows: | | | | | | | | | | | |  |  |  |  |
|  |  |  | *c*−1 | | *s* | *k* |  | *s* | *c* | | − 1 |  |  | 16 |  |  |
| *s* | � | 24X | | | *cρ* |  |  | *cρ* |  | 35 | *,* |  | ( | ) |  |
| *p*0 |  |  |  |  |  |  |  |
|  | *k*�0 | | *k*! |  | + *c*!1− *ρs* | | | |  |  |  |  |



where *Lmq* and *Lsq* 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].](#page11)

*3.4. EdgeCloud Model.* After receiving the oﬄoaded task, theMEC server performs the calculation immediately. maximum workload of the MEC system is limited to the maximum receiving rate, which is expressed as *λcmax*. In the system, the total request rate from diﬀerent MDs is expressed as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | 5 | |  |
|  | *N* |  |  | *N* |  |  |  |  | 17 |  |  |
| *D* | � X *pc* | | | � Xh *mpm* | | + | *s ps*i*.* | ( | ) |  |
| *λtotal* | *i*�1 | *λi* | *i* | *i*�1 | *λp i* |  | *λp i* |  |  |  |

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 re-ceive and process the result, which is similar to [[38].](#page12)

**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 con-sumption of MDs based on multi-base station cooperation. Similar to the work in reference [[39],](#page12) the total delay con-sumption of the user’s task processing can be obtained:

|  |  |
| --- | --- |
| *Ti pil, pim, pis* � *TiD pil* + *Tim pim* + *Tis pis ,* | (18) |

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

|  |  |
| --- | --- |
| *Ei pil, pim, pis* � *EiD pil* + *Eim pim* + *Eis pis .* | (19) |

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | *N* |  |  |  |  |  |  |  |  | 20 |  |  |
| *T* | *pl* | *, pm* | *, ps* | � | X *T* | |  | *pl* | *, pm* | *, ps* |  | *,* | ( | ) |  |
|  | *i* | *i* | *i* | *i*�1 |  | *i* | *i* | *i* | *i* |  | |  |  |  |
|  |  |  |  |  | *N* |  |  |  |  |  |  |  |  |  |  |  |
|  | *l* | *m* | *s* |  | X |  |  | *l* | *m* | *s* | | *.* | ( | | ) |  |
| *E pi* | | *, pi* | *, pi* | � | *i*�1 | *Ei pi* | | | *, pi* | *, pi* |  |  |  | 21 |  |  |

Since this paper considers the multiobjective optimi-zation of MDs’ energy and delay consumption, the trans-mission 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. the objective function and the restriction conditions are as follows:

*P*1: *min* n*T pli, pmi, psi* *, E pli, pmi, psi* o*,*

{*pli* *,pmi* *,psi* }

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *C*1: *pilλi* | | |  | *uiD* 1− *liD*(*i* � 1*,* 2*,* 3 . . . *N*)*,* | | | | | | | | | | | | | |  |  |  |  |
| *C*2: *λtotalD*< | | | | | *λmaxc* | | | *,* |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | *m* |  |  | *s* |  | *max* | |  |  |  |  | 1*,*2*,*3...*N*)*,* | | |  |  |  |  |
| *C*3:0 *P*<*i* | | | | | + *Pi* | | | *Pi* | |  | (*i* � | | | |  |  |  |  |
|  | *s* |  |  | *m* | |  |  | *l* | 1 (*i* � 1*,* | | | | | | 2*,*3...*N*)*,* | | |  |  |  |  |
| *C*4: *pi*≤+ *pi* | | | | | | + *pi*≤� | | | ( | 22 | ) |  |
| *C*5: *pil* | |  | 0*, pim* | | | |  | 0*, pis* | | 0 (*i* � 1*,* 2*,* 3 . . . *N*)*,* | | | | | | | |  |  |  |
|  | *N* |  |  |  |  |  | > |  |  | >(*i* � | |  |  |  |  |  |  |  |  |  |  |
| *C* | X> | |  | *pm* | |  | *m* |  |  | *,* |  | *,* |  | ...*N*)*.* |  |  |  |  |
|  | 6: *i*�1 | *λi* | |  | *i* | <*λmax* | | |  |  |  | 1 |  | 2 |  | 3 |  |  |  |  |  |
|  | *N* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

C7: X *λipsi* <*λsmax*(i � 1*,* 2*,* 3 . . . *N*)*.*

*i*�1

Here, *C*1 indicates that the local arrival rate of the task should be less than the remaining execution capacity of the

6

local device. *C*2 ensures that the total task arrival rate oﬀ-loaded 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 *C*3. probability of oﬄoading in diﬀerent ways for diﬀerent tasks should meet *C*4 and *C*5. *C*6 and *C*7 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



Mobile Information Systems

diﬀerent demands of MD*i* in various application environ-ments, we introduce the delay weight factor *α*, and then the energy consumption weight factor is (1 − *α*), where 0 ≤*α* ≤1, so *P*1 can be transformed into the following form:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *P*2: | *min* | *αT p il, pim, pis* +(1− *α*)*E pil, pim, pis ,* | ( | 23 | ) |  |
| subject | {to*i* *Ci*1∼*i* | *C*7, where |  |  |  |
|  | *pl ,pm ,ps* } | |  |  |  |  |

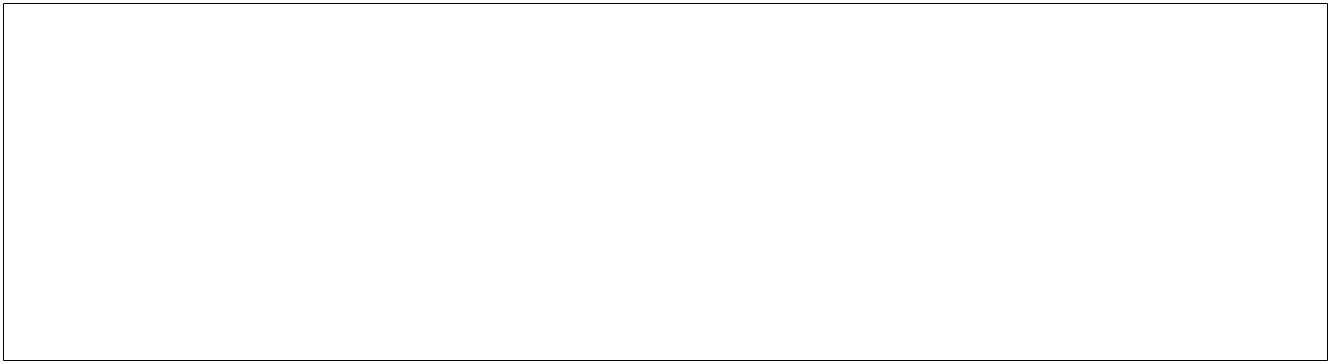
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *N* | |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  | *m* | | *λi* | *θi* | |  |  |  |  | *N* |  |  | *s* | *λiθi* |  |  |  |  |  |  |  |  |
| *T* �X | |  |  |  |  |  | + X *N* | | |  | " |  |  |  | *pi* |  | # + *Wm* + " | | | | |  |  | *pi* |  | # + *Ws,* | | |  |  |  |  |
| *i*�1 | | *uiD* | | 1 − | *liD* − | *pilλi* | |  |  | *i*�1 | |  | *B*log21+ | | | |  | *PimHim*/*σ*2 | | |  |  |  | *i* | *i*�1 | *B*log2 | | 1 + | *PisHis*/*σ*2 |  | | *i* | |  | ( | ) |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *N* | |  |  |  | 1 |  |  |  |  | *N* |  |  |  |  |  |  |  | *m* | | *λiθi* |  |  |  |  |  |  | *N* |  | *s* | *λiθi* | |  |  |  |  | 24 |  |
| *E* � | |  |  |  |  |  |  | + |  | *Pm* | | |  |  |  | *pi* | |  |  |  |  | + *c* | *Wm* | + |  | *Ps* | *pi* |  |  | + *c* | *Ws.* |  |
|  |  |  |  |  |  |  |  |  |  | 1 + | | |  | *PmHm*/*σ*2!# | | | |  | *B*log 1+ |  | *PsHs*/*σ*2!# | | |  |
| *i*� | *ξiuD* | | | 1 | − *lD* − | *pl* | |  |  | X" |  | *i* | *B*log | | 2 |  | *i* | *i* |  | X" | *i* |  | *i* | *i* |  |
|  | 1 |  | *i* | *i* | *iλi* | |  |  | *i*�1 |  |  |  |  |  |  |  |  | *ii* |  |  |  |  |  |  | *i*�1 |  | 2 |  | *i* | *i* | |  |  |  |  |
| **5. Problem Solution through TA-QNIP Method** | | | | | | | | | | | | | | | | | | | | |  | we adopt quasi-Newton algorithm based on Broyden– | | | | | | | | | | | | | | |  |
| interior point method is an optimization algorithm for | | | | | | | | | | | | | | | | | | | | |  | Fletcher–Goldfarb–Shanno optimization algorithm (BFGS) | | | | | | | | | | | | | | |  |
|  | with the best performance to design the TA-QNIP method. | | | | | | | | | | | | | | |  |
| solving the constraint problem. basic idea is to convert | | | | | | | | | | | | | | | | | | | | |  |  | the interior point method, the constraint problem | | | | | | | | | | | | |  |  |
| the constraint optimization problem into the noncon- | | | | | | | | | | | | | | | | | | | | |  | is transformed into an unconstrained problem at first. By | | | | | | | | | | | | | | |  |
| strained problem by introducing a penalty function method | | | | | | | | | | | | | | | | | | | | |  | adopting the BFGS quasi-Newton optimization algorithm to | | | | | | | | | | | | | | |  |
| and then use the nonconstrained optimization method to | | | | | | | | | | | | | | | | | | | | |  | approximate the optimal value and using the gradient vector | | | | | | | | | | | | | | |  |
| iteratively solve the target value and continuously update the | | | | | | | | | | | | | | | | | | | | |  | information, a positive definite symmetric matrix that ap- | | | | | | | | | | | | | | |  |
| penalty function, then approach the optimal solution of the | | | | | | | | | | | | | | | | | | | | |  | proximates the Hessian matrix is constructed. Because it is | | | | | | | | | | | | | | |  |
| objective function. Since the Newton algorithm has the | | | | | | | | | | | | | | | | | | | | |  | not necessary to solve the second partial derivative of the | | | | | | | | | | | | | | |  |
| advantages of fast convergence, etc., when the interior point | | | | | | | | | | | | | | | | | | | | |  | objective function, the diﬃculty in the calculation is greatly | | | | | | | | | | | | | | |  |
| method is applied in the past work, most of the optimization | | | | | | | | | | | | | | | | | | | | |  | reduced. | | | | the | |  | *P*2 can be transformed into a | | | | | | | |  |
| iterative processes adopt the Newton method. | | | | | | | | | | | | | | | | |  | flow of | | |  | nonconstraint problem of minimizing penalty function: | | | | | | | | | | | | | | |  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| the Newton algorithm is shown in Algorithm [1.](#page7) | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | *P*3: | | | *min* | | |  |  | *ϕ pil, pim, pis,* | | | | ∇ | | (*k*) *,* | ( ) | | |  |
| In this paper, the quasi-Newton method is used to solve | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  | *l* | *m* | *s* | *,*∇ | (*k*) |  |  |  |  |  |  |  |
| the optimization problem, and diﬀerent objective functions | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  | {*pi* | *,pi ,pi* | |  | } |  |  |  |  |  |  |  |  |  |
| can be solved by diﬀerent quasi-Newton methods [[40].](#page12) In | | | | | | | | | | | | | | | | | | | | | | |  | where the penalty function can be expressed as follows:25 | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |
| order to solve this nonlinear optimization problem better, | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | ∇ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | *N* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | *pl* | *, pm* | *, ps* | *,* | (*k*) |  | � |  | *T* |  | *pl, pm, ps* | | | | |  | +( |  | − |  |  | )*E* |  | *pl, pm* | | | *, ps* | | |  | − | |  | (*k*)*ln* Yh*uD* | | | | |  |  | − *lD* | | − *pl* |  | i | |  |  |  |  |
| *ϕ* | *i* | *i* | *i* |  |  |  |  | h*α* |  |  |  | *i* | *i* | | *i* |  |  | 1 |  |  | *α* |  |  | *i* |  | *i* |  |  | *i* | i |  |  | ∇ | | *i*�1 | |  | *i* |  | 1 |  | *i* | *iλi* | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | *N* | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | *N* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | − |  | (*k*) | |  | 24 | *c* |  | − | Y | |  |  |  | *m* |  |  | *s* | 35 | − | |  | (*k*) | |  |  | Y *Pmax* | | | − *Pm* − | | | | | *Ps* |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | ∇ | |  | *ln* |  | *λmax* | |  | *i*�1 | | *λi* |  | *pi* | | + *pi* | | | ∇ | |  | ∇ | |  | *ln* | | *i*�1 | *i* |  |  |  | *i* |  |  | *i* |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | *N* | |  |  |  |  |  |  |  | *N* |  |  |  |  |  |  | *N* | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | (*k*) | |  | Y *pl* | | | − | (*k*)*ln* Y *pm* | | | | | | | | − | (*k*)*ln* Y *ps* | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | − ∇ | | |  | *ln* | *i*�1 | | *i* |  | ∇ |  |  |  | *i*�1 | |  | *i* |  |  |  |  | *i*�1 | | | *i* |  |  |  |  |  |  |  |  |  |  |  |  | ( | 26 | ) |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | *N* | |  |  |  |  | *N* | | |  |  |  |  |  |  |  |  |  |  |  | *N* |  |  |  | *N* | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | − | ∇(*k*)*ln* Y0@*λmaxm* − | | | | | | | | | | X *λipim*1A − | | | | | | |  | ∇(*k*)*ln* Y0@*λmaxs* | | | | | | | | | − | X *λipis*1A | | | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | *i*�1 | |  |  |  |  | *i*�1 | | |  |  |  |  |  |  |  |  |  |  |  | *i*�1 | |  |  | *i*�1 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | 1 | |  |  | *N* |  |  |  |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | Y h1 | | | |  | *l* |  |  | *m* | |  | *s* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | + | ~~p���~~ | | | | − *pi* | | − | *pi* − | | |  | *pi* i *.* | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | ∇(*k*) | | | *i*�1 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



|  |  |
| --- | --- |
| Mobile Information Systems | 7 |



1. Initial feasible point *x*0, define *ε* as a suﬃciently small positive real number, *k* � 0.
2. Calculate **g***k* and **H***k*.
3. **If** ‖**g***k*‖<*ε*,

it should stop iterating;

**else**

determine the search direction *qk* � − **H**−*k*1 · **g***k*

1. Calculate next iteration points: *xk*+1 � *xk* + *qk*
2. *k* � *k* +1 and turn to step 2.

where **g***k* is the gradient vector of the objective function, **H**−*k* 1 is the inverse of the Hessian matrix, and the Newton iteration direction is *qk* � − **H**−*k* 1 · **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 ((*pli*)0*,* (*pmi*)0*,* (*psi*) 0)*Ni*�1 approaches the constraint boundary, the function value will increase rapidly, forcing the optimal value to be solved within the feasible domain. ∇(*k*) >0 (*k* � 0, 1, 2 . . .) is penalty factor, it is a decreasing coeﬃcient, and the

reduction factor is set to Γ. the penalty factor can be denoted as ∇(*k*+1) � Γ∇(*k*) (*k* � 0, 1, 2 . . .), where (*pli*(∇(*k*))*, pmi*(∇(*k*))*, psi*(∇(*k*)))*Ni*�1 is the extreme point ob-tained by the penalty function under the TA-QNIP method.

We express **g***k* as the gradient vector of the objective function and **D***k* as the approximate matrix of the inverse of the Hessian matrix of the objective function, so that the search direction is *qk* � − **D***k* · **g***k*. When solving **D***k*, we first need to derive the quasi-Newton conditions that the approximate matrix of the Hessian matrix needs to satisfy. Let the ob-jective function be *f*(P), P is the set of solutions, and then expand the Taylor series of *f*(P) at P � P*k*+1, that is:

*f*(P)� *f* P*k*+1 + *f*′ P*k*+1 P − P*k*+1

+ 12 P − P*k*+1 *Tf*″ P*k*+1 P − P*k*+1 + *Rn*(P)



≈*f* P*k*+1 + *f*′ P*k*+1 P − P*k*+1

+ 12 P − P*k*+1 *Tf*″ P*k*+1 P − P*k*+1 *.*



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Take the first-order partial derivative of *f*(P): | | | | | | | | | | | | | | | | | | |  |  | (27) | | | |  |
|  |  |  |  |  | *f* (P) | | | | ≈ | *f* | | P*k*+1 | | + *f* P*k*+1 | | | | P − P*k*+1 *,* | | ′ |  |  | 128 | | |  |  |
|  |  |  |  |  |  |  |  | ( |  |  | ) |  |
|  |  | � | *k*+1+ | | | |  | *k* |  | (*k*+1−. | | | |  | ) |  | ″obtain: | ′ |  |  | − |  |  | � | |  |
|  |  | When′ | | | |  | P | | � P′*k*, | | |  | we | can | | | *f* (P*k*)� *f* | | | (P*k*+ )+ | | | | | |  |
| *f* (P | | | |  | )(P−P | | | | | | |  | ) | conversion, | | | | we can | get: | |  |  |  |  |  |  |  |
| **g** | *k*″ |  | **g** | | *k*+1 |  | **H** | | *k*+1) | | P | *k* | P | *k*+1 |  | , | that | is, | **g** | *k*+1 | |  | **g** | *k* |  |  |  |
|  |  | ( |  | − | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **H***k*+1 | | |  | P*k*+1 | |  |  | P*k* | | | . To facilitate the definition, we set | | | | | | | | | | |  |  |  |  |  |  |

where **B***k*+1 is the approximation of the Hessian matrix and **D***k*+1is the approximation of the inverse matrix **H**−*k*+11of theHessian matrix, then

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **s***k*� **D***k*+ | 1 | **y***k.* | ( | 33) |
| **y***k*� **B***k*+ | | **s***k,* | ( | ) |
|  | 1 |  |  | 34 |

above formula is the quasi-Newton condition, which constrains the approximation of the Hessian matrix in the iteration. 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

|  |  |
| --- | --- |
| **B***k*+1� **B***k*+ **B***k.* | (35) |

Let **B***k* � *α***uu***T* + *β***vv***T*, where vectors **u** and **v** are undetermined vectors, and their dimensions are *n* × 1 (*n* is the dimension of P ). variable quantity of matrix ob-tained by this way must be symmetric matrix; then

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **y***k*� **B***k*+1**s***k*� **B***k*+*α***uu***T*+*β***vv***T* **s***k* | | | | ( | 36 | ) |  |
| *T* �**B***k***s***k* + *α***u***T* **s***k* **u**+ *β***v** | |  | **s***k* **v***.* |  |
|  | *T* | *T* |  |  |  |
|  |  |  |  |  |  |
| Let *α***u s***k* � 1, | *β***v s***k* �−1, |  | then we | have | | |  |
| **y***k*− **B***k***s***k*� **u** − **v**; let | **u** � **y***k*and |  | **v** � **B***k***s***k*, we |  | get | |  |

* � (1/**y***Tk***s***k*), *β* � − (1/**s***Tk***B***Tk***s***k*). Finally, the correction ma-trix is obtained:

**B***k*�*α***uu***T*+*β***vv***T*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | |  | *T* |  | 1 | |  | *T* |  |
| � |  | **uu** |  | − |  |  | **vv** | (37) |  |
| **y***kT***s***k* |  | **s***kT***B***kT***s***k* | |  |

�**y***k***y***Tk* −**B***k***s***k***s***kT***B***kT.*

**y***Tk***s***k***s***Tk***B***Tk***s***k*

**y***k*� **g***k*+1− **g***k,*

**s***k*�P*k*+1−P*k,*

**B***k*+1≈**H***k*+1*,*

**D***k*+1≈**B**−*k*+11*,*

(29)

(30)

(31)

(32)

above formula can be replaced by **B***k*+1�

**B***k*+ (**y***k***y***Tk*/**y***Tk***s***k*) − (**B***k***s***k***s***kT***B***kT*/**s***Tk***B***Tk***s***k*). We introduce the

identity matrix **I**, by using Sherman–Morrison formula, and

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| the above equation can be converted into **B***k*−+1 | | | | | | 1�(**I**− | | (**s***k***y***kT*/ |  |
| **y***kT***s***k*))**B***k*−1(**I** − (**y***k***s***kT*/**y***kT***s***k*)) + (**s***k***s***kT*/**y***kT***s***k*), that is: | | | | | | | | (38) |  |
|  | **s***k***y***kT* | | **y***k***s***kT* | | | **s***k***s***kT* | |  |
| **D***k*+1� **I** − |  | !**D***k* **I** − |  |  | ! + |  | *.* |  |
| **y***kT***s***k* | **y***kT***s***k* | **y***kT***s***k* |  |

8 Mobile Information Systems



1. **Input:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Initialize the feasible point ((1 − | | | | | *p* | *m* |  |  | *s* | | 0 | *m* 0 | | *s* 0 | | ) | *N* | , initialize the penalty coeﬃcient | | | | | | (0) | , set the dropping factor , |  |
|  | *i* | − *p* | | | ) *,* (*p* | | *i* | ) | *,* (*p* | ) | *i*�1 |  |  |
|  | *k* �0,**D**0�**I.** | |  |  |  |  |  |  | *i* |  |  |  | *i* |  |  |  | > |  |  |  |  | ∇ | Γ |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| (2) | Define *ε*1 and *ε*2 as a suﬃciently small positive real number, where *ε*1 | | | | | | | | | | | | | | | | | | | *ε*2. |  |  |  |  |
| (3) | Determine search direction *q* | | | | � − **D** · **g** . | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| (4) | Find the optimal step factor: *k* | | | |  | *k* |  |  | *k* | |  |  |  | (*k*)))*N* | | |  |  |  |  |  |  |  |  |  |  |
|  | *ℓk* �arg *min ϕ*(((1− | | *pm* − | *ps*)( | (*k*))*, pm* ( | | | | | (*k*))*, ps* ( | | | | + *ℓqk*) | | |  |  |  |  |  |  |  |  |
|  | **s***k*�*ℓkqk**ℓ* |  | *i* | *i* |  |  | *i* |  |  |  |  | *i* |  |  | *i*�1 | |  |  |  |  |  |  |  |  |  |  |  |
|  | **R** |  |  | ∇1 )*, pi* (∇ | | | |  | ∇1 ))*i*�1 �∇((1 − *pi* | | | | | | |  | − *pi* )(∇ )*, pi* | | | | (∇ )*, pi* | (∇ ))*i*�1 + **s***k* | | |  |  |
|  | ((1 − *pi* −∈ | *pi* )(∇1)*, pi* (∇ | | |  |  |  |  |
|  | *m* | *s*(*k*+ ) | *m* | (*k*+ ) | | *s* |  |  | (*k*+ ) | | | *N* |  |  |  | *m* | |  | *s*(*k*) |  | *m* | (*k*)*s* | (*k*) | *N* |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Iteration:

**While** ‖**g***k*+1‖>*ε*1 **do y***k*� **g***k*+1− **g***k*

**D***k*+1�(**I** − (**s***k***y***Tk*/**y***Tk***s***k*))**D***k*(**I** − (**y***k***s***Tk*/**y***Tk***s***k*)) + (**s***k***s***Tk*/**y***Tk***s***k*)

*k* � *k* +1

Go to step 4

**end while**

**output** ((1−*pmi*−*psi*)(∇(*k*))*, pmi*(∇(*k*))*, psi*(∇(*k*)))*Ni*�1

(6) Set the algorithm termination condition:

**while** ‖((1−*pmi*−*psi*)(∇(*k*))*, pmi*(∇(*k*))*, psi*(∇(*k*)))*Ni*�1− ((1−*pmi*−*psi*)0*,*(*pmi*)0*,*(*psi*)0)*Ni*�1‖>*ε*2 **do**

Iteration: ∇(*k*+1) � Γ∇(*k*) (*k* � 0, 1, 2, . . .)

((1 − *pmi* − *psi*)0*,* (*pmi*)0*,* (*psi*)0)*Ni*�1 � ((1 − *pmi* − *psi* )(∇(*k*))*, pmi*(∇(*k*))*, psi*(∇(*k*)))*Ni*�1, *k* � *k* + 1

**end while**

1. **Return:** ((1−*pmi*−*psi*)(∇(*k*))*, pmi*(∇(*k*))*, psi*(∇(*k*)))*Ni*�1
2. **Output** ((*pli*)∗*,*(*pmi*)∗*,*(*psi*)∗)*Ni*�1is the approximate optimal solution of the objective function.

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

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 continu-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ously, | and | | the | approximate | | optimal solution |
| ((*pl*) *,* (*pm*) *,* (*ps*) | | | | )*N* | is obtained. | task assignment |
| *i* |  | *i* | *i* | *i*�1 |  |  |
| algorithm∗ | | based∗ | on∗ | quasi-Newton interior point method is | | |
| shown in Algorithm [2.](#page8) | | | | |  |  |
| *5.1. Algorithm Complexity Analysis.* service diﬀerence | | | | | | |
| between∗ | base stations will aﬀect the oﬄoad probability, and | | | | | |

the number of MDs and base stations will aﬀect the algo-rithm 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 algo-rithm 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. 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 com-plexity 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. the complexity of the TA-NIP method is

*O*((2*N*)3∗*k*), and the complexity of the TA-QNIP methodis *O*(2*N* ∗*k*).

**6. Simulation Results**

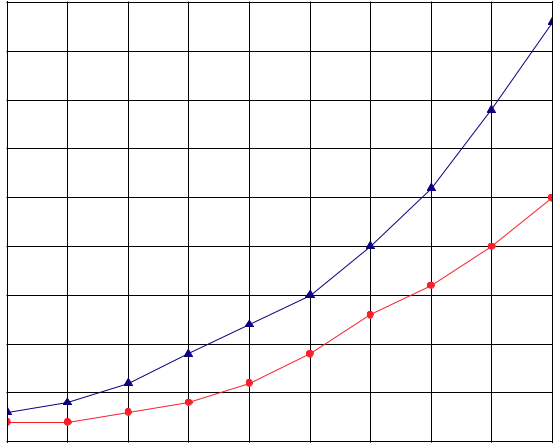
In this section, we evaluate the performance of the proposed TA-QNIP method through simulation results. At the same time, according to the simulation results, the advantages of the cooperative base station model are also proved. We consider that the distance *dm* between MBS and MDs is 1000 m, and the distance *ds* between SBS and MDs is 50 m [[34].](#page11) task generation rate of the device *λi* satisfies [0.1, 1.1] MB/s and the task size randomly generated by each device is *θi* � [2.5, 5] MB [[41].](#page12) Local device execution ca-pability is *uDi* � 0.5 GHz [[10],](#page11) and the CPU occupied pro-portion *lDi* of MD*i* is randomly selected in [0, 1]. response power coeﬃcient of MD*i* is set at*ξi* � 0.1, and MD*i*’s waiting power coeﬃcient is *ci* � 0*.*01 [[42].](#page12) channel bandwidth *B* � 5 MHz [[4],](#page11) Gaussian white noise power

* 2 � -127 dbm, and the transmission power *Pmi* and *Psi* 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 con-sumption 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.

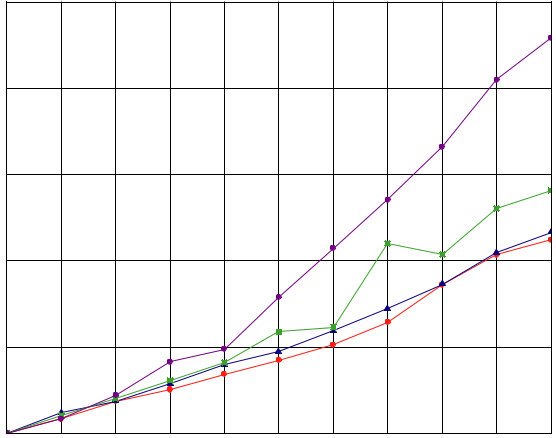
Figures [3](#page9) and [4](#page9) reflect the convergence of the TA-QNIP method and the TA-NIP algorithm. To facilitate the re-search, we discuss the convergence of the algorithm at *ρ* <0*.*8

Mobile Information Systems

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 45 |  |  |  |  |  |  |  |  |  | 25 |  |
|  | 40 |  |  |  |  |  |  |  |  |  |  |  |
|  | 35 |  |  |  |  |  |  |  |  |  | 20 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | 30 |  |  |  |  |  |  |  |  | Total consumption | 15 |  |
| Iterations | 25 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  | 10 |  |
|  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | 10 |  |  |  |  |  |  |  |  |  | 5 |  |
|  | 5 |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 0 |  |
|  | 10 |  |  |
|  |  |  |  |  | Number of MDs | |  |  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | 9 |
| 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|  |  |  |  | Number of MDs | | |  |  |  |  |

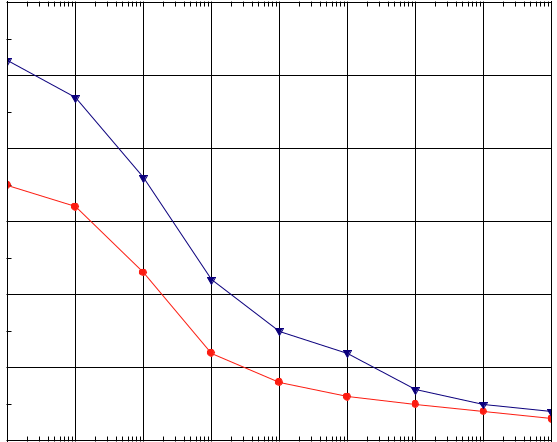


|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | TA-QNIP algorithm |  | TA-QNIP algorithm |  |  |  | Random task assignment |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  | algorithm |  |
|  |  | TA-NIP algorithm |  |  |  |  |  |  |
|  |  |  | TA-NIP algorithm |  |  |  | Task assignment algorithm |  |
| FIGURE 3: influence of the number of MDs on the iterations. | | |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  | under a single base station |  |



FIGURE 5: impact of the number of MDs.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 60 |  |  |  |  |  |  |  |  |  |
|  | 50 |  |  |  |  |  |  |  |  |  |
|  | 40 |  |  |  |  |  |  |  |  |  |
| Iterations | 30 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | 20 |  |  |  |  |  |  |  |  |  |
|  | 10 |  |  |  |  |  |  |  |  |  |
|  | 0 | 10–9 | 10–8 | 10–7 | 10–6 | 10–5 | 10–4 | 10–3 | 10–2 |  |
|  | 10–10 |  |
|  |  |  |  |  | Precision |  |  |  |  |  |



TA-QNIP algorithm



TA-NIP algorithm



FIGURE 4: eﬀect of algorithm precision on the iterations.

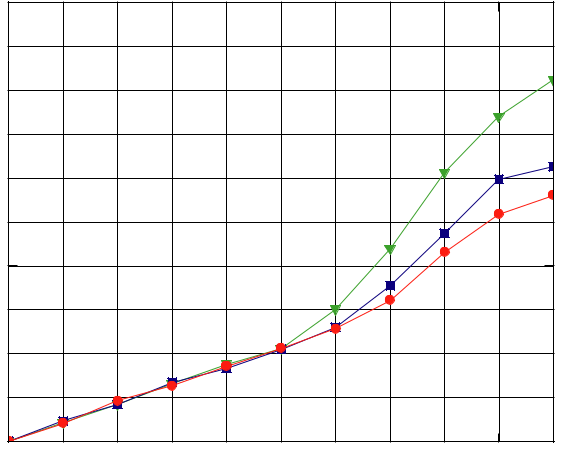
and *α* � 0*.*5. Figure [3](#page9) shows the influence of the number of MDs on the iterations; when the algorithm precision is 10− 5, the iterations of the two algorithms increase with the in-crease of the number of MDs, but the TA-QNIP algorithm shows better convergence performance, especially when the number of MDs increases significantly. Figure [4](#page9) 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 verifi-cation of the above simulation results, the TA-QNIP algo-rithm has better convergence performance.

As shown in Figure [5,](#page9) diﬀerent schemes are applied to optimize the total consumption of MDs. In order to reflect

the advantages of the proposed scheme in this paper, based on the TA-QNIP algorithm, the TA-NIP algorithm, random task assignment algorithm [[4],](#page11) and task assignment algo-rithm 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 al-gorithm 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 cor-rectness 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. 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 consump-tion of MDs is more.

Figure [6](#page10) shows the eﬀect of total consumption on dif-ferent queue strength. We discuss the situation when the delay weight coeﬃcient *α* � 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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 20 |  |  |  |  |  |  |  |  |  |  |  |
|  | 18 |  |  |  |  |  |  |  |  |  |  |  |
|  | 16 |  |  |  |  |  |  |  |  |  |  |  |
| consumption | 14 |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |
| Total |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4 |  |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |  |
|  | 0 |  |



Number of MDs

*ρ*<0.4



*ρ*<0.6

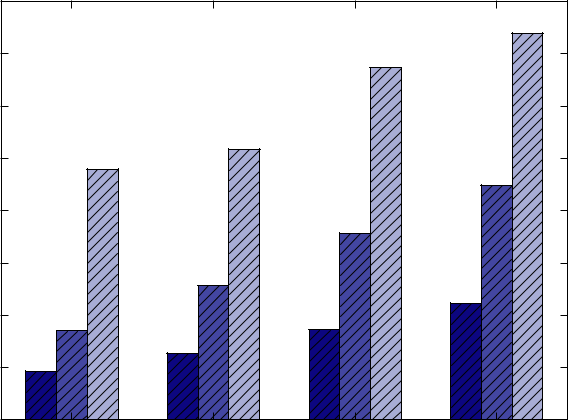


*ρ*<0.8



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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 16 |  |  |  |  |
|  | 14 |  |  |  |  |
|  | 12 |  |  |  |  |
| consumption | 10 |  |  |  |  |
| 8 |  |  |  |  |
|  |  |  |  |  |
| Total | 6 |  |  |  |  |
|  |  |  |  |  |
|  | 4 |  |  |  |  |
|  | 2 |  |  |  |  |
|  | 0 | 0.5 | 0.7 | 0.9 |  |
|  | 0.3 |  |



Delay weight coeﬃcient *α*

*N*=30



*N*=60



*N*=90



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

be oﬄoaded are successfully entered into the queuing se-quence. under the constraint of diﬀerent allow-

able 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 *ρ* <0*.*4 and *ρ* <0*.*6, respectively, the growth trend of the total MDs’ consumption is significantly faster than that of *ρ* <0*.*8, because the allowable queue length of the oﬀ-loading task is reduced, and the tasks to be oﬄoaded cannot successfully enter the queuing sequence, resulting in addi-tional consumption due to system congestion.

Figure [7](#page10) shows the influence of diﬀerent delay weight coeﬃcients *α* on the total consumption of MDs. It can be

Mobile Information Systems

seen that when the number of MDs is constant, with the

increase of delay weight and the decrease of energy con-

sumption weight, the types of tasks that MDs need to process

tend to be more sensitive to delay. 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 de-

mand 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, con-sidering 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**

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

**Conflicts of Interest**

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

**Acknowledgments**

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 In-formatics*, 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-

|  |  |
| --- | --- |
| Mobile Information Systems | 11 |

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

1. 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.
2. 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.
3. 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 Dis-tributed Computing*, vol. 135, pp. 140–155, 2020.
4. 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 Journal*, vol. 6, no. 3,

* + 1. 4162–4175, 2019.

1. 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.
2. 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*,
   * 1. 750–755, Paris, France, 2019.
3. 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.
4. 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.
5. C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, “Com-putation 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.
6. 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.
7. I. Ketyko,´ L. Kecskes,´ 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.
8. D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, “Mobile-edge computing architecture: the role of

MEC in the Internet of *IEEE Consumer Electronics*

*Magazine*, vol. 5, no. 4, pp. 84–91, 2016.

1. 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.
2. 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.
3. 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.
4. 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.
5. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication per-spective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
6. Q. Liu, T. Han, and N. Ansari, “Joint radio and computation resource management for low latency mobile edge comput-ing,” in *Proceedings of the 2018 IEEE Global Communications* *Conference*, pp. 1–7, Abu Dhabi, United Arab Emirates, 2018.
7. S. Sardellitti, M. Merluzzi, and S. Barbarossa, “Optimal as-sociation of mobile users to multi-access edge computing resources,” in *Proceedigs of the 2018 IEEE International* *Conference on Communications Workshops*, pp. 1–6, KansasCity, MO, USA, 2018.
8. M. Guan, B. Bai, L. Wang, S. Jin, and Z. Han, “Joint opti-mization for computation oﬄoading and resource allocation

in Internet of in *Proceedings of the 2017 IEEE 86th* *Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Tor-onto, Canada, 2017.

1. 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 In-ternational Conference on Distributed Computing Systems (ICDCS)*, pp. 615–624, Atlanta, GA, USA, 2017.
2. 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.
3. L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, “Multiobjective optimization for computation oﬄoading in fog computing,” *IEEE Internet of Journal*, vol. 5, no. 1,
   1. 283–294, 2018.
4. Z. Jiang and S. Mao, “Energy delay tradeoﬀ in cloud oﬄoading for multi-core mobile devices,” *IEEE Access*, vol. 3,
   1. 2306–2316, 2015.
5. L. Liu, Z. Chang, and X. Guo, “Socially aware dynamic computation oﬄoading scheme for fog computing system with energy harvesting devices,” *IEEE Internet of* *Journal*, vol. 5, no. 3, pp. 1869–1879, 2018.
6. Y. Yang, Y. Ma, W. Xiang, X. Gu, and H. Zhao, “Joint op-timization of energy consumption and packet scheduling for mobile edge computing in cyber-physical networks,” *IEEE* *Access*, vol. 6, pp. 15576–15586, 2018.
7. K. Li, “Computation oﬄoading strategy optimization with multiple heterogeneous servers in mobile edge computing,” *IEEE Transactions on Sustainable Computing*, vol. 1, 2019.
8. X. Guo, L. Liu, Z. Chang, and T. Ristaniemi, “Joint optimi-zation of energy and delay for computation oﬄoading in cloudlet-assisted mobile cloud computing,” *Wireless Net-works*, vol. 25, no. 4, pp. 2027–2040, 2019.
9. J. P. Zhou, *Communication Networks* , People’s

Posts and Telecommunications Press, China, Beijing, 2nd edition, 2009.

1. S. M. Ross, *Introduction to Probability Models*, pp. 481–558, Academic Press, Boston, MA, USA, 11th edition, 2014.
2. 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.
3. H. Liao, Z. Zhou, X. Zhao et al., “Learning-based context-aware resource allocation for edge computing-empowered industrial IoT,” *IEEE Internet of Journal*, vol. 34, 2019.

12

1. 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.
2. 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.
3. 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.
4. 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 Sym-posium on Computers and Communication (ISCC)*, pp. 589–596, Messina, Italy, 2016.
5. J. Nocedal, “Updating quasi-Newton matrices with limited storage,” *Mathematics of Computation*, vol. 35, no. 151, p. 773, 1980.
6. 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.
7. J. Du, L. Zhao, J. Feng, and X. Chu, “Computation oﬄoading and resource allocation in mixed fog/cloud computing sys-tems with min-max fairness guarantee,” *IEEE Transactions on* *Communications*, vol. 66, no. 4, pp. 1594–1608, 2018.

Mobile Information Systems