QoS Driven Task Offloading with Statistical Guarantee in Mobile Edge Computing

Qing Li, Shangguang Wang, *Senior Member, IEEE,* Ao Zhou, *Member, IEEE,* Xiao Ma, *Member, IEEE,* Fangchun Yang, *Senior Member, IEEE,* and Alex X. Liu *Fellow, IEEE*

Abstract—In mobile edge computing, popular mobile applications, such as augmented reality, usually offload their tasks to resource-rich edge servers. The user experience can be considerably affected when many mobile users compete for the limited communication and computation resources. The key technical challenge in task offloading is to guarantee the Quality of Service (QoS) for such applications. Existing work on task offloading focus on deterministic QoS (delay) guarantee, which means that tasks have to complete before the given deadline with 100%. However, it is impractical to impose a deterministic QoS guarantee for tasks due to the high dynamics of the wireless environment when offloading to edge servers. In this paper, we focus on task offloading with statistical QoS guarantee (tasks are allowed to complete before a given deadline with a probability above the given threshold), which can further save more energy by loosing the QoS requirement. Specially, we first propose a statistical computation model and a statistical transmission model to quantify the correlation between the statistical QoS guarantee and task offloading strategy. Then, we formulate the task offloading problem as a mixed integer non-Linear programming problem with the statistical delay constraint. We transform the statistical delay constraint into the constraints on CPU cycle numbers and the delay exponent respectively. We propose an algorithm to provide the statistical QoS guarantee for tasks using convex optimization theory and Gibbs sampling method. Experiment results show that the proposed algorithm outperforms the three baselines.

Index Terms—Mobile edge computing, task offloading, resource allocation, statistical QoS requirement, energy efficiency.

1 Introduction

1.1 Background & Motivation

In mobile edge computing, popular mobile applications, such as augmented reality and mobile object recognition, usually offload their tasks to nearby resource-rich edge servers [1–3] because mobile devices have limited computation resource and battery power. The key technical challenge in task offloading is to guarantee the delay-bounded Quality of Service (QoS) for such applications due to two reasons. First, the experience of mobile users can be considerably affected when a large number of offloading users compete for the limited communication and computation resources. Second, offloading computational tasks to edge servers can incur extra communication overhead in terms of delay and energy consumption.

However, most of the existing studies [4–9] on task offloading in mobile edge computing focuses on deterministic delay-bounded QoS guarantees, which means that tasks have to complete before the given deadline with 100%. Although they are effective to satisfy users' QoS requirement, it is difficult and impractical to impose a deterministic delay requirement for tasks in a highly dynamic wireless environment because the wireless environment is noisy and may result in transient performance drawdown [10]. In this paper, we concern task offloading problem with statistical delay-bounded QoS guarantee to maximize energy efficiency of mobile devices. Statistical delay-bounded QoS guarantee means that tasks are allowed to complete before a required deadline with a probability above a given threshold. For example, to guarantee the performance of multimedia applications, a video frame decoding task is required to complete before 10ms with a probability bigger than 96% [11, 12].

1.2 Challenges and Our Solutions

Supporting statistical delay-bounded QoS provisioning [13–17] for task offloading in mobile edge computing imposes two key challenges. The first challenge is how to quantify the correlation between the statistical delay requirement and the task offloading strategy. Only quantifying this correlation can we know how to make task offloading decisions to efficiently guarantee the specified statistical delay requirement under the constrained communication and computation resource. This is challenging because the correlation is not straightforward, which causes a gap between task offloading strategy design and statistical delay requirements. Mathematical analysis is badly in need to bridge this gap. The second technical challenge is to design an efficient holistic solution with low time complexity. This is challenging because mobile users are heterogeneous in computing capabilities and task requests, and the heterogeneity of edge server capacity further augments the complexity. In addition, the offloading decisions of different mobile users are coupled with each other. The time complexity of task offloading strategy is exponential when all the mobile devices are coordinated.

- *Qing Li, Shangguang Wang, Ao Zhou, Xiao Ma and Fangchun Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. E-mail: {q_li;sgwang;aozhou;maxiao18;fcyang}@bupt.edu.cn.*
- *Alex X. Liu is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, USA. E-mail: alexliu@cse.msu.edu.*

To deal with above issues, we propose a novel task offloading algorithm with statistical delay-bounded QoS guarantee by leveraging convex optimization theory and Gibbs sampling method. More precisely, we first propose a statistical computation model and a statistical transmission model to quantify the correlation between the statistical QoS guarantee and task offloading strategy. With the statistical computation model, we can provide the statistical QoS guarantee to save more energy by configuring CPU clock frequencies. With the statistical transmission model, we can provide the statistical QoS guarantee by introducing a statistical delay exponent to the traditional wireless transmission rate. Then, we formulate the task offloading problem as a Mixed Integer Non-Linear Program (MINLP) with the statistical delay constraint. By probability analysis and queue theory, we transform the statistical delay constraint into the constraint on the number of CPU cycles and the constraint on the statistical delay exponent respectively. At last, we propose a novel holistic task offloading algorithm with low time complexity to provide the statistical QoS guarantee. The algorithm works in an iterative manner, which has two layers. In the outer layer, the algorithm makes task offloading decisions based on a variation of Gibbs sampling. In the inner layer, the algorithm optimizes resource allocations by convex optimization under a fixed task offloading decision.

1.3 Our Contributions

The contributions of this paper are summarized as follows.

1) We conduct an end-to-end analysis for task offloading in mobile edge computing with the statistical QoS guarantee.
2) We propose a statistical computation model and a statistical transmission model to quantify the correlation between the statistical QoS guarantee and task offloading strategy.
3) We formulate the statistical QoS driven task offloading problem as an MINLP problem. And we propose a novel holistic task offloading algorithm based on convex optimization theory and a variation of Gibbs sampling. The proposed algorithm is proven to converge with a high probability to the global optimal solution.
4) We conduct extensive simulations to evaluate the proposed algorithm from two aspects: effectiveness and efficiency. In terms of effectiveness, the proposed algorithm can achieve near optimality in small network size and improve the energy efficiency of mobile devices by 17.7%, 31.1%, and 268.3% compared with the three baselines: hJTORA scheme, Greedy scheme and LOC scheme. In terms of efficiency, the proposed algorithm can converge and scale linearly with the network size.

The reminder of this paper is organized as follows. In Section 2, the related work is reviewed. The system model is introduced in Section 3. The problem formulation and analysis is presented in Section 4. In Section 5, the holistic solution for statistical QoS driven task offloading and resource allocation is developed. The numerical results are shown in Section 6. Conclusions are drawn in Section 7.

2 Related Work

Prior task offloading schemes in mobile edge computing system can be divided into two categories: deterministic QoS guarantee and statistical QoS guarantee.

Task Offloading with Deterministic QoS Guarantee: Most work has studied the task offloading problem with deterministic delay-bounded QoS guarantee [3–9]. These work can be divided into centralized schemes and distributed schemes. For the centralized schemes, some work focuses on the task offloading problem for a single user. Wang *et. al* [4] propose an algorithm to jointly optimize of the offloading ratio, transmit power and CPU-cycle frequency to minimize the mobile energy consumption (or latency) subject to a delay(or energy consumption) constraint for a single user. Zhang *et. al* [7] propose an energy-optimal execution strategy based on threshold to decide executing mobile applications in the mobile device or offloading to the cloud for a single user. While these works may be not practical when many users offload their tasks simultaneously. Some work focuses on task offloading problem for multiple users. Mao *et. al* [5] assume a stochastic task arrival model. They propose an online joint radio and computational resource management algorithm for multi-user mobile edge computing system based on Lyapunov optimization to minimize the long-term average power consumption of the mobile devices and the edge server. You *et. al* [8] prove that the optimal strategy for scheduling the offloading data size and time allocation has a threshold based structure for multi-user mobile edge computing system with respect to an offloading priority function derived from the mobile energy consumption and channel condition. These work solves the task offloading problem in a centralized manner, which may lead to high time complexity when network size scales up. Some other research targets distributed schemes using game theory. A distributed algorithm based on game theory has been designed to make task offloading decisions to minimize the energy consumption in [3]. Furthermore, this work was extended in [9], where the mobile device can offload tasks to multiple access points equipped with a common edge server. In comparison, we are interested in providing a statistical QoS guarantee for task offloading in mobile edge computing.

Task Offloading with Statistical QoS Guarantee: The statistical delay-bounded QoS provisioning has been proposed and shown to be a powerful technique to characterize and implement the statistical delay-bounded QoS provisioning for wireless transmission [18–22]. Zhang and Wang [19] propose the collaborative learning schemes by choosing the optimal operation strategies and power allocation policies through learning from the energy harvesting process while satisfying the heterogeneous statistical delay-bounded QoS constraints over full-duplex cognitive radio networks. Zhang *et al.* [20] establish the heterogeneous statistical QoS provisioning framework to support the diverse real-time services over the airborne mobile wireless networks. Cheng *et al.* [21] develop and efficient statistical delay-bounded QoS driven green power allocation schemes to maximize the effective power efficiency, enabling the effective implementation of green 5G wireless networks. The statistical QoS provisioning has been studied for edge

computing-enabled wireless networks [13–17]. Zhang and Zhu [13] propose efficient hierarchical edge caching mechanisms to guarantee the statistical delay-bounded QoS for multimedia transmissions while minimizing redundant transmissions. D2D communication schemes are proposed in [14] over edge-computing networks under the heterogeneous statistical delay-bounded QoS requirements. Wang *et al.* [15] propose a novel end-to-end effective capacity for the edge network to analyze the long-term performance of edge network slicing. Zhang and Zhu [16] propose a novel software-defined network architecture for heterogeneous statistical QoS provisioning over 5G multimedia mobile wireless networks based on network-function virtualization. They also propose the overall two-hop wireless link QoS provisioning schemes by deriving the overall effective capacity's expression of the two-hop tandem wireless links as a function of the single-hop's effective capacities [17]. Different from them, we study the end-to-end task offloading process with statistical QoS guarantee by determining the task offloading decisions and corresponding resource allocation to maximize mobile devices' energy efficiency in mobile edge computing.

3 System Model

We consider a multi-user multi-server mobile edge computing system in the ultra-dense network. Each base station (BS) is equipped with an edge server to provide computation ability to User Equipments (UEs). We introduce a new computation control entity in the system model, which is called Small cell Cloud Manager (SCM) [8]. We deploy the SCM as functions of one BS in the network area. The SCM makes task offloading strategy periodically at the beginning of each task offloading round. The task offloading strategy includes both task offloading decision (whether and where to offload) and corresponding resource allocation. SCM obtains the information of BSs, UEs, and network, e.g., edge computation resource availability, parameters and QoS requirements of UEs, and the instantaneous channel information. Then, the SCM optimize the offloading strategy. If tasks are executed in UEs (*local execution*), SCM will optimally schedule the CPU clock frequency of UEs. If tasks are executed in edge servers (*edge execution*), SCM will optimally allocate the transmit power of UEs and computational resource of edge servers. Finally, the offloading decision is delivered to UEs and BSs.

Denote the set of BSs (edge servers) and UEs as $\mathcal{S} = \{1, 2, \cdots M\}$ and $\mathcal{U} = \{1, 2, \cdots N\}$ respectively. BSs can establish wireless links with UEs. Orthogonal frequency division multiple access scheme is considered in the uplink and the operational frequency band B_{band} is divided into K equal sub-bands of size $B = B_{\text{band}}/K$. Let $\mathcal{B} = \{1, \cdots, K\}$ be the set of available sub-bands at each BS. Each UE is assigned to at most one sub-band. The task offloading strategy is defined as $\mathbf{a} = \{a_{ijk} | i \in \mathcal{U}, j \in \mathcal{S}, k \in \mathcal{B}\}$, where $a_{ijk} = 1$ indicates that task from the i_{th} UE is offloaded to the j_{th} BS on the k_{th} sub-band, otherwise $a_{ijk} = 0$. A feasible offloading decision must satisfy the constraints,

$$\sum_{k \in \mathcal{B}} \sum_{j \in \mathcal{S}_i} a_{ijk} \leq 1, \forall i \in \mathcal{U} \tag{1}$$

TABLE 1: Notation

Notation	Definition/Description
N	Number of UEs
M	Number of BSs
K	Number of sub-bands
T_s	Duration of an offloading round
L_i	Input data size of the i_{th} UE
X_i	Number of CPU cycles for computing one bit of tasks
$D_{i,\text{ddl}}$	Task completion deadline of the i_{th} UE
ρ_i	Task completion probability of the i_{th} UE
W_i	Number of CPU cycles required by the task
$W_{i,\rho}$	Required CPU cycles with task completion probability ρ
$f_i(w)$	Clock frequency of the CPU in each cycle
$\eta^1_{i,\rho}$	Energy efficiency of local execution
B	Bandwidth of a sub-band
a_{ijk}	Task offloading decision of the i_{th} UE
h_{ijk}	Channel gain of the i_{th} UE
p_i	Transmit power of the i_{th} UE
r_{ijk}	Transmission rate of the i_{th} UE
θ_i	Statistical delay exponent of the i_{th} UE
$C^e_i(\theta_i)$	Effective capacity of the i_{th} UE
Z_j	Computation capacity of the j_{th} BS
Z_{ij}	CPU cycles that edge the j_{th} server allocates to the i_{th} UE
λ_i	Weight of the i_{th} UE
$\eta^2_{i,\rho}$	Energy efficiency of edge execution

and

$$\sum_{i \in \mathcal{U}} a_{ijk} \leq 1, \forall j \in \mathcal{S}, k \in \mathcal{B}. \tag{2}$$

The set of UEs offloading their tasks to the j_{th} BS is denoted by $\mathcal{U}_j = \{i \in \mathcal{U} | \sum_{k \in \mathcal{B}} a_{ijk} = 1\}$, then the set of users those decide to offload their tasks can be denoted by $\mathcal{U}_{\text{off}} = \bigcup_{j \in \mathcal{S}} \mathcal{U}_j$ and the set of users those execute tasks locally can be denoted by $\mathcal{U} \backslash \mathcal{U}_{\text{off}}$. The main parameters are summarized in TABLE 1.

3.1 Task Model

Each UE has one computation task during an offloading decision round [4]. It is necessary to capture the essential of tasks when depicting them. Hence, a task model is characterized by a tuple of four parameters, denoted by $A_i(L_i, X_i, D_{i,\text{ddl}}, \rho_i)$, in which L_i specifies the size of input data necessary for the task execution, X_i is the number of CPU cycles demanded to compute one bit of tasks. X_i has shown to be a random variable and can be modeled by a Gamma distribution, which is suitable for characterizing the distribution of CPU cycle demands [7]. Tasks have a statistical QoS requirement with $D_{i,\text{ddl}}$ specifying the task completion deadline and ρ_i specifying the task completion probability, i.e., $\Pr(D_i \leq D_{i,\text{ddl}}) \geq \rho_i$. The task is atomic and cannot be divided into subtasks.

3.2 Statistical Computation Model

In local execution, the energy consumption is dominated by the CPU workload. The CPU workload can be measured by the number of CPU cycles required by tasks, denoted as W_i. We define the CPU workload as the product of the input

data size and the complexity of the algorithm in the task, which can be expressed as

$$W_i = L_i X_i. \tag{3}$$

The CPU power consists of the dynamic power, the short circuit power and leakage power, where the dynamic power dominates [12]. As a result, we only consider the dynamic power for the local execution. The energy consumption in local execution can be minimized by configuring the clock frequency of the chip via dynamic voltage scaling [12]. The clock frequency of UEs is defined as $\mathbf{f} = \{f_i(w) \,|i \in \mathcal{U}\}$, which cannot exceed the frequency limit $f_{i,\max}$. The energy per operation $E_{i,w}(f_i(w))$ is proportional to V^2, where V is the supply voltage to the chip [12]. Moreover, it has been observed that, when operating at low voltage limits, the clock frequency of the chip is approximately linear proportional to the voltage supply V [7]. The energy consumption per CPU cycle can be expressed as $E_{i,w}(f_i(w)) = \kappa f_i^2(w)$, where κ is the effective switched capacitance depending on the chip architecture.

In the statistical computation model, we define the statistical QoS requirement as that the task should finish the execution before the deadline with a probability above ρ_i by allocating $W_{i,\rho}$ CPU cycles, $F_{W_i}(W_{i,\rho}) = \Pr[W_i \leq W_{i,\rho}] \geq \rho_i$. Since W_i is a linear function of X_i , $W_{i,\rho}$ can be calculated as $W_{i,\rho} = F^{-1}_{W_i}(\rho_i) = L_i F^{-1}_{X_i}(\rho_i)$, where $F^{-1}_{X_i}(\rho_i)$ is the inverse cumulative distribution function of X_i. Therefore, we can derive the total energy consumption under statistical delay requirement as

$$E_i = \kappa \sum_{w=1}^{W_{i,\rho}} F^c_{W_i}(w)[f_i(w)]^2, \tag{4}$$

where $F^c_{W_i}(w) = 1 - F^c_{W_i}(w)$, which is the complementary cumulative distribution function of W_i. The completion time of local execution is

$$t_i^1 = \sum_{w=1}^{W_i} \frac{1}{f_i(w)}. \tag{5}$$

When the task execution fails to meet its deadline, it will continue to execute at the maximum frequency for completion. The additional computation energy is negligible when the task completion probability is very close to 1. The statistical computation model has shown its effectiveness in saving energy without substantially affecting performance.

3.3 Statistical Transmission Model

In edge execution, we make some assumptions. First, tasks have been replicated on the edge server initially. UEs only transmit the input data of size necessary for execution to the edge server. Second, the delay of edge server execution comprises the following three parts, (i) the time to transmit the input to edge servers, (ii) the time to execute tasks at the edge servers, and (iii) the time to transmit the output from edge servers back to users. Assume that the output data size is much less than that of input, and the downlink transmission rate is relatively high, so the return time of results is negligible.

We consider a block fading model for underlying wireless links. The channel coefficients stay invariant within a block of duration T_b (here $T_b < \min_i\{D_{i,\text{ddl}}\}$) and vary independently from block to block. Denote the channel gain of the i_{th} UE on the k_{th} sub-band of the j_{th} BS as h_{ijk}. SCM knows the probability distribution of h_{ijk}. Denote transmit power allocation strategy as $\mathbf{p} = \{p_i | 0 < p_i \leq p_{i,\max}\}$, where $p_{i,\max}$ is the maximal transmit power of the i_{th} UE.

When UEs transmit their tasks simultaneously, they will suffer interference from each other. The intra-cell interference is well mitigated due to that UEs connected to the same BS use different sub-bands. Still, the inter-cell interference from UEs of different cell using the same sub-band exists. In this case, the signal to interference and noise ratio of the i_{th} UE and the j_{th} BS on the k_{th} sub-band is given by: $\gamma_{ijk} = \frac{p_i h_{ijk}}{I_{ijk}+N_0}$, here, $I_{ijk} = \sum_{m\in\mathcal{S}\setminus\{j\}} \sum_{n\in\mathcal{U}\setminus\mathcal{U}_j} a_{nmk} P_n h_{nmk}$. For better tractability and simplicity, an achievable upper bound of I_{ijk} is given as follows, $\hat{I}_{ijk} = \sum_{m\in\mathcal{S}\setminus\{j\}} \sum_{n\in\mathcal{U}\setminus\mathcal{U}_j} a_{nmk} P_{n,\max} h_{nmk}$. And the estimation of signal to interference and noise ratio is $\hat{\gamma}_{ijk} = \frac{p_i h_{ijk}}{\hat{I}_{ijk}+N_0 B}$. Then the achievable data rate of the i_{th} UE associated with the j_{th} BS on the k_{th} sub-band in one block with duration T_b can be calculated as

$$r_{ijk} = T_b B \log_2(1 + \hat{\gamma}_{ijk}). \tag{6}$$

Denote θ_i ($\theta_i > 0$) to be the statistical delay exponent of the i_{th} UE. The effective capacity, which is the maximum constant arrival rate under the statistical delay requirement specified by θ_i , is as follows,

$$C_i^e(\theta_i) = -\frac{1}{\theta_i} \sum_{k\in\mathcal{B}} \sum_{j\in\mathcal{S}} \ln \mathbb{E}\left\{e^{-\theta_i a_{ijk} r_{ijk}}\right\}. \tag{7}$$

Let $g_{ijk}\,(h_{ijk})$ be the probability density distribution of h_{ijk}. $g_{ijk}\,(h_{ijk})$ is continuously differentiable in h_{ijk}, which is true for almost all practical situations. Therefore, the expectation in Eq.(7) with respect to the random variables h_{ijk} can be evaluated with the following expression,

$$C_i^e(\theta_i) = -\frac{1}{\theta_i} \sum_{k\in\mathcal{B}} \sum_{j\in\mathcal{S}} \ln \int_0^\infty e^{-\theta_i a_{ijk} r_{ijk}} g_{ijk}\,(h_{ijk}) dh_{ijk}. \tag{8}$$

We can derive the transmission time as

$$t_i^2 = \frac{LT_b}{C_i^e(\theta_i)}. \tag{9}$$

3.4 Edge Computation Resource Allocation Model

Edge servers are able to provide computation offloading service to multiple UEs concurrently. All task executions are parallel in each task offloading round. As mentioned in Section 3.1, the task A_i requires $W_{i,\rho}$ CPU cycles to complete the task in a probabilistic execution in the UE. The edge execution also requires $W_{i,\rho}$ CPU cycles and edge servers can automatically schedule the clock frequency in each CPU cycle for each task. The computing resources of each edge server are quantified by the computational rate z_j, expressed in terms of number of CPU cycles per second. After receiving the offloaded task from a UE, the server will execute the task on behalf of the UE and, upon completion, will return the output result back to the UE. The computing resource allocation strategy is defined as

$\mathbf{z} = \{z_{ij} | i \in \mathcal{U}, j \in \mathcal{S}\}$, in which z_{ij} ($z_{ij} > 0$) is the amount of computing resource that edge the j_{th} server allocates to task A_i offloaded from user $i \in \mathcal{U}_j$. Hence, clearly $z_{ij} = 0, \forall i \notin \mathcal{U}_j$. In addition, a feasible computation resource allocation strategy must satisfy the computation resource constraint, expressed as,

$$\sum_{i \in \mathcal{U}} z_{ij} \leq z_j, \forall j \in \mathcal{S}. \tag{10}$$

Given the computation resource assignment $\{z_{ij}, \forall j \in \mathcal{S}\}$, the execution time of task A_i at the edge servers is,

$$t_i^3 = \sum_{j \in \mathcal{S}} \frac{a_{ij} W_{i,\rho}}{z_{ij}}, \forall i \in \mathcal{U}. \tag{11}$$

In this paper, we only care about the energy efficiency of UEs, when the task being executed in the edge server, the execution time will influence the energy efficiency of the UE, hence we only consider the execution time but the energy consumption of the edge server.

3.5 Optimization Objective

As mentioned, UEs have limited energy capacity, energy-efficient design in mobile edge computing is urgent. Our Objective is to maximize the energy efficiency of all UEs under statistical QoS constraint. Energy efficiency is a performance metric that we learn form conventional wireless communication network, which is defined as the ratio of the overall transmitted bits to the totally consumed energy [23]. Throughout this paper, we expand the meaning of the energy efficiency to be the ratio of the overall executed(transmitted) bits of tasks to the totally energy consumption of UEs. In the local execution, the energy efficiency can be expressed as the ratio of input data size and the energy consumption, i.e.,

$$\eta_{i,\rho}^1 = \frac{L_i}{\kappa \sum\limits_{w=1}^{W_{i,\rho}} F_{W_i}^c(w) [f_i(w)]^2}. \tag{12}$$

In the edge execution, the energy efficiency can be expressed as the ratio of effective capacity and the energy consumption during a channel block,

$$\eta_{i,\rho}^2 = \frac{C_i^e(\theta_i)}{T_b(p_i + p_c)}, \tag{13}$$

where p_c is the static circuit power consumption.

4 Problem Formulation and Analysis

The weighted sum energy efficiency of all UEs can be written as,

$$\eta_\rho(\mathbf{a}, \mathbf{f}, \mathbf{p}, \mathbf{z}) = \sum_{i \in \mathcal{U}} \lambda_i \left((1 - a_{ij}) \eta_{i,\rho}^1 + a_{ij} \eta_{i,\rho}^2 \right), \tag{14}$$

with $\lambda_i \in [0, 1]$ specifying the priority of the i_{th} UE. And the total execution time can be expressed as,

$$D_i = (1 - a_{ij}) t_i^1 + a_{ij} \left(t_i^2 + t_i^3 \right). \tag{15}$$

Hence, the QoS driven task offloading problem with statistical guarantee is formulated as follows,

$$\begin{aligned}
\mathbf{P1} \quad & \max \eta_\rho(\mathbf{a}, \mathbf{f}, \mathbf{p}, \mathbf{z}) \\
\text{s.t.} \quad & \text{C1}: \Pr\left(D_i \leq D_{i,\text{ddl}}\right) \geq \rho_i, \forall i \in \mathcal{U} \\
& \text{C2}: a_{ijk} \in \{0, 1\}, \forall i \in \mathcal{U}, j \in \mathcal{S}, k \in \mathcal{B} \\
& \text{C3}: \sum_{i \in \mathcal{U}} a_{ijk} \leq 1, \forall j \in \mathcal{S}, k \in \mathcal{B} \\
& \text{C4}: \sum_{j \in \mathcal{S}} \sum_{k \in \mathcal{B}} a_{ijk} \leq 1, \forall i \in \mathcal{U} \\
& \text{C5}: 0 \leq f_i(w) \leq f_{i,\max}, \forall i \in \mathcal{U} \backslash \mathcal{U}_{\text{off}} \\
& \text{C6}: 0 \leq p_i \leq p_{i,\max}, \forall i \in \mathcal{U}_{\text{off}} \\
& \text{C7}: \sum_{i \in \mathcal{U}} z_{ij} \leq z_j, \forall j \in \mathcal{S} \\
& \text{C8}: z_{ij} \geq 0, \forall i \in \mathcal{U}, \forall j \in \mathcal{S}.
\end{aligned}$$

In P1, C1 reflects the delay constraint in a probabilistic manner. C2 and C3 imply that each task can be either executed locally or offloaded to at most one edge server on one sub-band. C4 implies that at most one user can establish a link with a BS on one sub-band. C5 and C6 are the computation frequency constraint and transmit power budget imposed by CPU of UE and the radio interface respectively. C7 and C8 state that the amount of computation resources that the edge server allocate to UEs should be positive, and that the allocated computation resource must not excess the edge server capacity. Obviously, P1 is an MINLP problem, which is NP-hard [24]. P1 is polynomially reducible to the knapsack problem [25], which is NP-complete. Even with the statistical delay constraints in C1, the problem is not tractable directly, so some equivalent transformations are made for both local execution and edge execution.

4.1 Equivalent Problem Transformation

First, a transformation of statistical delay constraint is made for statistical computation model. As stated in Section 3.1, the statistical QoS requirement is in the form of CPU cycle, namely, $\Pr[W_i \leq W_{i,\rho}] \geq \rho_i$. It is necessary to prove that the statistical delay constraint and statistical CPU cycle constraint are equivalent. Suppose the i_{th} UE executes its task locally, we have the following theorem.

Theorem 1. *With the statistical computation model, we can transform the statistic delay constraint* $\Pr(t_i^1 \leq D_{i,ddl}) \geq \rho_i$ *into* $\Pr(W_i \leq W_{i,\rho}) \geq \rho_i$ *and* $\sum\limits_{w=1}^{W_{i,\rho}} \frac{1}{f_i(w)} \leq D_{i,ddl}$.

Proof. As mentioned in Section 3.1, the local execution time $t_i^1 = \sum\limits_{w=1}^{W_i} \frac{1}{f_i(w)}$. We define $t_{i,\rho}^1 = \sum\limits_{w=1}^{W_{i,\rho}} \frac{1}{f_i(w)}$, by the time $t_{i,\rho}^1$, the task completes with a probability ρ_i while meets the deadline, that is, $\Pr(t_i^1 \leq t_{i,\rho}^1) \geq \rho_i$ and $t_{i,\rho}^1 \leq D_{i,\text{ddl}}$. And we have $\Pr(\sum\limits_{w=1}^{W_i} \frac{1}{f_i(w)} \leq \sum\limits_{w=1}^{W_{i,\rho}} \frac{1}{f_i(w)}) \geq \rho_i$ and $\sum\limits_{w=1}^{W_{i,\rho}} \frac{1}{f_i(w)} \leq D_{i,\text{ddl}}$. Therefore, $\Pr(W_i \leq W_{i,\rho}) \geq \rho_i$ and $\sum\limits_{w=1}^{W_{i,\rho}} \frac{1}{f_i(w)} \leq D_{i,\text{ddl}}$. □

Suppose the i_{th} UE offloads its task to the j_{th} BS, we have the following theorem to prove the statistical delay constraint and effective capacity constraint are equivalent.

Theorem 2. *With the statistical transmission model, we can transform the statistical delay constraint* $\Pr(t_i^2 + t_i^3 \leq D_{i,ddl}) \geq \rho_i$ *into the effective capacity constraint, i.e.,*

$$C_i^e(\theta_i) = -\frac{1}{\theta_i} \sum_{k\in\mathcal{B}} \sum_{j\in\mathcal{S}} \ln \mathbb{E}\left\{e^{-\theta_i a_{ijk} r_{ijk}}\right\} \geq L/T_s, \quad (16)$$

where, $\theta_i = \frac{-2\ln((1-\rho_i)/c)}{\left(D_{i,ddl}-t_i^3\right)L_i/T_s}$.

Proof. Suppose that data arrives in the transmit buffer at a constant rate $r_i^{\text{arrival}}(t) = \frac{L}{T_s}$. At time t, the delay experienced by the data, i.e., $t_i^2(t)$ is related to the queue length of the buffer $Q_i(t)$ through $Q_i(t) = \frac{L}{T_s} t_i^2(t)$. $Q_{i,\max}$ is the maximal buffer size. For a specified delay bound $D_{i,\max}(D_{i,\max} = D_{i,\text{ddl}} - t_i^3)$, a service constraint may require the delay-violation probability to be no greater than a certain threshold $(1-\rho_i)$, i.e., $\Pr\left\{t_i^2(\infty) > D_{i,\max}\right\} \leq 1-\rho_i$ where $t_i^2(\infty)$ represents the probability distribution of the delay experienced by packets at steady state.

Lemma 1. *[26] Assume that the joint process* $(Q_i(t), h_{ijk}(t))$*, where* $h_{ijk}(t)$ *is stationary and ergodic. When the system is stable, we have*

$$\Pr\left\{t_i^2(\infty) > D_{i,\max}\right\} \geq c\Pr\left\{Q_i(\infty) > Q_{i,\max}\right\} \quad (17)$$

and

$$\lim_{Q_{i,\max}\to\infty} \frac{\ln \Pr[Q_i(t) \geq Q_{i,\max}]}{Q_{i,\max}} = -\theta_i. \quad (18)$$

Here, c is some positive constant, $Q_i(\infty)$ *is the steady-state queue distribution of the buffer,* $Q_{i,\max} = \frac{L}{T_s} D_{i,\max}$.

Lemma 2. *[26] When the value of* $Q_{i,\max}$ *is typically large, the buffer overflow constraint will be met provided that*

$$\theta_i \geq \theta_{i,0} = \frac{-2\log((1-\rho_i)/c)}{(D_{i,\max})\, L_i/T_s}. \quad (19)$$

Lemma 3. *[26] QoS exponent satisfies* $\theta_i \geq \theta_{i,0}$ *whenever the constant arrival rate fulfills*

$$-\frac{1}{\theta_i} \sum_{j\in N} \ln \mathbb{E}\left\{e^{-\theta_i a_{ijk} r_{ijk}}\right\} \geq L_i/T_s. \quad (20)$$

Lemma 2 connects the delay-violation probability to the buffer overflow probability. Lemma 3 shows that a constraint on statistical delay-violation probability can be transformed into a requirement on the decay rate of buffer occupancy. Lemma 4 shows that a requirement on the decay rate of buffer occupancy can be transformed into effective capacity constraint. Therefore, the statistical delay constraint $\Pr(t_i^2 + t_i^3 \leq D_{i,\text{ddl}}) \geq \rho_i$ can be equivalently transformed into the effective capacity constraint. □

From Theorem 1 and Theorem 2, we can quantify the correlation between the statistical QoS guarantee and task offloading strategy. We transform P1 to P2,

$$\begin{aligned}
\mathbf{P2}\ & \max \eta_\rho(\mathbf{a}, \mathbf{f}, \mathbf{p}, \mathbf{z}) \\
\text{s.t.}\quad & \text{C1a}: \sum_{w=1}^{W_{i,\rho}} \tfrac{1}{f_i(w)} \leq D_{i,\text{ddl}}, \forall i \in \mathcal{U}\backslash\mathcal{U}_{\text{off}} \\
& \text{C1b}: C_i^e(\theta_i) \geq L_i/T_s, \forall i \in \mathcal{U}_{\text{off}} \\
& \text{C2} - \text{C8}.
\end{aligned}$$

Still, P2 is an MINLP problem, our goal is to design a holistic solution with low complexity that achieves competitive performance while being practical to implement, given the large number of UEs, edge servers, and sub-bands. As the problem is so complicated, the idea of our solution is to make problem decomposition according to the feature of the problem in the first place, and then settle the sub-problems one by one.

4.2 Problem Decomposition

Given the high complexity of the problem P2 due to the combinatorial nature of the task offloading decision, we firstly schedule the communication and computation resources under a fixed task offloading decision, which is stated in P3.

$$\begin{aligned}
\mathbf{P3}\quad & \max_{\mathbf{f},\mathbf{p},\mathbf{z}} \eta_\rho(\mathbf{a}, \mathbf{f}, \mathbf{p}, \mathbf{z}) \\
& \text{s.t. C1}, \text{C5} - \text{C8}.
\end{aligned}$$

Based on the scheduling result of resource allocation, the task offloading decisions are made in P4.

$$\begin{aligned}
\mathbf{P4}\quad & \max_{\mathbf{a}} \ \hat{\eta}_\rho(\mathbf{a}, \mathbf{f}^*, \mathbf{p}^*, \mathbf{z}^*) \\
& \text{s.t. C2} - \text{C4},
\end{aligned}$$

where $\hat{\eta}_\rho(\mathbf{a}, \mathbf{f}^*, \mathbf{p}^*, \mathbf{z}^*)$ is the optimal-value function corresponding to the resource allocation problem stated in P3. The problem decomposition makes the problem in P1 easier to address. In the next section, a holistic solution for problem in P1 is given.

5 Statistical QoS Driven Task Offloading Algorithm Design

In this section, we design a low-complexity holistic algorithm to guarantee the statistical QoS requirement for task offloading in mobile edge computing. First, we settle the resource allocation problem in P3 under a fixed task offloading decision. Then, we achieve the offloading decisions based on the resource allocation results.

5.1 Resource Allocation Mechanism

5.1.1 Local Computation Resource Allocation

In this section, we study the local computation resource allocation sub-problem to maximize the energy efficiency of UEs by optimally scheduling the clock frequency of the CPU. The problem can be expressed as,

$$\begin{aligned}
\mathbf{P3.1}\quad & \max_{\mathbf{f}} \sum_{i\in\mathcal{U}} \lambda_i(1-a_{ij}) \frac{L_i}{\kappa \sum_{w=1}^{W_{i,\rho}} F_{W_i}^c(w)[f_i(w)]^2} \\
& \text{s.t. C1a}, \ \text{C5}.
\end{aligned}$$

As stated in Theorem 4.1 of [7], the optimal clock scheduling vector is given by

$$f_i^*(w) = \frac{\varsigma_i}{D_{i,\text{ddl}}[F_{u,W}^c(w)]^{1/3}}, 1 \leq w \leq W_{i,\rho}, \quad (21)$$

where $\varsigma_i = \sum_{w=1}^{W_{i,\rho}} [F_{W_i}^c(w)]^{1/3}$. The optimal energy consumption of local execution is $E_i^* = \frac{\kappa}{D_{i,\text{ddl}}^2}\{\sum_{w=1}^{W_{i,\rho}} [F_{W_i}^c(w)]^{1/3}\}^3$.

Simplifying the above result, the optimal energy consumption for computation is $E_i^* = \frac{K_i(L_i)^3}{D_{i,\mathrm{ddl}}^2}$, where K_i is a constant factor determined by κ and ρ_i. And the optimal energy efficiency of local execution is

$$\eta_{i,\rho}^{1*} = \frac{D_{i,\mathrm{ddl}}^2}{K_i(L_i)^2} \tag{22}$$

5.1.2 Edge Server Computation Resource Allocation

Still, that how the edge server computation resource allocation affects the energy efficiency of task offloading is not evident. Form C1 in P1, here is

$$\Pr(t_i^2 \leq D_{i,\mathrm{ddl}} - t_i^3) \geq \rho_i. \tag{23}$$

Observing Eq.(23), we can see that reducing t_i^3 can expand the feasible region of p_i, which consequently decreases the optimal value of transmit power allocation. Hence, instead of directly solving transmit power allocation sub-problem, edge server computation resource allocation sub-problem is resolved in advance. We minimize the sum of execution time of all the tasks,

$$\mathbf{P3.2} \quad \min_{\mathbf{z}} \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{U}_j} \frac{\lambda_i W_{i,\rho}}{z_{ij}}$$
$$\text{s.t. C7, C8.}$$

Notice that the constraint in C8 is convex. Denote the objective function in P3.2 as $obj(\mathbf{z})$. By calculating the second-order derivatives of $obj(\mathbf{z})$ w.r.t. z_{ij}, here are, $\frac{\partial^2 obj(\mathbf{z})}{\partial {z_{ij}}^2} = \frac{\lambda_i W_{i,\rho}}{{z_{ij}}^3} > 0, \forall i \in \mathcal{U}_j, \forall j \in \mathcal{S}$, and $\frac{\partial^2 obj(\mathbf{z})}{\partial z_{ij}\partial z_{uv}} = 0, \forall (i,j) \neq (u,v), \forall i,u \in \mathcal{U}_j, \forall j,v \in \mathcal{S}$. The Hessian matrix of the objective function in P3.2 is diagonal with the strictly positive elements, thus it is positive-definite. Hence, P3.2 is a convex optimization problem, and can be solved using Karush-Kuhn-Tucker conditions. In particular, we can obtain the optimal edge server computation resource allocation as,

$$z_{ij}^* = \frac{z_j\sqrt{\lambda_i W_{i,\rho}}}{\sum\limits_{i\in\mathcal{U}_j}\sqrt{\lambda_i W_{i,\rho}}}, \forall j \in \mathcal{S}. \tag{24}$$

And the optimal execution time for the i_{th} UE offloading its task to edge sever j is $t_i^{3*} = \frac{\sqrt{W_{i,\rho}}\sum\limits_{i\in\mathcal{U}_j}\sqrt{\lambda_i W_{i,\rho}}}{z_j\sqrt{\lambda_i}}$.

5.1.3 Transmit Power Allocation

After the execution time in edge server is determined, the transmit power allocation problem is,

$$\mathbf{P3.3} \quad \max_{\mathbf{p}} \sum_{i\in\mathcal{U}} \lambda_i \frac{C_i^e(\theta_i)}{T_b p_i}$$
$$\text{s.t. C1b, C6.}$$

The energy efficiency of edge execution has the same properties in [23]. The properties demonstrate the quasi-concavity of $\eta_{i,\rho}^2$, and imply both the existence and the uniqueness of the global maximum for P3.3. More importantly, as a result of the quasi concavity, problem P3.3 can be solved iteratively by the derivative-aided bisection search method similar to [23]. With a fixed transmit power, the algorithm can find the maximum energy efficiency $\eta_{i,\rho}^{2*}$ and the sign of its derivative for a given transmit power, $p_i \leq p_{i,\max}$. Then the algorithm searches for the transmit power p_i^*, that results in the maximum energy efficiency by derivative-aided bisection power search in Algorithm 1.

Algorithm 1 Derivative-aided bisection search for transmit power allocation

Input:
a: A task offloading decision.
$p_{i,\max}$: Maximal transmit power.
Output:
$\mathbf{p}^*$: Optimal transmit power allocation strategy.

Solve the problem with $p_i = p_{i,\max}$.
Calculated the energy efficiency $\eta_{i,\rho}^2$ and the derivative $\frac{\mathrm{d}C_i^e(p_{i,\max})}{\mathrm{d}p_i}$ according to Eq.(13) and Eq.(25).
if $\frac{\mathrm{d}C_i^e(p_{i,\max})}{\mathrm{d}p_i} \geq 0$ **then**
  Return $p_{i,\max}$ (maximum value is achieved),
else
  $p^{(1)} = p_{i,\max}$.
end if
Solve the transmit power minimization (convex optimization) problem,$p_{i,\min} = \min_{p_i\geq 0} p_i$ under the constraints C1b and C6 to get the $P_{i,\min}$.
Calculate $\eta_{i,\rho}^{2*}(p_{i,\min})$ and $\frac{\mathrm{d}C_i^e(p_{i,\min})}{\mathrm{d}p_i}$.
if $\frac{\mathrm{d}C_i^e(p_{i,\min})}{\mathrm{d}p_i} \leq 0$ **then**
  Return $p_{i,\min}$(maximum value is achieved),
else
  $p^{(2)} = p_{i,\min}$.
end if
while no convergence **do**
  $p_i = \frac{p^{(1)}+p^{(2)}}{2}$.
  Calculate $\eta_{i,\rho}^{2*}(p_i)$ and $\frac{\mathrm{d}C_i^e(p_i)}{\mathrm{d}p_i}$.
  if $\frac{\mathrm{d}C_i^e(p_i)}{\mathrm{d}p_i} < 0$ **then**
    $p^{(1)} = p_i$.
  else
    $p^{(2)} = p_i$.
  end if
end while

5.2 Task Offloading Decision

Still, due to the NP-hardness of the task offloading problem, designing efficient algorithm to find the optimal task offloading decision remains an open issue. We propose a statistical QoS driven task offloading algorithm based on a variation of Gibbs sampling [27]. The task offloading decision evolves as an N-dimensional Markov chain in which the i_{th} dimension corresponds to i_{th} UE task offloading decision. Applying a variation of Gibbs sampling, we first construct a transition probability matrix to guarantee that the N-dimensional Markov chain is irreducible and aperiodic. Hence, the stationary distribution can be achieved, where we can obtain the optimal decision with high probability (see Theorem 3). The task offloading decision is achieved in a decentralized manner by performing improvement step one at a time. The proposed algorithm is described in Algorithm 2. In each iteration, a randomly selected UE

$$\frac{\mathrm{d}\mathcal{C}_u^e(\theta_i)}{\mathrm{d}p_i} = \log_2 e \sum_{j\in\mathcal{N}} \frac{a_{ijk} \ln \int_0^\infty e^{-\theta_i a_{ijk} T_b B \log_2(1+\frac{p_i h_{ijk}}{\tilde{I}_{ijk}+N_0 B})} \frac{h_{ijk}}{\tilde{I}_{ijk}+N_0 B+h_{ijk}p_i} g_{ijk}(h_{ijk}) dh_{ijk}}{\int_0^\infty e^{-\theta_i a_{ijk} r_{ijk}} g_{ijk}(h_{ijk}) dh_{ijk}}. \tag{25}$$

$$\Pr(S_{a_1,a_2} | S_{a'_1,a'_2}) = \begin{cases} \frac{e^{-f(S_{a'_1,a'_2})/\tau}}{2MK(e^{-f(S_{a'_1,a'_2})/\tau}+e^{-f(S_{a'_1,a'_2})/\tau})}, a'_1 = a_1 \text{or } a'_2 = a_2 \\ 0, \text{otherwise} \end{cases}. \tag{26}$$

Algorithm 2 Distributed statistical QoS driven task offloading algorithm

Input:
a: A feasible task offloading decision.
Output:
$\mathbf{a}^*, \mathbf{f}^*, \mathbf{p}^*, \mathbf{z}^*$: Optimal task offloading strategy.
1: Randomly choose a UE to alter offloading decision to $\tilde{a}_i$.
2: **if** $\tilde{a}_i$ is feasible **then**
3: $\tilde{\mathbf{a}} = \{\mathbf{a}_{-i}, \tilde{a}_i\}$.
4: Obtain the $\tilde{\mathbf{f}}, \tilde{\mathbf{p}}, \tilde{\mathbf{z}}$ by maximizing P3.
5: $Prob = \frac{1}{1+e^{\frac{(\eta_{i,\rho}-\eta_{\tilde{i},\rho})}{\tau}}}$,
6: With probability $Prob$,
7: $\{\mathbf{a}^* = \tilde{\mathbf{a}}, \mathbf{f}^* = \tilde{\mathbf{f}}, \mathbf{p}^* = \tilde{\mathbf{p}}, \mathbf{z}^* = \tilde{\mathbf{z}}.\}$
8: With probability $1 - Prob$,
9: the i_{th} UE keeps **a** unchanged.
10: Broadcast task offloading decision **a**.
11: **end if**
12: Return $\mathbf{a}^*, \mathbf{f}^*, \mathbf{p}^*, \mathbf{z}^*$, if the stopping criterion is satisfied, otherwise go to Line 1.

change its offloading decision a_i to $\tilde{a}_i$ (Line 4), then the optimal resource allocation $\mathbf{f}^*, \mathbf{p}^*, \mathbf{z}^*$ is derived by addressing P3 (Line 5). Afterwards, the offloading decision of the selected UE is updated to the new one with a probability $Prob = \frac{1}{1+e^{\frac{(\eta_{i,\rho}-\eta_{\tilde{i},\rho})}{\tau}}}$, which depends on the difference between $\eta_{i,\rho}$ and $\eta_{\tilde{i},\rho}$, or remain unchanged with a probability $1 - Prob$ (Line 6-11). Then offloading decision is updated (Line 12). The algorithm converges with a higher probability to the global optimal solution.

Theorem 3. *As τ ($\tau > 0$) decreases, Algorithm 2 converges with a higher probability to the global optimal solution of P4. When $\tau \to 0$, Algorithm 2 converges to the global optimal solution with probability 1.*

Proof. Let $\mathcal{X} = \{x_1, x_2, \cdots, x_{MK}\}$ be the offloading decision space of all the users. At each offloading round, the user choose an offloading decision $a_{ijk} \in \mathcal{X}$. We denote the offloading decision as **a**. Following the iterations of algorithm, **a** evolves as an N-dimensional Markov chains in which the i_{th} dimension corresponds to the i_{th} UE offloading decision. We begin with the case of two UEs and denote the state of Markov chains as S_{a_1,a_2} ,where $a_i \in \mathcal{X}, i = 1, 2$. Since only one user is selected to explore a new offloading decision at each iteration with equal probability among all BSs, we have the expression in Eq.(26), where $f(S_{a_1,a_2})$ is the objective function in P4. Then we derive the stationary distribution $\Pr^*$ for each state and examine the balance equation as follows

$$\begin{aligned} &\sum_{i=2}^{MK} \Pr^*(S_{x_1,x_1}) \times \Pr(S_{x_1,x_i} | S_{x_1,x_1}) \\ &= \sum_{i=2}^{MK} \Pr^*(S_{x_1,x_i}) \times \Pr(S_{x_1,x_1} | S_{x_1,x_i}) \end{aligned}. \tag{27}$$

Substituting Eq.(26) into Eq.(27), we have

$$\begin{aligned} &\sum_{i=2}^{MK} \Pr^*(S_{x_1,x_1}) \times \frac{e^{-f(S_{x_1,x_i})/\tau}}{2MK(e^{-f(S_{x_1,x_i})/\tau}+e^{-f(S_{x_1,x_i})/\tau})} \\ &= \sum_{i=2}^{MK} \Pr^*(S_{x_1,x_i}) \times \frac{e^{-f(S_{x_1,x_1})/\tau}}{2MK(e^{-f(S_{x_1,x_1})/\tau}+e^{-f(S_{x_1,x_1})/\tau})} \end{aligned}. \tag{28}$$

Observing the symmetry of equation Eq.(28), we note that the set of equations in Eq.(28) are balance if for arbitrary state $\tilde{S}$ in the strategy space Ω, the stationary distribution is $\Pr^*(\tilde{S}) = Ke^{-f(\tilde{S})/\tau}$, where K is a constant. By applying the probability conservation law, we obtain the stationary distribution for the Markov chain as

$$\overset{*}{\Pr}(\tilde{S}) = \frac{e^{-f(\tilde{S})/\tau}}{\sum_{S_i\in\Omega} e^{-f(\tilde{S}_i)/\tau}}, \tag{29}$$

for arbitrary state $\tilde{S}$ in the strategy space Ω. In addition, we observe that the Markov chain is irreducible and aperiodic. Therefore, the stationary distribution given in Eq.(29) is valid and unique. Let S^* be the optimal state which yields the maximum value in P4, i.e., $S^* = \arg\max_{S_i\in\Omega} f(S_i)$. From Eq.(29), we have $\lim_{\tau\to 0} \Pr^*(\tilde{S}) = 1$ which substantiates that the algorithm converges to the optimal state in probability. Finally, the analogous analysis can be straightforwardly extended to an N-dimensional Markov chain. □

6 Experiment Results

6.1 Evaluation Setup

Extensive experiments are performed on a MATLAB-based simulator. In the simulated network area, BSs and UEs are randomly located spatially according to the Poisson Point Process distribution with the density of μ_1 and μ_2, respectively. Each UE can establish a link with any BS. TGn pathloss model and Rician fading with 6dB Rician factor are considered. Main parameters are listed in TABLE 2.

We evaluate the performance of the proposed algorithm using the metric of energy efficiency from two aspects: effectiveness and efficiency. In the effectiveness evaluation, we first verify the near-optimality of the proposed algorithm

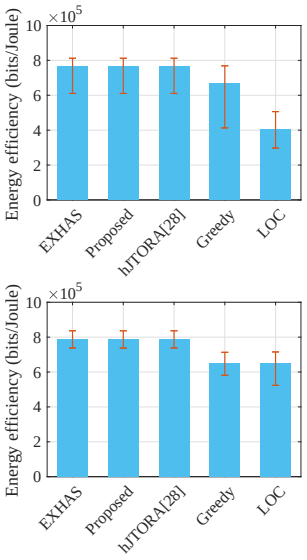


Fig. 1: Near-optimality.

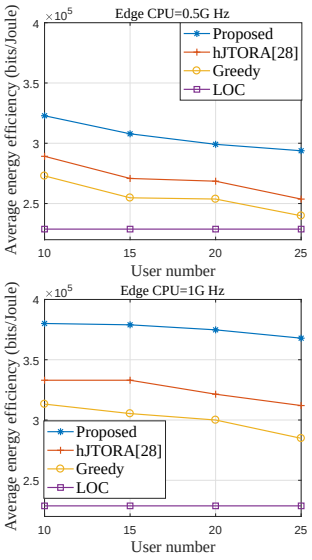


Fig. 2: UE number.

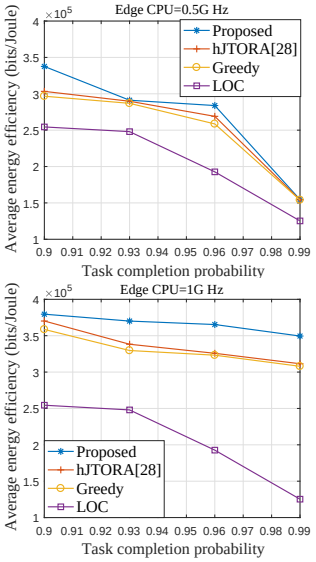


Fig. 3: Probability.

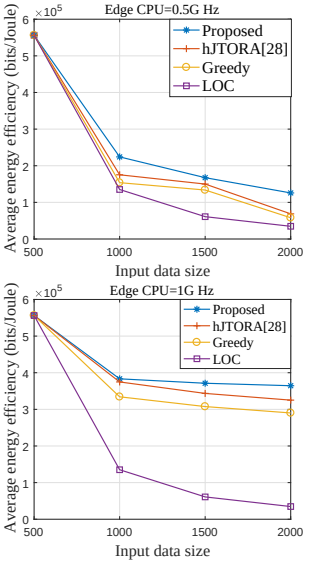


Fig. 4: Input data size.

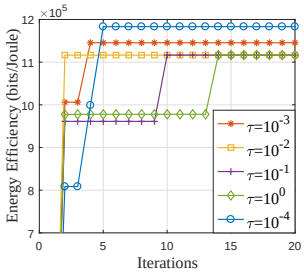


Fig. 5: Convergence.

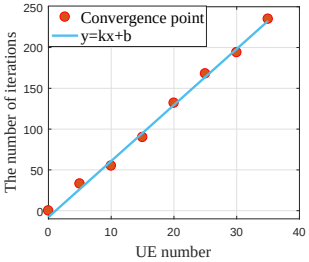


Fig. 6: Scalability.

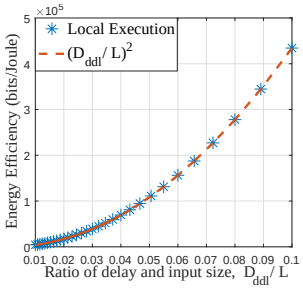


Fig. 7: Local execution.

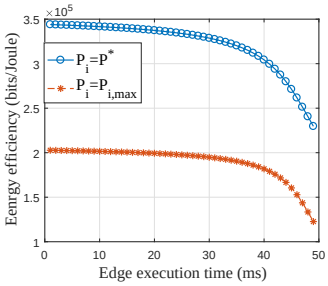


Fig. 8: Edge execution.

TABLE 2: Simulation Parameters

Parameters	Value
T_s	1 s (unchanged)
T_b	1 ms [23](unchanged)
κ	10^{-11} (unchanged)
B	15 KHz[23](unchanged)
$p_{i,\max}$	0.1 W [23](unchanged)
Noise power	-174 dBm (unchanged)
λ_i	1 (default)
L_i	1000 bits [7](default)
$D_{i,\max}$	50 ms [7](default)
ρ_i	0.995 [7](default)
z_j	1 GHz (default)

and then show the energy efficiency improvement compared with the baselines in different parameter settings. In the efficiency evaluation, we validate the convergence and scalability of the proposed algorithm. At last, we also validate the influence of some parameters on energy efficiency, which have been analyzed in theoretical results in Section 5. The proposed algorithm is compared with the following four algorithms.

1) *EXHAS (Exhaustive search)*: It is a brute-force method that finds the optimal offloading strategy via exhaustive search over $2^n (n = NMK)$ solutions.
2) *hJTORA [28] (holistic Joint Task Offloading and Resource Allocation)*: In this scheme, the offloading decisions are made by the hJTORA algorithm in [28]. The hJTORA algorithm also has two layers. In the out layer, offloading decision is improved based on any initial scheme by two operations: remove() and exchange(). It eventually converges to a local optimum. In the inner layer resources are optimally allocated under the offloading decision.
3) *Greedy*: In this scheme, all the tasks are offloaded and the offloading users are greedily assigned to the sub-bands with the highest channel gains until all the user are admitted or all the sub-bands are occupied.
4) *LOC (All-local execution)*: All tasks are executed locally, and the local computation resources are optimally scheduled.

6.2 Effectiveness

6.2.1 Near Optimality

The proposed algorithm's performance is the same as the exhaustive search method. Due to the high time complexity of exhaustive search method, the comparison is carried out in small network scales with $\mu_1 = 2, \mu_2 = 2$ and $\mu_1 = 2, \mu_2 = 3$. As shown in Fig. 1, the near-optimality

of the proposed algorithm in terms of the average energy efficiency (with 95% confidential interval) is guaranteed by Theorem 3. The hJTORA of [28] achieves the same performance with our algorithm. Note that when a user is added into the system, the energy efficiency of local execution is significantly increased, while the energy efficiency of the other three schemes is increased slightly. The reason is that when there are more users in the system, the competition for sub-bands and edge server computation resources will reduce the energy efficiency of edge execution.

6.2.2 Energy efficiency Improvement

Due to the exponential time complexity of the exhaustive method, the proposed algorithm is only compared with hJTORA, Greedy, and LOC algorithms when the network size goes large.

The average energy efficiency of the proposed algorithm is improved by 14.4%, 22.4% and 49.7% than hJTORA, Greedy, and LOC respectively with the increase of UE number. In Fig. 2, the variation of average energy efficiency (of each UE) is shown with $\mu_1 = 5$ and μ_2 varying from 10 to 25. There are two kinds of edge CPU speed settings: 0.5G Hz and 1G Hz. The proposed algorithm achieves the energy efficiency because it can converge to the global optimal solution with high probability by iteratively update the offloading decision. The hJTORA and Greedy algorithms always fall into local optimal solution. We observe that the energy efficiency of the proposed algorithm decrease with the increase of UE number. The reason is that, when the UE number increase, the competition for sub-band and edge computation resources would be fiercer, the edge computation resources available to each user is reduced averagely, which would lead to the decrease of the average energy efficiency.We also observe that the energy efficiency of all algorithms (except LOC) are higher when the edge CPU speed increases. The reason is that, higher edge CPU speed can lower the edge execution time and further loose the wireless transmission requirement, which lead to higher energy efficiency. The edge CPU speed has no influence on local execution.

The average energy efficiency of the proposed algorithm is improved by 6.5%, 8.8% and 61.2% than hJTORA, Greedy, and LOC respectively with the task completion probability. As shown in Fig. 3, the average energy efficiency of all solutions decreases with the increase of the task completion probability due to the two following reasons. Firstly, due to that the Gamma distribution is exponentially tailed, as the task completion probability increases, the energy consumption of local execution increases monotonically, as a result, the energy efficiency of local execution decreases monotonically. Secondly, in edge execution, the increase of task completion probability ρ will lead to the increase of delay exponent (see Lemma 3), which means a more stringent delay requirement. As effective capacity is a strictly and monotonically decreasing function of statistical delay exponent, the energy efficiency of edge execution will also decrease along with the increase of task completion probability ρ. The energy efficiency of all algorithms (except LOC) are higher when the edge CPU speed increases and the reason is the same as analyzed in UE number part (in the second paragraph of this subsection).

The average energy efficiency of the proposed algorithm is improved by 17.7%, 31.1%, and 268.3% than hJTORA, Greedy and LOC respectively with the variation of task input data size. As shown in Fig. 4, the average energy efficiency of all four solutions decreases in both edge CPU settings (0.5G Hz and 1G Hz) when the task input data size varies from 500 bits to 2000 bits. The decrease of average energy efficiency is due to the two following reasons. Firstly, in local execution, the increase of input data size can result in the decrease of energy efficiency directly (see Eq.(22)). Secondly, in edge execution, the increase of input data size means higher data arrival rate and longer edge computation time, both of which will lead to the decrease of energy efficiency. When the data size is small (say 500 bits), all the four solutions have same value of the average energy efficiency in that the local execution performs better that the edge execution, all the four solutions choose to execute the task on the UE. As the data size goes larger, the average energy efficiency of the local execution will decrease with a scaling law of $\frac{1}{L_i^2}$. The edge execution performs better than local execution. Hence, task offloading can improve UEs' energy efficiency significantly when the data size increases. Comparing the two different edge CPU settings, we can infer that the energy efficiency improvement of all the three offloading algorithms (Proposed, hJTORA, and Greedy) is higher when the edge CPU speed increases. Because higher edge CPU speed can reduce the edge execution time and the wireless transmission deadline can be loosened from Eq.(23). This can save more transmit power which lead to higher energy efficiency from Eqs.(7), (9) and (13).

6.3 Efficiency

The efficiency of the proposed algorithm is evaluated by the convergence and scalability. Convergence means the algorithm can achieve a stable solution after some iterations. Scalability means that the algorithm can scale well with the network size.

The proposed algorithm can converge to the near-optimal results when $\tau \leq 10^{-4}$. The impact of τ on the convergence of the proposed algorithm is shown in Fig. 5. It can be observed that when $\tau = 10^{-4}$, the algorithm converges to the global optimum. And the algorithm runs into local optimums several times before it finds the global optimum. When $\tau = 10^{-3}(10^{-2}, 10^{-1}, 10^{0})$, the proposed algorithm falls into the local optimum and converges to inferior solutions. The simulation result verifies Theorem 3 that the stationary probability of the optimal task offloading result increases with the decrease of τ, and the probability $\rightarrow 1$ when $\tau \rightarrow 0$. Because the smaller is τ, the more probable that the selected UE updates to the better task offloading decision in each iteration. Therefore, the smaller is τ, the more probable that the proposed algorithm converges to the optimal task offloading decision.

The proposed algorithm scales linearly with the variation of UE number, as shown in Fig. 6. We set $\mu_1 = 5$ and the UE number μ_2 varies from 0 to 35 with the increment of 5. And we record the iteration numbers when the proposed algorithm converges. All the iteration numbers are averaging over 100 times. We can fit the variation of iteration number with a linear function, $y = 6.871x - 7.857$. So the proposed algorithm can scale when network size goes large.

6.4 Effect of Specific Parameters on Energy Efficiency

The energy efficiency in local execution $\eta_{i,\rho}^{1*}$ scales at $\left(\frac{D_{i,\mathrm{ddl}}}{L_i}\right)^2$. In Fig. 7, the maximal energy efficiency of local execution is plotted as a function of the ratio of the task deadline (ranging from 20ms to 50ms) and the input data size (ranging from 500bits to 2000bits). Then it is compared with a scaling law of $\left(\frac{D_{i,\mathrm{ddl}}}{L_i}\right)^2$. The experiment result matches the calculation result in Eq.(22). It can be known that if the input data size increases and the delay constraint becomes stringent, the energy efficiency of local execution will decline. It infers that location execution may be not suitable for computation intensive and delay stringent tasks.

The energy efficiency in edge execution decrease along with the increase of the edge execution time. In Fig. 8, the energy efficiency in edge execution with the edge execution time under different transmit power is evaluated. As the edge execution time is relatively short, the energy efficiency decrease slightly. When the edge execution time exceed a certain threshold (about 40ms), the energy efficiency decrease sharply. The increase of edge execution time will influence the transmission time requirement directly (see Eq.(23)). This will decrease the overall effective capacity.

7 Conclusions

In this paper, we make the following four key contributions. First, we propose the statistical QoS guarantee for task offloading in mobile edge computing. Second, we quantify the correlation between statistical QoS requirement and task offloading strategy in mobile edge computing. Third, we propose an algorithm to provide the statistical QoS guarantee using convex optimization theory and Gibbs sampling method. Forth, we conducted extensive experiments in different parameter settings. Our results show the proposed algorithm can converge and significantly improve the energy efficiency. In the future works, we will investigate how to extend to scenarios with multiple QoS requirements.

References

[1] J. L. D. Neto, S. Yu, D. F. Macedo, J. M. S. Nogueira, R. Langar, and S. Secci, "ULOOF: A user level online offloading framework for mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2660–2674, 2018.

[2] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, 2017.

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

[4] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, 2016.

[5] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 9, pp. 5994–6009, 2017.

[6] W. Cheng, X. Zhang, and H. Zhang, "Full-duplex spectrum-sensing and MAC-protocol for multichannel nontime-slotted cognitive radio networks," *IEEE J. Select. Areas Commun.*, vol. 33, no. 5, pp. 820–831, 2015.

[7] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 9, pp. 4569–4581, 2013.

[8] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 3, pp. 1397–1411, 2017.

[9] X. Ma, S. Zhang, W. Li, P. Zhang, C. Lin, and X. Shen, "Cost-efficient workload scheduling in cloud assisted mobile edge computing," in *proc. IEEE/ACM IWQoS*, 2017, pp. 271–278.

[10] Q. Du and X. Zhang, "Statistical QoS provisionings for wireless unicast/multicast of multi-layer video streams," *IEEE J. Select. Areas Commun.*, vol. 28, no. 3, pp. 420–433, 2010.

[11] X. Peng, J. Ren, L. She, D. Zhang, J. Li, and Y. Zhang, "BOAT: A block-streaming app execution scheme for lightweight IoT devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1816–1829, 2018.

[12] W. Yuan and K. Nahrstedt, "Energy-efficient CPU scheduling for multimedia applications," *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 292–331, 2006.

[13] X. Zhang and Q. Zhu, "Hierarchical caching for statistical QoS guaranteed multimedia transmissions over 5G edge computing mobile wireless networks," *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 12–20, 2018.

[14] X. Zhang and J. Wang, "Heterogeneous statistical QoS-driven power allocation for collaborative D2D caching over edge-computing networks," in *proc. IEEE ICDCS*, 2019, pp. 944–953.

[15] Y. Wang, Y. Gu, and X. Tao, "Edge network slicing with statistical QoS provisioning," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1464–1467, 2019.

[16] X. Zhang and Q. Zhu, "Scalable virtualization and offloading-based software-defined architecture for heterogeneous statistical QoS provisioning over 5G multimedia mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 36, no. 12, pp. 2787–2804, 2018.

[17] X. Zhang and Q. Zhu, "D2D offloading for statistical QoS provisionings over 5G multimedia mobile wireless networks," in *proc. IEEE INFOCOM*, 2019, pp. 82–90.

[18] W. Cheng, X. Zhang, and H. Zhang, "Optimal power allocation with statistical QoS provisioning for D2D and cellular communications over underlaying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 34, no. 1, pp. 151–162, 2016.

[19] X. Zhang and J. Wang, "Heterogeneous statistical QoS driven collaborative learning based energy harvesting over full-duplex cognitive radio networks," in *proc. IEEE ICDCS*, 2019, pp. 831–840.

[20] X. Zhang, W. Cheng, and H. Zhang, "Heterogeneous statistical QoS provisioning over airborne mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 36, no. 9, pp. 2139–2152, 2018.

[21] W. Cheng, X. Zhang, and H. Zhang, "Statistical-QoS driven energy-efficiency optimization over green 5G

mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 34, no. 12, pp. 3092–3107, 2016.
[22] W. Cheng, W. Zhang, L. Liang, and H. Zhang, "Full-duplex for multi-channel cognitive radio Ad Hoc networks," *IEEE Netw.*, vol. 33, no. 2, pp. 118–124, 2019.
[23] C. Xiong, G. Y. Li, Y. Liu, Y. Chen, and S. Xu, "Energy-efficient design for downlink OFDMA with delay-sensitive traffic," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 6, pp. 3085–3095, 2013.
[24] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numer.*, vol. 22, pp. 1–131, 2013.
[25] R. Kannan and C. L. Monma, "On the computational complexity of integer programming problems," in *Optimization and Operations Research*. Springer, 1978.
[26] Dapeng Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Trans. Wirel. Commun.*, vol. 2, no. 4, pp. 630–643, 2003.
[27] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *proc. IEEE INFOCOM*, 2018, pp. 207–215.
[28] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, 2019.

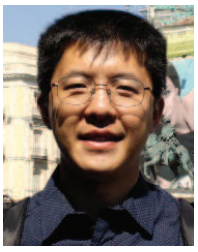
Qing Li received the B.S. and M.S. degrees from Hebei University in 2014 and Xidian University in 2017, respectively, both in communication engineering. She is currently a Ph.D. candidate at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include cloud computing and mobile edge computing.

Shangguang Wang received his PhD degree at Beijing University of Posts and Telecommunications in 2011. He is Professor and Vice-Director at the State Key Laboratory of Networking and Switching Technology (BUPT). He has published more than 100 papers, and played a key role at many international conferences, such as general chair and PC chair. His research interests include service computing, cloud computing, and mobile edge computing. He is a senior member of the IEEE, and the Editor-in-Chief of the International Journal of Web Science.

Ao Zhou received the P.H.D degrees in Beijing University of Posts and Telecommunications, Beijing, China, in 2015. She is currently an Associate Professor with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She has published 20+ research papers. She played a key role at many international conferences. Her research interests include Cloud Computing and Edge Computing.

Xiao Ma received her Ph.D. degree in Department of Computer Science and Technology from Tsinghua University and B.S. degree in Telecommunication Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018 and 2013, respectively. She is currently a postdoctoral fellow at the State Key Laboratory of Networking and Switching Technology, BUPT. From October 2016 to April 2017, she visited the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Her research interests include task scheduling, resource deployment and allocation in mobile cloud computing and mobile edge computing.

Fangchun Yang received his Ph.D. degree in communication and electronic system from the Beijing University of Posts and Telecommunication in 1990. He is currently a professor at the Beijing University of Posts and Telecommunication, China. His research interests include network intelligence and communications software. He is a fellow of the IET.



Alex X. Liu received his Ph.D. degree in Computer Science from The University of Texas at Austin in 2006, and is a professor at the Department of Computer Science and Engineering, Michigan State University. He received the IEEE & IFIP William C. Carter Award in 2004, a National Science Foundation CAREER award in 2009, and the Michigan State University Withrow Distinguished Scholar Award in 2011. He has served as an Editor for IEEE/ACM Transactions on Networking, and he is currently an Associate Editor for IEEE Transactions on Dependable and Secure Computing and IEEE Transactions on Mobile Computing, and an Area Editor for Computer Communications. He has served as the TPC Co-Chair for ICNP 2014 and IFIP Networking 2019. He received Best Paper Awards from ICNP-2012, SRDS-2012, and LISA-2010. His research interests focus on networking and security. He is a Fellow of the IEEE.