Joint optimization of computing ratio and access points’ density for mixed mobile edge/cloud computing

Tianqi Jing¹, Shiwen He²,³*, Fei Yu¹, Yongming Huang¹, Luxi Yang¹ and Ju Ren²

Abstract
Cooperation between the mobile edge computing (MEC) and the mobile cloud computing (MCC) in offloading computing could improve quality of service (QoS) of user equipments (UEs) with computation-intensive tasks. In this paper, in order to minimize the expect charge, we focus on the problem of how to offload the computation-intensive task from the resource-scarce UE to access point’s (AP) and the cloud, and the density allocation of APs’ at mobile edge. We consider three offloading computing modes and focus on the coverage probability of each mode and corresponding ergodic rates. The resulting optimization problem is a mixed-integer and non-convex problem in the objective function and constraints. We propose a low-complexity suboptimal algorithm called Iteration of Convex Optimization and Nonlinear Programming (ICONP) to solve it. Numerical results verify the better performance of our proposed algorithm. Optimal computing ratios and APs’ density allocation contribute to the charge saving.

Keywords: Computation offloading, Mobile edge computing, Density allocation, Mode selection

1 Introduction
With the rapid development of smart mobile user equipments (UEs), many applications with advanced features have emerged, such as augmented reality, facial recognition, and online games. The UEs who have computation-intensive applications to compute may demand powerful computing capacity and huge amounts of energy [1]. The demands lead to contradictions in UEs which are resource-scarce. The conflict between demands and equipments has become the bottleneck to improve the experience satisfaction of users. Mobile cloud computing (MCC) [2–4] has been proposed as a promising way to address challenges by offloading computing tasks to the cloud which has abundant computing resources and energy. However, for delay sensitive applications, the delay of cloud computing is noneligible because of the long distance between the terminal device and the cloud [5]. Meanwhile, the burden on fronthaul is huge, which may lead to heavy jam in data transmission and computing.

In order to solve above problems, the vision of mobile edge computing (MEC) [2] or Fog Computing [6] is proposed as a supplement to MCC [7], which enables applications
to run directly at the edge of the network. It extends the traditional cloud computing paradigm to the network edge [5] by putting a substantial amount of storage, communication, control, configuration, measurement, and management at the edge servers [8, 9]. With the help of MEC, low latency, location awareness, and high quality of service (QoS) for streaming media and real-time applications at resource-scarce UEs can be realized. To incorporate MEC in edge devices, some of the traditional access points (APs) are evolved to the edge computing-based access points by equipping with a certain caching, computing capabilities [10], which are more to be called as fog computing-based access points (F-APs).

Some outstanding works have been dedicated to computation offloading. [11] introduced many equivalence definitions of mobile edge computing, mobile edge computing platforms and architecture design. [12] and [13] discussed security threats of mobile edge computing, such as hacking. [14] illustrated the application of mobile edge computing in combination with the Internet of Things. In [15], the UEs, APs, and the cloud made up a three layer structure. They process a task collaboratively by offloading in the mixed MEC/MCC system. [16] and [17] thoroughly described the envisioned network architecture, proposed resource management scheme and analyzed its performance for edge/mobile edge computing.

There are also many previous works improve the system performance through the optimization of offloading decisions and resource allocation, such as the allocation of transmit power, bandwidth, and computation resource. Improvement of the system performance contains reduction in delay or energy consumption [18–20], minimization of the system cost [21, 22], improvement of QoS [23], maximization of the revenue of the server [24], adaptation user access mode selection mechanism [25]. However, most of those previous works put their emphasis on offloading decision making, resource allocation, or access mode selection, without a joint consideration of them.

Different from the above approaches, in this paper, we study the joint optimization of offloading decision making and access mode selection for a mixed MEC/MCC system to minimize the expect charge. It is embodied in optimization of computing ratios at each layers and the distribution density of APs. It is meaningful to study the distribution density of APs in MEC due to the edge servers have mobility and controllability. To the best of our knowledge, the joint design of offloading decision making and access mode selection in a mixed MEC/MCC system has not been addressed in previous works. The main contributions of this work are summarized as follows.

- We analyze the selection probability and corresponding ergodic rate of each mode.
- We formulate an optimization problem to minimize the expect charge of computing a task in the mixed MCC/MEC system. Due to the multi-access mode, the expect charge is in the form of the product of connecting probability of each mode and its corresponding charge.
- We devise a low complexity algorithm called Iteration of Convex Optimization and Nonlinear Programming (ICONP) to solve the formulated NP-hard optimization problem. It first fixes the specific variable and transform the original problem into a convex problem by geometric mean inequality method. Solve the convex problem by CVX tool and get the optimal values of other variables. Then fix those variables
which are got from last step and solve the problem with the specific variable by con-
strained nonlinear programming. Do iteration until meet convergence.
- We prove the convergence of the proposed algorithm. Simulation results show the
effectiveness of the proposed scheme with different system parameters.

The rest of this paper is organized as follows. The system model is described in Section 2.
The mode selection and its corresponding ergodic rates are represented in Section 3.
Section 4 formulates original problem. Section 5 represents the design of optimization
algorithm. Simulation results are discussed in Section 6. Finally, we conclude this study
in Section 7. “Appendix” can be seen in Sect. 8.

2 System model
We consider a three-layer mixed mobile edge/cloud uplink system, which is consisted of
a user equipment (UE), a large number of APs, and a remote cloud as illustrated in Fig. 1.
In this paper, the UE and APs are assumed to be equipped with a single antenna. APs
which is capable of computing are called as F-APs as well. F-APs are deployed according
to a two-dimensional PPP(Poisson Point Procession) \( \Phi_f \) with density of \( \lambda_1 \) in a disc plane,
whose center is the UE. Thus, the deployment of all the APs is an expanded homogene-
ous PPP \( \Phi_d \) with density of \( \lambda_2 = \lambda_1/k \), where \( k \in (0, 1] \) denotes the probability of an AP
supporting computation.

Without loss of generality, only one intensive computing task \( \Gamma_u = \{N, \omega\} \) needs to be
completed for the UE, where \( N \) is the size of computing task, \( \omega \) denotes the number of
CPU cycles required for computing one bit. In this paper, we assume that computing
task is divisible, which means that the computing task can be divided into two or more
parts.

Three computing modes are considered in this paper including F-AP execution mode, AP
relay mode, and local execution mode, denoted as mode \( i, i \in \Psi = \{1, 2, 3\} \), respectively.
Mode 1 means that the UE computes the task collaboratively with the F-AP and the cloud, while mode 2 means that the UE computes the task collaboratively with the cloud, and mode 3 is that the UE executes the task locally by adapting its computation capacity. In mode \(i, i \in \Psi\), the UE first processes \(\alpha_i\) of the task, where \(\alpha_i \in [0, 1], i \in \Psi\), and \(\alpha_3 = 1\).

Let \(\alpha = [\alpha_1, \alpha_2, 1]\). Then, the UE transmits the rest \((1 - \alpha_i)N\) bits to the selected AP. After that, the selected AP processes \(\rho = \max(\beta(2 - i), 0)\) of the received data and then transmits \((1 - \alpha_i)(1 - \beta(2 - i))N\) to the cloud, where \(\beta \in [0, 1]\). Finally, the cloud computes the received data.

When compute some data, the energy consumption \(E_c\) and time consumption \(T_c\) are given as \([26, 27]\)

\[
E_c = \kappa \cdot f^2 \cdot D_1 \tag{1}
\]

\[
T_c = \frac{D_1 \omega}{f} \tag{2}
\]

where \(\kappa\) denotes the effective capacitance coefficient, \(f\) is the computation capacity of the central processing unit (CPU), \(D_1\) is the size of computing data (in bits).

When transmit some data, the energy consumption \(E_t\) and time consumption \(T_t\) are given as

\[
E_t = p \cdot \frac{D_2}{r} \tag{3}
\]

\[
T_t = \frac{D_2}{r} \tag{4}
\]

where \(p\) and \(r\) denote the transmit power and rate, respectively. \(D_2\) is the size of transmitting data (in bits).

The size of computation outcome is much smaller than that of the computing task. Thus, the charge due to downlink transmission of the result is negligible compared to the uplink \([28, 29]\). Combined with Eq. (1)–(4), the charge is the sum of the product of consumed energy and its corresponding price and the product of computing delay and its corresponding price at each layer. The charge in mode \(i, i \in \Psi\), can be computed as

\[
Z_i = V_{\text{loc}} \left( \kappa \cdot f_{\text{loc}}^2 \cdot \alpha_i N \omega + p_1 \frac{(1 - \alpha_i)N}{R_i} \right) + G_{\text{loc}} \frac{\alpha_i N \omega}{f_{\text{loc}}} + 
V_{\text{AP}} \left( \beta(2 - i)(1 - \alpha_i)N \omega \cdot f_{\text{AP}}^2 + p_2 \frac{(1 + \beta(i - 2))(1 - \alpha_i)N}{r_i} \right) + 
G_{\text{AP}} \frac{\beta(2 - i)(1 - \alpha_i)N \omega}{f_{\text{AP}}} + 
G_{\text{C}} \frac{[1 + \beta(i - 2)](1 - \alpha_i)N \omega}{f_{\text{C}}} \tag{5}
\]

where \(f_{\text{loc}}\) is the local computation capacity (in CPU cycles/s) of the UE, \(f_{\text{AP}}\) is the computation capacity of the selected F-AP, \(f_{\text{AP}}, i \in \Upsilon = [2, 3]\), is a meaningless constant which is not equal to 0 for the rigor of the formula. \(f_{\text{C}}\) is the cloud’s computation capacity. \(R_i\) is the ergodic rate from the UE to the selected AP, while \(r_i\) is the transmission
rate from the selected AP to the cloud in mode $i, i \in \Omega = \{1, 2\}$, which will be discussed in the following section in details. $V_{\text{loc}}, V_{\text{AP}}$ and $V_{\text{Ci}}$ are prices per Joule (in yuan/J) at the UE, the selected AP and the cloud, respectively. The price raises in proportion to the corresponding amount of data needed to be computed or offloaded. For simplicity, we define $V_{\text{loc}} = \nu_1$, $V_{\text{AP}} = \nu_2 \cdot \beta^{2-i}(1 - \alpha_i)N$, $V_{\text{Ci}} = \nu_3 \cdot (1 - \alpha_i)(1 + \beta(i - 2))N$. $G_{\Omega}, \Omega \in \{\text{loc,AP,C}\}$ are prices per second (in yuan/s) for computing delay at the served UE, the selected AP and the cloud, respectively. All notations in this paper and their definitions are collected in Table 1.

3 Mode selection and ergodic rate

The UE first tries to select an F-AP which is nearest to it and the received signal-to-noise ratio (SNR) is larger than a pre-set SNR threshold $T_1$. If the UE cannot find an F-AP which meets the requirements, the UE will select an AP which is nearest to it meanwhile the SNR between them is larger than a pre-set SNR threshold $T_2$ as a relay. If neither of them can be achieved, the UE will compute data by itself. The probability of finding an AP which is nearest to the UE meanwhile the SNR between them is larger than $T_i$ is expressed as $F(\lambda_i), i \in \Omega$ [25].

$$F(\lambda_i) = \frac{1}{1 + T_1B_1\sigma^2/(p_1\lambda_i\pi)}, i \in \Omega$$ (6)

where $B_1, \sigma^2$ and $p_1$ are the transmission bandwidth, the mean noise power per Hz, and the transmission power of the UE. The prove of $F(\lambda_i)$ can be seen in “Appendix 1”.

The probability of selecting mode $i, i \in \Psi$, is denoted as $M_i$, and expressed as

$$M_1 = F(\lambda_1) = \frac{1}{1 + T_1B_1\sigma^2/(p_1\lambda_1\pi)}$$ (7)

$$M_2 = [1 - F(\lambda_1)]F(\lambda_2) = \left[1 - \frac{1}{1 + T_1B_1\sigma^2/(p_1\lambda_1\pi)}\right] \cdot \frac{1}{1 + T_2B_1\sigma^2/(p_1\lambda_2\pi)}$$ (8)

$$M_3 = \prod_{i=1}^{2} [1 - F(\lambda_i)] = \left[1 - \frac{1}{1 + T_1B_1\sigma^2/(p_1\lambda_1\pi)}\right] \cdot \left[1 - \frac{1}{1 + T_2B_1\sigma^2/(p_1\lambda_2\pi)}\right]$$ (9)

Next, we focus on the derivation of ergodic rate in mode $i, i \in \Omega$. Since the APs are deployed according to PPP, the ergodic rate (in bps) is defined as [25]

$$R = \mathbb{E}[B_1 \log_2(1 + \text{SNR})|\text{SNR} \geq T]$$ (10)

where $\mathbb{E}(\cdot)$ is the expectation with respect to the channel fading distribution as well as the locations of the random receiver nodes. SNR and $T$ are the real-time SNR and the pre-set threshold of SNR in the wireless connection between the UE and the selected AP, respectively.

The ergodic rate from the UE to the selected AP in mode $i, i \in \Omega$, can be derived as [25]
where \( \rho(\lambda_i) = \int_{\log_2(1+T_i)}^{\infty} \frac{P_1 \lambda_i \pi}{P_1 \lambda_i \pi + T_i B_1 \sigma^2} \, d\theta \). More details about \( R_i \) can be seen in "Appendix 2".

The transmission rate from the selected AP to the cloud is \([25]\):

\[
R_i = B_1 \left[ \rho(\lambda_i) + \log_2(1 + T_i) \frac{P_1 \lambda_i \pi}{P_1 \lambda_i \pi + T_i B_1 \sigma^2} \right]
\]

(11)

where \( \rho(\lambda_i) \) is the probability of choosing mode 1, \( F_{\min} \) and \( F_{\max} \) are the lower and upper bound of probability, respectively. Due to the relationship between \( \lambda_1 \) and \( \lambda_2 \), constraint \( C5 \) contains the constraint of \( F(\lambda_2) \).

4 Problem formulation

Since three execution modes are all likely to occur, the overall charge in our paper is defined as expected charge. Expected charge is the sum of product of select probability and corresponding charge of each mode, i.e.,

\[
P = \sum_{i=1}^{3} M_i \cdot Z_i
\]

(13)

In this paper, the objective is to minimize the charge of offloading computing, which is formulated as follows:

\[
\mathcal{P}_1 : \min_{\alpha, \beta, \lambda_1, \lambda_2} P
\]

s.t \( C1 : (1 - \alpha_i) N \leq n_1, i \in \Omega \)

\( C2 : [1 - \beta(2 - i)](1 - \alpha_i) N \leq n_2, i \in \Omega \)

\( C3 : 0 \leq \beta \leq 1 \)

\( C4 : 0 \leq \alpha_i \leq 1, i \in \Omega \)

\( C5 : F_{\min} \leq F(\lambda_1) \leq F_{\max} \)

(14)

The constraint \( C1 \) means the size of offloading data which is offloaded from the UE to the selected F-AP should be no larger than \( n_1 \), \( C2 \) guarantees that the data size is no larger than \( n_2 \) when it offloaded from the selected AP to the cloud, where \( n_1 \) is the maximum receive capacity to the UE offered by the selected AP, \( n_2 \) is the maximum receive capacity of the cloud which is offered to the selected AP. Constraints \( C3 \) and \( C4 \) ensure that the computing ratio should no more than 1 and no smaller than 0. Constraint \( C5 \) ensures multi-mode corporation, where \( F(\lambda_1) \) is the probability of choosing mode 1, \( F_{\min} \) and \( F_{\max} \) are the lower and upper bound of probability, respectively. Due to the relationship between \( \lambda_1 \) and \( \lambda_2 \), constraint \( C5 \) contains the constraint of \( F(\lambda_2) \). It is hard to solve this complex and non-convex problem due to the existence of product relationship between variables in the objective function and the constraint \( C2 \). Thus, we need to reduce the complexity and get the suboptimal values of variables by transforming the problem into a convex form.
5 Design of optimization algorithm

Note that $\lambda_1$ is related to transmission rate and the probability of each mode's selection. The coupling among $\lambda_1$, $\beta$, and $\alpha$ makes transforming the objective function into a convex form difficult. To overcome these difficulties, we propose to address problem $P_1$ in an alternative manner. Specifically, we firstly solve problem $P_1$ with respect to $\alpha$ and $\beta$ for fixed $\lambda_1$. Then, we solve problem $P_1$ with respect to $\lambda_1$ for fixed $\alpha$ and $\beta$. Do iteration until convergence.

When the value of $\lambda_1$ is given, the value of $M_i$, $i \in \Psi$, and transmission rate $R_i$, $r_i$, $i \in \Omega$, are all known. By taking various expressions which had been illustrated above into the problem, the objective function and constraints of problem $P_2$ are shown as bellow.

$$\mathcal{P}_2 : \min_{\alpha_1, \alpha_2, \beta} \quad M_1 \left\{ v_1 \left[ \kappa \cdot f_{loc}^2 \cdot \alpha_1 N \omega + p_1 \left( 1 - \alpha_1 \right) N \right] + G_{loc} \frac{\alpha_1 N \omega}{f_{loc}} \right. \left. + v_2 \beta^2 \left( 1 - \alpha_1 \right)^2 N^2 \kappa \cdot f_{AP}^2 \omega + v_3 \beta N^2 p_2 \frac{\left( 1 - \beta \right) \left( 1 - \alpha_1 \right)^2}{r_1} \right. \left. + G_{AP} \frac{\beta \left( 1 - \alpha_1 \right) N \omega}{f_{AP}} + v_3 \cdot (1 - \beta)^2 (1 - \alpha_1)^2 N^2 \kappa \cdot f_C^2 \omega \right. \left. + G_C \frac{(1 - \beta) \left( 1 - \alpha_1 \right) N \omega}{f_C} \right\} + M_2 \left\{ v_1 \left[ \kappa \cdot f_{loc}^2 \cdot \alpha_2 N \omega + p_1 \left( 1 - \alpha_2 \right) N \right] \right. \left. + G_{loc} \frac{\alpha_2 N \omega}{f_{loc}} + v_2 (1 - \alpha_2) N \cdot p_2 \frac{\left( 1 - \alpha_2 \right) N}{r_2} + v_3 \cdot (1 - \alpha_2)^2 N^2 \kappa \cdot f_C^2 \omega \right. \left. + G_C \frac{(1 - \alpha_2) N \omega}{f_C} \right\} + M_3 \left( v_1 \kappa \cdot f_{loc}^2 N \omega + G_{loc} \frac{N \omega}{f_{loc}} \right)$$

s.t $C3$

$C6 : \left( 1 - \alpha_1 \right) N \leq n_1$

$C7 : \left( 1 - \alpha_2 \right) N \leq n_1$

$C8 : \left( 1 - \beta \right) (1 - \alpha_1) N \leq n_2$

$C9 : \left( 1 - \alpha_2 \right) N \leq n_2$

$C10 : 0 \leq \alpha_1 \leq 1$

$C11 : 0 \leq \alpha_2 \leq 1$

(15)

where the constraints $C6$ and $C7$ come from $C1$ when $i=1$ and 2, respectively. The constraints $C8$ and $C9$ come from $C2$ when $i=1$ and 2. The constraints $C10$ and $C11$ come from $C4$ when $i=1$ and 2.

In problem $P_2$, the objective function and the constraint $C8$ exist product relationships between variables $\alpha_1$ and $\beta$. It is obviously that the objective function and constraint $C8$ are not convex. The remaining constraints are linear. Before solving this problem, it is necessary to transform them into convex forms. In the arithmetic geometric mean inequality theorem, for real numbers $a$, $b$, there exists $a^2 + b^2 \geq 2ab$. So $\frac{a^2 + b^2}{2}$ is the upper bound of the value of $ab$. Based on arithmetic geometric mean inequality theorem, problem $P_2$ is relaxed to problem $P_3$ whose objective and constraints are transformed according to variables $\alpha$ and $\beta$. 
\[ \mathcal{P}_3: \min_{\alpha_1, \alpha_2, \beta} M_1 \left\{ v_1 \left( \kappa \cdot f_{loc}^2 \cdot \alpha_1 N \omega + \frac{p_1(1-\alpha_1)N}{R_1} \right) + G_{loc} \frac{\alpha_1 N \omega}{f_{loc}} \right. \\
+ \frac{v_2}{2} \frac{\beta^4 + (1-\alpha_1)^4}{4} N^2 \kappa \cdot f_{AP}^2 \omega + \frac{v_2 N^2 p_2}{4} \frac{\beta^2 + (1-\beta)^4}{4r_1} \\
+ \frac{v_2 N^2 p_2}{4} \frac{\beta^2 + (1-\alpha_1)^8}{4r_1} + G_{AP} N \omega \frac{\beta^2 + (1-\alpha_1)^2}{2f_{AP_1}} \\
+ v_3 \cdot \frac{(1-\beta)^4 + (1-\alpha_1)^4}{2} N^2 \kappa \cdot f_C^2 \omega + G_C N \omega \frac{(1-\beta)^2 + (1-\alpha_1)^2}{2f_C} \right\} \\
+ M_2 \left\{ v_1 \left[ \kappa \cdot f_{loc}^2 \cdot \alpha_2 N \omega + \frac{p_1(1-\alpha_2)N}{R_2} \right] + v_2(1-\alpha_2)N \cdot p_2 \frac{(1-\alpha_2)N}{r_2} \\
+ G_{loc} \frac{\alpha_2 N \omega}{f_{loc}} + v_3 \cdot (1-\alpha_2)^2 N^2 \kappa \cdot f_C^2 \omega + G_C \frac{(1-\alpha_2)N}{f_C} \right\} \\
\left. + M_3 \left\{ v_1 \kappa \cdot f_C^2 N \omega + G_{loc} \frac{N \omega}{f_{loc}} \right\} \right\} \\
\text{s.t} \ C3, C6-7, C9-11 \\
C12: \frac{(1-\beta)^2 + (1-\alpha_1)^2}{2} N \leq n_2 \]

(16)

The second derivative of the objective and constraints of problem \(\mathcal{P}_3\) with respect to the variable \(\alpha\) and \(\beta\) are greater than or equal to 0. Thus, the problem \(\mathcal{P}_3\) is a convex problem, which can be solved by CVX tool easily and efficiently. When the values of \(\alpha\) and \(\beta\) are given, the optimal solution of \(\lambda_1\) can be obtained via solving the following problem \(\mathcal{P}_4\). The expression of the objective is same as the objective in problem \(\mathcal{P}_3\). However, the unknown variable is \(\lambda_1\) in problem \(\mathcal{P}_3\). Thus, the constraint is related to variable \(\lambda_1\) in problem \(\mathcal{P}_4\) as constraint \(C5\).

\[ \mathcal{P}_4: \min_{\lambda_1} P(\lambda_1) \]
\[ \text{s.t} \ C5 \]

where

\[
P(\lambda_1) = \frac{1}{1 + T_1 B_1 \sigma^2 / p_1 \lambda_1 \pi} \left\{ v_1 \left( \kappa \cdot f_{loc}^2 \cdot \alpha_1 N \omega + \frac{p_1(1-\alpha_1)N}{R_1} \right) + G_{loc} \frac{\alpha_1 N \omega}{f_{loc}} \right. \\
+ \frac{v_2}{2} \frac{\beta^4 + (1-\alpha_1)^4}{4} N^2 \kappa \cdot f_{AP}^2 \omega + \frac{v_2 N^2 p_2}{4} \frac{\beta^2 + (1-\beta)^4}{4r_1} \\
+ \frac{v_2 N^2 p_2}{4} \frac{\beta^2 + (1-\alpha_1)^8}{4r_1} + G_{AP} N \omega \frac{\beta^2 + (1-\alpha_1)^2}{2f_{AP_1}} \\
+ v_3 \cdot \frac{(1-\beta)^4 + (1-\alpha_1)^4}{2} N^2 \kappa \cdot f_C^2 \omega + G_C N \omega \frac{(1-\beta)^2 + (1-\alpha_1)^2}{2f_C} \right\} \\
+ \frac{1}{1 + T_1 B_1 \sigma^2 / p_1 \lambda_1 \pi} \left\{ v_1 \kappa \cdot f_{loc}^2 \cdot \alpha_2 N \omega \\
+ \frac{p_1(1-\alpha_2)N}{R_2} \right\} + G_{loc} \frac{\alpha_2 N \omega}{f_{loc}} + v_2(1-\alpha_2)N \cdot p_2 \frac{(1-\alpha_2)N}{r_2} \\
+ v_3 \cdot (1-\alpha_2)^2 N^2 \kappa \cdot f_C^2 \omega + G_C \frac{(1-\alpha_2)N \omega}{f_C} \right\} \\
\left. + \left( v_1 \kappa f_{loc}^2 \cdot N \omega + G_C \frac{N \omega}{f_C} \right) \cdot \prod_{i=1}^{2} \left( 1 - \frac{1}{1 + T_i B_1 \sigma^2 / p_1 \lambda_i \pi} \right) \right\} 
\]

(18)
\( P_4 \) is a nonlinear constrained optimization problem which only contains variable \( \lambda_1 \). We can get the range of \( \lambda_1 \) from the constraint C5 and record as \( \lambda_{\text{min}} \leq \lambda_1 \leq \lambda_{\text{max}} \). There only exists one inequality constraint in \( P_4 \), thus we can get the optimal value of \( \lambda_1 \) by interior point penalty function method [30]. The main idea of penalty function method is to transform nonlinear constrained optimization problem into nonlinear unconstrained optimization problem. Firstly, define barrier function

\[
G(\lambda_1, r) = P(\lambda_1) + r \left( \frac{1}{\lambda_1 - \lambda_{\text{max}}} + \frac{1}{\lambda_i - \lambda_{\text{min}}} \right)
\]

(19)

where \( r \) is a very small positive number. In this way, when \( \lambda_1 \) is close to \( \lambda_{\text{min}} \) or \( \lambda_{\text{max}} \), \( G(\lambda_1, r) \) is tending to infinity. Otherwise, \( G(\lambda_1, r) \approx P(\lambda_1) \). Thus, we can solve \( P_5 \) to get the optimal value of \( \lambda_1 \) equivalently.

\[
P_5 : \min_{\lambda_1} G(\lambda_1, r)
\]

(20)

\( P_5 \) is a nonlinear unconstrained optimization problem and can be solved by one dimensional linear search method. The derivative of the objective can be solved by Newton’s method [31]. The derivative of \( G(\lambda_1, r) \) with respect to \( \lambda_1 \) is denoted as \( g(\lambda_1, r) \). For one dimensional search function \( g(\lambda_1, r) \), suppose that a close point to the extreme minimum point has been given as \( \delta^0 \). Near the point \( \delta^0 \), we use a quadratic function \( h(\delta, r) \) to approximate the original function \( g(\delta, r) \). The original function is obtained by Taylor expansion as

\[
g(\delta, r) \approx h(\delta, r) = g(\delta^0, r) + g'(\delta^0, r)(\delta - \delta^0) + \frac{1}{2}g''(\delta^0, r)(\delta - \delta^0)^2
\]

(21)

where \( g'(\delta^0, r) = \frac{dg(\delta^0, r)}{d\delta} \bigg|_{\delta=\delta^0} \) and \( g''(\delta^0, r) = \frac{d^2g(\delta^0, r)}{d\delta^2} \bigg|_{\delta=\delta^0} \). Then the extreme minimum point of the quadratic function \( h(\lambda_1, r) \) is used as the new close point to the extreme minimum point of \( G(\delta, r) \), and record as \( \delta^1 \). According to the necessary conditions of extreme value, \( \delta^1 = \delta^0 - \frac{g'(\delta^0, r)}{g''(\delta^0, r)} \). From the equation \( \frac{dh(\delta, r)}{d\delta} = 0 \), we can draw the update formula as

\[
\delta^{m+1} = \delta^m - \frac{g'(\delta^m, r)}{g''(\delta^m, r)}.
\]

The algorithm is shown in Algorithm 1.

According to the definition of \( G(\lambda_1, r) \), the smaller the \( r \) is, the closer the solution of \( P_5 \) to the solution of \( P_4 \). Thus, we adopt Series Unconstrained Minimization Method (SUMT) to make the solution of \( P_5 \) more closer to the solution of \( P_4 \) [30]. Set an infinite penalty factor series \( \{r_k\} \) which is strictly monotonic decreasing and tending to zero. Then solve \( G(\lambda_1, r_k) \) according to each \( r_k \) until meet the iterative termination requirement. The complete algorithm of solving \( P_4 \) is shown in Algorithm 2.

---

**Algorithm 1 Newton’s Method**

1. Receive original value and record as \( \delta^0 \), permissible error \( \epsilon > 0 \), \( m = 0 \)
2. \( g(\lambda_1, r) = \frac{dg(\lambda_1, r)}{d\lambda_1} \)
3. \( g'(\delta^m, r_k) = \frac{dg'(\delta^m, r_k)}{d\delta} \), \( g''(\delta^m, r_k) = \frac{d^2g(\delta^m, r_k)}{d\delta^2} \)
4. Update \( \delta^{m+1} = \delta^m - \frac{g'(\delta^m, r_k)}{g''(\delta^m, r_k)} \)
5. If \( |\delta^{m+1} - \delta^m| \leq \epsilon \) then
6. \( \lambda_k = \delta^{k+1} \), break
7. Else
8. \( m = m + 1 \), go back to step 2.
9. end if
Finally, take the $\lambda_1$ which is obtained by Algorithm 2 back to the problem $P_3$. $\lambda_1$ is a known value in $P_3$ and then derive optimized value of $\alpha$ and $\beta$ by CVX tool. After that, we solve problem $P_4$ with fixed $\alpha$ and $\beta$. In conclusion, the algorithm for solving $P_2$ is firstly solving problem $P_3$ with respect to $\alpha$ and $\beta$ for fixed $\lambda_1$. Then, we solve problem $P_4$ with respect to $\lambda_1$ for fixed $\alpha$ and $\beta$. When we solve problem $P_3$ with respect to $\alpha$ and $\beta$ for fixed $\lambda_1$, the value of $P$ with optimized $\alpha$ and $\beta$ is smaller than before. Similarly, when we solve problem $P_4$ with respect to $\lambda_1$ for fixed $\alpha$ and $\beta$, the value of $P$ with optimized $\lambda_1$ is smaller than before. Thus, the algorithm of solving $P_2$ is convergent, and it is shown in Algorithm 3.

### Algorithm 2 Interior Point Penalty Function Method

1. Set original value $\lambda^0 \in [\lambda_{\min}, \lambda_{\max}]$, permissible error $\epsilon > 0$, initial parameter $\tau_1 > 0$, $k = 1$
2. $\lambda^{k-1}$ as the original point, use Algorithm 1 to solve $P_3$, record the optimal value as $\lambda^k$
3. If $r_1(\frac{1}{\tau_1^2} + \frac{1}{\tau_2^2}) < \epsilon$ then
   4. break, $\lambda_1 = \lambda^k$
   5. Else
   6. $\tau_{k+1} = \tau_k$, $\tau \in (0, 1)$, $k = k + 1$, go back to step 2
   7. end if

6 Simulation results and analysis

In this section, the impact of $N$, $n_1$, and $n_2$ on latency, computing ratios, expect charge are evaluated by using MATLAB with CVX tool. The simulation parameters are listed as follows in Table 2.

Figure 2a, b show the delay and charge of the offloading system with an increasing data size $N$ when $n_1 = 1200$, $n_2 = 800$. The delay and the charge of the system increase with the increasing of the data size. Compared with local computing, the proposed offloading strategy can improve the QoS by saving about 4 seconds and 1.5 yuan when facing the same data size of the computing task under simulation parameters we set. This is because the objective function is a balance of energy consumption and delay at each layer. Thus, it will not only charge less, but also spend less time than local computing.

Next the computing ratios of each layers in mode 1, mode 2, and allocation of F-APs’ distribution density versus the value of data size are shown in Fig. 3a, b, and Fig. 4, where $n_1 = 1200$, $n_2 = 800$. In Fig. 3a, with the increasing of data size, the UE firstly computes
none, and then the computing ratio of the UE keeps increasing. The computing ratio of the F-AP is firstly unchanged, then increasing, and finally decreasing. The computing ratio of the cloud is firstly unchanged, then decreasing, and finally decreasing. Staying unchanged when the data size is smaller than 1000 bits is because that, the optimal data size which computed at each layer to minimize the charge is smaller than its receive capacity. When the data size becomes larger than 1000 bits and smaller than 1200 bits, the data which is optimized to offload to the cloud is larger than its receive capacity. Thus, the computing ratio of the cloud decreases. Meanwhile, the data which is optimized to offload to the F-AP is smaller than its receive capacity. That is why the computing ratio of the F-AP increases with the increase of data size of the task. When data size is larger than the F-AP's receive capacity, the UE needs to compute the part which is larger than \( n_1 \). Thus, when the data size of the task is larger than \( n_1 \), the larger the data

![Fig. 2 Charge and delay with different value of data size](image)

**Fig. 2** Charge and delay with different value of data size. **a** Charge with different value of data size, **b** Delay with different value of data size.

**Table 1 Notation**

| Notation | Definition | Notation | Definition |
|----------|------------|----------|------------|
| \( N \)  | data size of the task | \( f_{loc}, f_{AP}, f_C \) | Computation capacity of the UE, AP and cloud |
| \( \omega \) | CPU cycles for 1 bit | \( R_i \) | Ergodic rate between the UE and AP in mode \( i \) |
| \( \kappa \) | Effective capacitance coefficient | \( n_1, n_2 \) | Upper bounds of capacity of channel |
| \( B_1, B_2 \) | Transmit bandwidth of UE and AP | \( \eta \) | Ergodic rate between the AP and cloud in mode \( i \) |
| \( p_1, p_2 \) | Transmit power of the UE and AP | \( \nu_1, \nu_2, \nu_3 \) | Unit price of UE, AP and cloud |
| \( \sigma^2 \) | Noise density | \( \lambda_i \) | Distribution density of F-AP(\( i = 1 \)) and AP(\( i = 2 \)) |
| \( M_i \) | Coverage probability of mode \( i, i \in \Omega \) | \( F_{min}, F_{max} \) | Upper and lower bounds of coverage probability |
| \( \alpha, \beta \) | Optimized variables | \( T_i \) | Pre-set threshold of SNR value in mode \( i, i \in \Omega \) |
| \( \Omega \) | Set of \( \{1,2\} \) | \( \Psi \) | Set of \( \{1,2,3\} \) |
| \( Z_i \) | Charge of mode \( i, i \in \Psi \) | \( k \) | Probability of an AP supporting computation |
| \( G_{ij}, \bar{U} \in \{loc, AP, C\} \) | Prices per second for computing delay at UE, AP and the cloud | \( P \) | Expect charge of the system |
size is, the more ratio of the task the UE needs to compute. Meanwhile, both computing ratio of the F-AP and cloud decrease. In Fig. 3b, the cloud computes all the task while the UE computes none when the data size is smaller than $n_2$. When the data size is larger than $n_2$, the computing ratio of the UE keeps increasing while cloud’s keeps decreasing. This is because when the optimized data size which is allocated to the cloud is smaller than its receive capacity, the computing ratios of the cloud and the UE stay unchanged. When the optimized offloaded data size is larger than $n_2$, the amount of offloaded data is fixed at $n_2$ and the computing ratio of the cloud is decreasing while the computing ratio of the UE is increasing. The change of data size do not affect the optimal value of $\lambda_1$ as shown in Fig.4. This is due to the increase of data size has no relationship with the allocation of distribution density of F-APs.

Figure 5a, b shows the expect charge and distribution density of F-APs versus the values of $n_1$ and $n_2$, where $N = 2000$. In Fig. 5a, the expect charge decreases with the increase of $n_1$ and $n_2$. This is because the computing power of upper layers is larger than the UE, it could save charge by offloading. The optimized computing ratios at each layer are limited by the receive capacity of upper layers. Larger $n_1$ and $n_2$ mean the UE is permitted to offload more data to upper layers when do optimized allocation of the task. When receive capacity are larger than the optimized computing data size which allocated to corresponding layer, the computing ratios and charge stay unchanged with the increasing of $n_1$ and $n_2$. In Fig. 5b, the distribution density of F-AP is hardly influenced by $n_1$ and $n_2$ unless $n_1$ and $n_2$ are small. This is because when the receive capacity of the F-AP is too small to receive offloaded data, the whole task is computed locally. In that case, the value of distribution density of the F-APs do not need optimization. When the receive capacities become larger, the UE can offload data to upper layers. The distribution density of the F-APs has to be optimized to minimize the charge of the offloading system. From the simulation we found that, there is no direct connection between the distribution density of F-AP and receive capacity when the UE can do offloading.

Figure 6a–c shows the computing ratios at each layer in mode 1 versus the values of $n_1$ and $n_2$, where $N = 2000$. In Fig.6a, the UE’s computing ratio decreases with the increase of $n_1$ and $n_2$. This is because the UE will offload part of task to upper layers to save charge after optimization. The offloading size of data is limited by receive capacities. When the optimized allocated data size at upper layers is larger than their receive capacity, the UE

| Parameter | value         | Parameter | value         |
|-----------|---------------|-----------|---------------|
| $\kappa$  | 1e-28         | $f_{\text{AP}}$ | 3000 cycles/s |
| $\omega$  | 10 cycles/s   | $f_{\text{C}}$ | 4000 cycles/s |
| $B_2$     | 100MHz        | $B_1$     | 10MHz         |
| $v_1$     | 0.3 yuan/J    | $v_2$     | 0.5 yuan/J    |
| $p_1$     | 0.5W          | $v_3$     | 0.7 yuan/J    |
| $p_2$     | 1.5W          | $P_{\text{Fmin}}$ | 0.5          |
| $f_{\text{loc}}$ | 1500 cycles/s | $P_{\text{Fmax}}$ | 0.8          |
| $k$       | 0.8           | $G_{\text{loc}}$ | 0.3 yuan/s   |
| $G_{\text{AP}}$ | 0.2 yuan/s | $G_{\text{C}}$ | 0.1 yuan/s   |
| $\sigma^2$ | -174dBm/Hz    |           |               |
Fig. 3  Computing ratios with different data size. a Computing ratios with different data size in mode 1; b Computing ratios with different data size in mode 2

Fig. 4  Distribution density with different value of data size

Fig. 5  Expect charge and distribution density with different value of receive capacity. a Expect charge with different value of $n_1, n_2$; b Expect charge with different value of $n_1, n_2$
will offload as much as possible in the limit of receive capacity. In Fig. 6b, the F-AP’s computing ratio will first increase then decrease to a stable value when $n_2$ keeps increasing. The increase is due to the UE is permitted to offload more data to the cloud through
the F-AP with the increasing of \( n_2 \). It leads to more data can be computed at the F-AP. However, when the receive capacity of the cloud keeps increasing, more data will be offloaded to the cloud to save charge and the computing ratio of the F-AP decreases. The F-AP's computing ratio will increase to a stable value when \( n_1 \) keeps increasing. This is because when the optimized offloaded data size is larger than the receive capacity of the F-AP, in order to save charge, the UE will offload as much as possible under the limit of the receive capacity of the F-AP. That is why it increases with the increasing of \( n_1 \). When the receive capacity of upper layer is increasing to larger than the optimized allocated data size at corresponding layer, the computing ratios at each layer will stay unchanged with the increasing of receive capacities. In Fig. 6c, the computing ratio of the cloud is complemented with the sum of the UE's and the F-AP's.

Figure 7a, b shows the computing ratios at each layer in mode 2 versus the values of \( n_1 \) and \( n_2 \), where \( N = 2000 \). In Fig. 7a, the computing ratio of the UE decreases when the receive capacities becomes larger. The computing ratio of the cloud is complemented with UE's with the increase of \( n_1 \) and \( n_2 \) as shown in Fig. 7b. The reason is similar to mode 1. This is because when the optimized offloaded data size is larger than the receive capacity of the cloud, in order to save charge, the UE will offload as much as possible under the limit of the receive capacity. When the receive capacity of upper layer further increases to the value which is larger than the optimized allocated data size at the cloud, the computing ratios will stay unchanged with the increasing of receive capacities. Compared with mode 1, the computing burden on the cloud is larger when there is no edge severs.

7 Conclusion

In this paper, a mixed MEC/MCC system based on offloading computing was investigated, which joint optimized the computing ratios at each layer and distribution density of F-APs to minimize the expect charge. To address the non-convex problem, we had proposed ICONP algorithm to solve it. The suboptimal computing ratios of the computing task at each layers were obtained by fixing the value of the density of F-APs and using geometric mean inequality to transform the problem into a convex form. The density of APs was obtained via nonlinear unconstrained programming. The computing ratios and the density of F-APs were solved iteratively. Our simulation results verified that the proposed system
can achieve better performance than computing the whole task locally in respects of the charge and the delay. Actually, the research in our paper does not consider the interference between multi-cell and multi-user which indeed exists in real life. Meanwhile, the cost of calculation of optimization problem is not taken into account. Thus, there are several future directions of interest to pursue based on our work. It is interesting to study the multi-user and multi-M-AP coordinated communication under mobile edge computing to overcome the limitation of our work in this paper. In this case, inter-user interference and multi-user game on resources will be taken into consideration. It is also meaningful to consider the cost when deal with the optimization problem. Meanwhile, machine learning is a hot research topic at present. The way to combine machine learning with mobile edge computing effectively is also worth studying in the future.

**Abbreviations**
Gloss: Meaning; MEC: Mobile edge computing; AP: Access point; UE: User equipment; ICONP: Iteration of convex optimization and nonlinear programming; MCC: Mobile cloud computing; F-AP: Fog computing-based access point; QoS: Quality of service; SNR: Signal-to-noise ratio.

**Acknowledgements**
This work was supported by National Natural Science Foundation of China under Grants 61471120, Natural Science Foundation of Hunan Province under Grants 2020JJ4745, the National Science and Technology Major Project of China under Grant 2018ZX03001002-003.

**Authors’ contributions**
All the authors designed research, design the algorithm, performed research, analyzed data, and wrote the paper. All authors read and approved the final manuscript.

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
1 School of Information Science and Engineering, Southeast University, Southeast University Road, Nanjing 210096, China. 2 School of Computer Science and Engineering, Central South University, Lushan South Road, Changsha 410083, China. 3 The Purple Mountain Laboratories, Mozhou East Road, Nanjing 210096, China.

**Appendix 1**
The probability of choosing mode $i$, $i \in \Omega$, can be derived by

$$F(\lambda_i) = \Pr\left(\frac{p_1|\mathbf{h}_1||L_i|^{-\zeta_f}}{B_1\sigma^2} \geq T_i\right)$$

$$= \int_{0}^{\infty} \Pr\left(|\mathbf{h}_1|^2 \geq \frac{T_iL_i^{2\zeta_f}}{p_1B_1\sigma^2}\right)f(l_i)dl_i$$

$$= \int_{0}^{\infty} \mathbb{E}\left[\exp\left(-\frac{T_iL_i^{2\zeta_f}}{p_1B_1\sigma^2}\right)\right]f(l_i)dl_i$$

$$= \int_{0}^{\infty} \exp\left(-\frac{T_iL_i^{2\zeta_f}}{p_1B_1\sigma^2}\right)f(l_i)dl_i$$

where $B_1$ and $p_1$ are transmission bandwidth and power of the UE. $\sigma^2$ is the mean noise power per Hz. $|\mathbf{h}_1|^2 \sim \exp(1)$ characterize the exponentially distributed fading power over the flat Rayleigh fading channel between the UE and the selected AP. $||L_i||^{-\zeta_f}$ denotes the path loss of mode $i$ and $\zeta_f$ is the path loss exponent, where $L_i$ is the distance between the UE and the selected AP. $f(l_i) = 2\lambda_i \pi l_i e^{-\lambda_i \pi l_i^2}$ is the probability density function (PDF) of the distance between UE and the nearest AP [25]. A closed form...
expression can be expressed as \( F(\lambda_i) = \frac{1}{1+\xi_iB_1\sigma^2/(p_1\lambda_i\pi)} \) when \( \zeta_f = 1 \). In this paper, we analyze the problem base on \( \zeta_f = 1 \).

**Appendix B**

For a positive continuous random variable \( A \),
\[
E\left[ A \mid A \geq W \right] = W \Pr(A \geq W) + \int_W^\infty \Pr(A \geq a) da \quad [25].
\]
Thus, the transmission rate between the UE and the selected AP in mode \( i, i \in \Omega \) is derived as bellow.

\[
R_i = \mathbb{E}[B_1 \log_2(1 + \text{SNR})|\text{SNR} \geq T_i] = \mathbb{E}[B_1 \log_2(1 + \text{SNR})|1 + \text{SNR} \geq 1 + T_i]
= B_1 \int_0^\infty \log_2(1 + T_i) F(\lambda_i|T_i = 2^\theta) d\theta + B_1 \log_2(1 + T_i) F(\lambda_i)
= B_1 \left[ \rho(\lambda_i) + \log_2(1 + T_i) \frac{p_1\lambda_i\pi}{p_1\lambda_i\pi + T_iB_1\sigma^2} \right]
\]
where \( \rho(\lambda_i) = \int_0^\infty \log_2(1 + T_i) \frac{p_1\lambda_i\pi}{p_1\lambda_i\pi + T_iB_1\sigma^2} d\theta \).

**Appendix C**

The expect distance between the selected AP and the cloud can be expressed as \( X_i = \int_0^\infty \log_2(l_i) dl_i, i \in \Omega \). Suppose the distance between the UE and the cloud is \( H_{uc} \) and the schematic plan of three points is in Fig. 8:

where point \( R \) is the location of the selected AP, \( X_i, i \in \Omega \), is the expect distance between the selected AP and the UE. The expect distance between the AP and the cloud is calculated as the average distance between the point \( R \) and the point \((H_{uc}, 0)\).

\[
D_i = \mathbb{E}[H_{uc}R] = \frac{1}{2\pi} \int_0^{2\pi} \sqrt{(X_i \cos \theta - H_{uc})^2 + X_i \sin^2 \theta} d\theta
= \frac{1}{2\pi} \int_0^{2\pi} \sqrt{H_{uc}^2 + X_i^2 - 2X_iH_{uc} \cos \theta} d\theta
\]
Thus the transmission rate from AP to the cloud is

\[
r_i = \mathbb{E} \left[ B_2 \log_2 \left( 1 + \frac{p_2\|b_2\|^2||D_i||^{-2\zeta_c}}{B_2\sigma^2} \right) \right] = B_2 \log_2 \left( 1 + \frac{p_2\mathbb{E}[\|b_2\|^2]||D_i||^{-2\zeta_c}}{B_2\sigma^2} \right)
\]

![Fig. 8 Plan sketch of distances among each point](image-url)
where $|b_2|^2 \sim \text{exp}(1)$ characterize the exponentially distributed fading power over the flat Rayleigh fading channel between the AP and the cloud, $\mathbb{E}[|b_2|^2] = 1$, $||D_i||^{-\zeta}$ denotes the path loss, $B_2 \sigma^2$ represents the noise power received by the Cloud. Thus, $r_i = B_2 \log_2(1 + \frac{P_0 ||D_i||^{-\zeta}}{B_2 \sigma^2}), i \in \Omega$.

Received: 15 May 2018  Accepted: 6 January 2021
Published online: 25 January 2021
30. R. Liu, *Mathematical Modeling Method and Mathematical Experiment* (China Water Conservancy and Hydropower Press, Beijing, 2011)
31. B. Stephen, V. Lieven, *Convex Optimization* (Cambridge University Press, Britain, 2004)

**Publisher’s Note**
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.