Nonlinear Pricing Based Distributed Offloading in Multi-User Mobile Edge Computing

Bizheng Liang, Rongfei Fan, *Member, IEEE*, Han Hu, *Member, IEEE*, Yu Zhang, Ning Zhang, *Senior Member, IEEE*, and Alagan Anpalagan, *Senior Member, IEEE*

***Abstract*—Mobile edge computing (MEC) has emerged as a promising solution to alleviate mobile devices' (MUs') computational burden by offloading part or all of their computational tasks to a nearby edge server. To promote the wide deployment of MEC, charging the MUs and rewarding the edge server is a good mechanism to motivate the edge server to offer computing service. Current related literature usually assume a linear pricing strategy, in which the unit price in time is a linear function with the served computing capability. In this work, however, we explore the nonlinear pricing strategy for the first time, as the cost of a CPU presents super-linear feature with the computing capability. A MEC system with multiple MUs and a single edge server is considered and a two-level Stackelberg game with the leader being the edge server and the followers being MUs is formulated, whereby edge server's revenue is maximized in the upper level via optimizing the nonlinear pricing function while the defined cost function of individual MUs is minimized by deciding the amount of data for offloading and computing capability to purchase from the edge server in the lower level. Through analysis, steps of transformations, and relaxation, closed-form optimal solution for lower-level problem is derived, and the solution for upper-level problem is presented although it is non-convex. Numerical results verify the superiority of our proposed pricing strategy over traditional linear pricing strategy.**

***Index Terms*—Mobile edge computing (MEC), Stackelberg game, nonlinear pricing, offloading management.**

I. Introduction

With the proliferation of mobile applications, the tasks to process by the mobile devices (MUs) is becoming computationally intensive. To alleviate the computational burden of MUs, mobile edge computing (MEC) has emerged as a promising solution, which permits a MU to offload part or all of its data for computing to a nearby edge server [1], [2]. In practise, a MEC system is usually composed of a single edge server serving multiple MUs, in which the configuration of every MU's parameters, including the amount of data to offload, communication resources (including transmit power and bandwidth), and computing resources (the allocated computing capability at MU and the edge server), are explored in a centralized way [3]–[6] or distributed way [7]–[9]. For ease of presentation, the above two ways are called as centralized and distributed data offloading, respectively.

In centralized data offloading, the edge server collects all necessary information of each MU and makes the decision accordingly, to improve the performance in terms of MUs' energy consumption [3], latency of task completion [4], or a combination of both [5], [6]. To achieve these research goals, joint optimization of the amount of data for offloading with communication resources [3], computing resources [5], or both [6] is performed.

In distributed data offloading, every MU makes decisions independently and reaches a data offloading agreement spontaneously. Distinguished from centralized data offloading, every MU has the freedom to make decisions themselves based on their needs. In the framework of distributed data offloading, economic incentive plays an important role to promote the deployment of MEC, which allows the edge server to charge every MU for their received computing service [2]. In this regard, how to price the computing service becomes a key issue for the edge server. Considering different interests of parties in MEC systems, the model of Stackelberg game is generally introduced to study the interaction between the edge server and multiple MUs [2], [7]–[9]. In the upper level of Stackelberg game, the leader (i.e., the edge server) aims at maximizing its revenue function or profit function (defined by revenue minus service cost) through adjusting its pricing function, while in the lower level every follower (i.e., every MU) desires to minimize its cost function or utility function via optimizing its parameters individually for a given pricing function generated in the upper level. With such a framework, in [7], [8], the revenue of the edge server is maximized. Specifically, the cost of every MU, defined as payment plus latency, is minimized in [7], and the utility of every MU, defined as saved energy from computation offloading minus payment, is maximized in [8]. Different from [7] and [8], authors in [9] investigate a MEC system for mining in blockchains, in which multiple MUs require mining service from one edge server. A Stackelberg game is also formulated, where the leader maximizes its profit and every follower maximizes its defined utility through mining.

In the existing works [7]–[9], they all adopt a linear pricing function, in which the payment from a MU to the edge server in unit time is a linear function with the ordered computing capability. However, this may not be accurate or optimal due to the fact that the price of a CPU generally grows super-linearly with its frequency and number of cores (which represent the CPU's computing capability) in consumer electronic market [10]. In this paper, to capture this non-linear feature, we explore a totally different pricing strategy, i.e., nonlinear pricing strategy, for distributed offloading for the first time.

Manuscript received September 7, 2020; revised November 28, 2020; accepted December 13, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61601025, Grant 61971457 and Grant 61771054. The review of this article was coordinated by Dr. Yuanxiong Guo. *(Corresponding author: Rongfei Fan.)*

B. Liang, R. Fan, Han Hu, and Yu Zhang are with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, P. R. China (email: {liangbizheng, fanrongfei, hhu, yuzhang}@bit.edu.cn).

N. Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, N9B 3P4, Canada (ning.zhang@uwindsor.ca).

A. Anpalagan is with the Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto, ON, M5B 2K3, Canada (alagan@ee.ryerson.ca).

The framework of single-leader multiple-follower Stackelberg game is adopted. In the lower level, the cost of every MU is minimized and the closed-form optimal solution of every MU's parameters is derived, including the amount of data to offload and the amount of computing capability to order, for a given pricing function of the edge server. In the upper level, parameterized pricing function is optimized so as to maximize the edge server's revenue, which turns to be a non-convex optimization problem. With monotonic optimization, optimal solution of relaxed optimization problem is obtained. Numerical results validate and demonstrate the advantage of our proposed nonlinear pricing strategy and resource allocation scheme over existing linear pricing strategy.

II. System Model and Problem Formulation

Consider a MEC system with single edge server and K multiple MUs, which constitutes the set $\mathcal{K} \triangleq \{1, 2, ..., K\}$. Every MU has a computational task to complete with the aid of edge server, where the whole or part of data for computing can be offloaded from the MU to the edge server, while the rest of data is computed locally. After receiving the offloaded data, the edge server will complete the computing and feedback the computed results to the associated MU.

For the computational task of kth MU, $k \in \mathcal{K}$, there are two kernel characterizing parameters, R_k and C_k. R_k indicates the amount of data for computing and C_k denotes the amount of computation (in unit of CPU cycles) for computing one bit data of the task. Hence, the total computation amount of kth MU is $C_k R_k$ for $k \in \mathcal{K}$.

For computing, suppose kth MU has a computing capability F_k (in unit of CPU cycles/second) and the edge server would like to lease a partial of its computing capability f_k^c to kth MU (also in unit of CPU cycles/second). Suppose the total computation capacity of the edge server is f_C, then we have $\sum_{k\in\mathcal{K}} f_k^c \leq f_C$. Define $P(f_k^c)$ as the price kth MU has to pay in unit time (in terms of second) to the edge server for using a computational resource with capability f_k^c. Enlightened by the fact that the price of CPU grows super-linearly with its computing capability [10] and recalling that $P(\cdot)$ should be zero when no computing service is offered, we assume that the function $P(\cdot)$ should satisfy the following three properties: 1) $P(0) = 0$; 2) $P(\cdot)$ is an increasing function; and 3) $P(\cdot)$ is a convex function. In this work, to facilitate the discussion, we utilize a quadratic function to approximate $P(\cdot)$ by setting $P(x) = ax^2 + bx$. It can be checked that the required three properties of $P(x)$ can be fulfilled when $a \geq 0$ and $b \geq 0$ [1]. Denote t_k^c as the consumed time of kth MU for computing at edge server and u_k^p as the amount of payment the kth MU will give to the edge server, then we have

$$u_k^p = P(f_k^c) \cdot t_k^c, \forall k \in \mathcal{K}. \tag{1}$$

[1]The reason that we select quadratic function to approximate $P(\cdot)$ are threefold: 1) The quadratic function is in coordination with the trend shown in [10]; 2) The quadratic function is general since it also contains the case that the pricing model is linear, which is achieved when $a = 0$; 3) The quadratic function can make the optimization of the nonlinear pricing function $P(\cdot)$ tractable as shown in Section III-B.

On the other hand, to limit the computational burden of edge server, the total quantity of computation should be upper bounded [7], which can be given as

$$\sum_{k\in\mathcal{K}} C_k l_k \leq \bar{F}, \tag{2}$$

where $l_k \in [0, R_k]$ is the amount of data to offload for $k \in \mathcal{K}$ and $\bar{F}$ is a predefined threshold.

For offloading data and downloading computational results, with l_k defined for $k \in \mathcal{K}$, the amount of data to compute locally is $(R_k - l_k)$ and denote the amount of computational result to return as $\alpha_k l_k$, where $\alpha_k > 0$ [2].

Offloading and downloading are completed through wireless transmission. Suppose the system bandwidth is B and is equally shared by these K MUs. In other words, each MU has an exclusively allocated channel with bandwidth $\bar{B} = B/K$. The channel gains are assumed to be static in one fading block, which can cover the timespan for one round of edge computing, but are randomly distributed over fading blocks. This assumption is reasonable in slow-fading environment and is general in literature [8]. In addition, the channel gains are assumed to be reciprocal, i.e., the uplink channel gain and downlink channel gain for kth MU are identical, denoted as h_k for $k \in \mathcal{K}$. Assume the transmit power of uplink and downlink for kth MU are p_k and $p_{B,k}$, then the uplink transmission rate and downlink transmission rate can be written as $r_k = \bar{B}\log_2\left(1 + \frac{p_k h_k}{\bar{B}N_0}\right)$ and $r_{B,k} = \bar{B}\log_2\left(1 + \frac{p_{B,k} h_k}{\bar{B}N_0}\right)$ respectively, where N_0 is the power spectrum density of noise.

Time delay is a critical performance measure in the MEC system. Denote t_k^{tot} as the total processing time to complete the computational task for kth MU. Then t_k^{tot} can be expressed as the maximum between the consumed time for local computing and the consumed time for the computing through edge server, which can be denoted as t_k^{loc} and t_k^{off}, respectively. In other words, there is $t_k^{\text{tot}} = \max\left(t_k^{\text{loc}}, t_k^{\text{off}}\right)$ for $k \in \mathcal{K}$. With previous definitions, t_k^{loc}=$(R_k - l_k)C_k/F_k$. For t_k^{off}, it can be written as $t_k^{\text{off}} = t_k^u + t_k^c + t_k^d$, where t_k^u and t_k^d are the consumed time for offloading data and downloading computational results, respectively. Note that $t_k^u = l_k/r_k$, $t_k^c = C_k l_k / f_k^c$, and $t_k^d = \alpha_k l_k / r_{B,k}$ for $k \in \mathcal{K}$.

From the perspective of kth MU for $k \in \mathcal{K}$, the cost function, defined as $U_k(l_k, f_k^c)$ for given l_k and f_k^c, is composed of two parts: time delay for completing its task, i.e., t_k^{tot}, and the amount of money kth MU has to pay to the edge server, i.e., u_k^p. Hence, the cost function is given as

$$U_k(l_k, f_k^c) = w_p u_k^p + w_d t_k^{\text{tot}} = w_p P(f_k^c) t_k^c + w_d t_k^{\text{tot}}, \forall k \in \mathcal{K}, \tag{3}$$

where w_p and w_d are non-negative weights associated with u_k^p and t_k^{tot}, respectively. The $U_k(l_k, f_k^c)$ can be further written as

$$U_k(l_k, f_k^c) = \begin{cases} w_p P(f_k^c) t_k^c + w_d \frac{(R_k - l_k)C_k}{F_k}, & t_k^{\text{loc}} \geq t_k^{\text{off}}, \\ w_p P(f_k^c) t_k^c + w_d \left[t_k^c + l_k\left(\frac{1}{r_k} + \frac{\alpha_k}{r_{\text{B,k}}}\right)\right], & t_k^{\text{loc}} < t_k^{\text{off}}. \end{cases} \tag{4}$$

[2]Generally speaking, there would be more output data when more amount of computation is involved for an application. To reflect this rule, we assume the amount of computation result in proportion with the amount of data for offloading and this assumption can be always found in literature [11].

From the perspective of the edge server, the total revenue for given pricing function $P(\cdot)$, or say a and b, can be written as

$$Y(a,b)=\sum_{k\in\mathcal{K}}u_k^p=\sum_{k\in\mathcal{K}}P(f_k^c)\cdot t_k^c. \tag{5}$$

With every MU's cost function and edge server's revenue defined, kth MU for $k\in\mathcal{K}$ can determine the amount of data for offloading l_k and the amount of computing capability f_k^c by minimizing its cost function $U_k(l_k,f_k^c)$ for given a and b, while the edge server can select a set of a and b to maximize its utility function $Y(a,b)$. A Stackelberg game can be formulated and two levels of optimization problems can be formulated respectively. In the lower level, the cost function of kth MU for $k\in\mathcal{K}$ needs to be minimized with given a and b, which can be written as

Problem 1:

$$\begin{aligned}\min_{l_k,f_k^c}\quad & U_k(l_k,f_k^c)\\ \text{s.t.}\quad & 0\le l_k\le R_k,\\ & f_k^c\ge 0.\end{aligned}$$

In the upper level, edge server's revenue needs to be maximized, which is given as

Problem 2:

$$\begin{aligned}\max_{a\ge 0,b\ge 0}\quad & Y(a,b)\\ \text{s.t.}\quad & \sum_{k\in\mathcal{K}}C_k l_k^*(a,b)\le \bar{F},\\ & \sum_{k\in\mathcal{K}}f_k^c(l_k^*(a,b))\le f_C,\end{aligned}$$

where $l_k^*(a,b)$ and $f_k^c(l_k^*(a,b))$ are the optimal solution of l_k and f_k^c for Problem 1, $\forall k\in\mathcal{K}$ [3] [4].

III. Optimal Solution

A. Optimal Solution of Problem 1

In order to solve Problem 1, we first rewrite the objective function of Problem 1 by expressing f_k^c with t_k^c according to $t_k^c=C_kl_k/f_k^c$, then we have $P(f_k^c)\cdot t_k^c=P\left(\frac{C_kl_k}{t_k^c}\right)\cdot t_k^c$ for $k\in\mathcal{K}$. For ease of expression in the following, define $g(x,d)\triangleq xP(d/x)$ where $d>0$. Hence, the objective function of Problem 1, $U_k(l_k,f_k^c)$, can be rewritten as $U_k(l_k,f_k^c)=U_k(l_k,t_k^c)=w_pg(t_k^c,C_kl_k)+w_dt_k^{\text{tot}}$ for $k\in\mathcal{K}$. We have the following Lemma 1 for function $g(x,d)$.

Lemma 1: $g(x,d)$ is monotonically decreasing with respect to x for $x\ge 0$.

Proof: Recalling that $P(\cdot)$ is a convex function and $P(0)=0$, then for $d/x\ge 0$, there is

$$\begin{aligned}& P(d/x)+P'(d/x)(0-d/x)\le P(0)\\ \Leftrightarrow\quad & P(d/x)\le (d/x)P'(d/x).\end{aligned} \tag{6}$$

Combining (6) and checking the first-order partial derivative of $g(x,d)$ with x, there is $\frac{\partial g(x,d)}{\partial x}=P(d/x)-(d/x)P'(d/x)\le 0$, which indicates the decreasing monotonicity of $g(x,d)$ with x. ∎

With Lemma 1, it is clear that $g(t_k^c,C_kl_k)$ is monotonically decreasing with t_k^c for $t_k^c\ge 0,\forall k\in\mathcal{K}$. Then some property of Problem 1's optimal solution can be given as follows.

Lemma 2: The optimal l_k for $k\in\mathcal{K}$ of Problem 1 satisfies

$$t_k^c=\frac{C_kR_k}{F_k}-\beta_kl_k,\forall k\in\mathcal{K}, \tag{7}$$

where $\beta_k=\frac{C_k}{F_k}+\frac{1}{r_k}+\frac{\alpha_k}{r_{\text{B,k}}}$ for $k\in\mathcal{K}$, and l_k falls into the interval

$$0\le l_k\le\frac{C_kR_k}{F_k\beta_k},\forall k\in\mathcal{K}. \tag{8}$$

Proof: According to (3), (4), and Lemma 1, $U_k(l_k,t_k^c)$ is decreasing with t_k^c when $t_k^{\text{loc}}\ge t_k^{\text{off}}$, and is increasing with l_k when $t_k^{\text{loc}}<t_k^{\text{off}}$. Hence, the minimal $U_k(l_k,t_k^c)$ is achieved when t_k^c increases to its maximum if $t_k^{\text{loc}}\ge t_k^{\text{off}}$, and l_k decreases to its minimum if $t_k^{\text{loc}}<t_k^{\text{off}}$, which happens exactly when $t_k^{\text{loc}}=t_k^{\text{off}}$. This can lead to the holding of (7) recalling that $t_k^{\text{loc}}=\frac{(R_k-l_k)C_k}{F_k}$ and $t_k^{\text{off}}=t_k^u+t_k^c+t_k^d$. With the expression of (7) and by imposing both t_k^c and l_k to be no less than 0, it can be easily derived that $0\le l_k\le\frac{C_kR_k}{F_k\beta_k}$ for $k\in\mathcal{K}$. ∎

Remark: With Lemma 2 and the equality $t_k^c=C_kl_k/f_k^c$, we can deduce that $f_k^c(l_k^*(a,b))=\frac{C_kl_k^*(a,b)}{\frac{C_kR_k}{F_k}-\beta_kl_k^*(a,b)}$, which is a monotonically increasing function with $l_k^*(a,b)$.

According to the above analysis, t_k^c and f_k^c can be expressed by l_k, hence $U_k(l_k,t_k^c)$ can be further simplified as a single-variable function with l_k, which can be written as follows

$$\begin{aligned}U_k(l_k)&=w_pP\left(\frac{C_kl_k}{\frac{C_kR_k}{F_k}-\beta_kl_k}\right)\left(\frac{C_kR_k}{F_k}-\beta_kl_k\right)+w_d\frac{(R_k-l_k)C_k}{F_k}\\&=C_k\left[\frac{aw_pC_kF_kl_k^2}{C_kR_k-F_k\beta_kl_k}+\left(bw_p-\frac{w_d}{F_k}\right)l_k+\frac{w_dR_k}{F_k}\right],\forall k\in\mathcal{K}.\end{aligned}$$

and Problem 1 dwells into the following form

Problem 3:

$$\begin{aligned}\min_{l_k}\quad & U_k(l_k)\\ \text{s.t.}\quad & \text{Constraint }(8).\end{aligned}$$

For Problem 3, the optimal solution can be given in the following lemma.

Lemma 3: The optimal l_k of Problem 3 for $k\in\mathcal{K}$ for given a and b, denoted as $l_k^*(a,b)$, can be written as (9), which is given on the top of next page.

Proof: To find the optimal l_k, we need to set $\frac{\partial U_k(l_k)}{\partial l_k}$ to be zero and make sure that $\frac{\partial^2U_k(l_k)}{\partial l_k^2}\ge 0$. By setting $\frac{\partial U_k(l_k)}{\partial l_k}=0$, there is

$$l_k=\frac{C_kR_k}{F_k\beta_k}\cdot\left[1\pm\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}\right]. \tag{10}$$

According to (8), $l_k\le\frac{C_kR_k}{F_k\beta_k}$, hence the case that $l_k=\frac{C_kR_k}{F_k\beta_k}\cdot\left[1+\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}\right]>\frac{C_kR_k}{F_k\beta_k}$ should be discarded. To guarantee the holding of the other case that $l_k=\frac{C_kR_k}{F_k\beta_k}\cdot\left[1-\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}\right]$, we need to

[3] Intuitively, the optimal f_k^c of Problem 1 for $k\in\mathcal{K}$ should be a function with a and b. According to the discussion in Section III-A, the optimal f_k^c is actually a function of $l_k^*(a,b)$, and is expressed as $f_k^c(l_k^*(a,b))$ here.

[4] In addition to the second constraint of Problem 2, the first constraint of Problem 2, i.e., Eqn. (2), also imposes a restriction on edge server's computation capability. Actually this is to facilitate the comparison with [7] under the same condition, which is the only work we can make a comparison with in literature.

$$l_k^*(a,b)=\begin{cases}\frac{C_kR_k}{F_k\beta_k}\left[1-\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}\right], & 0\le b\le\frac{w_d}{w_pF_k},\\ 0, & \text{otherwise.}\end{cases} \tag{9}$$

require that $(w_d-bw_pF_k)\ge 0$, which will be satisfied when $b\le\frac{w_d}{w_pF_k}$. In the case that $b\le\frac{w_d}{w_pF_k}$, we also need to check the sign of $\frac{\partial^2U_k(l_k)}{\partial l_k^2}$. Also with $l_k\le\frac{C_kR_k}{F_k\beta_k}$ according to (8), there would be $\frac{\partial^2U_k(l_k)}{\partial {l_k}^2}=\frac{2aw_pC_k^4F_kR_k^2}{(C_kR_k-F_k\beta_kl_k)^3}\ge 0$.

For the other case, i.e., $b>\frac{w_d}{w_pF_k}$, with $0\le l_k\le\frac{C_kR_k}{F_k\beta_k}$, $\frac{\partial U_k(l_k)}{\partial l_k}=\left(bw_p-\frac{w_d}{F_k}\right)C_k+\frac{aw_pC_k^2F_kl_k(2C_kR_k-F_k\beta_kl_k)}{(C_kR_k-F_k\beta_kl_k)^2}>0$, which indicates that the minimum $U_k(l_k)$ is achieved at $l_k=0$. ■

B. Solution of Problem 2

In what follows, we will solve Problem 2. With $l_k^*(a,b)$ derived in Lemma 3 and t_k^c and f_k^c expressed by $l_k^*(a,b)$ as the way shown in Section III-A, the function $u_k^p=P(f_k^c)\cdot t_k^c$ becomes a function of a and b, i.e.,

$$u_k^p(a,b)=\left[\frac{aC_kF_kl_k^*(a,b)}{C_kR_k-F_k\beta_kl_k^*(a,b)}+b\right]\cdot l_k^*(a,b)\cdot C_k,\forall k\in\mathcal{K}. \tag{11}$$

However, with the expression of $u_k^p(a,b)$ in (11) and the expression of $l_k^*(a,b)$ in (9), both the objective function and constraint's left-hand side function of Problem 2 are non-convex, which is challenging to find the optimal solution.

To solve Problem 2, we first approximate the objective function of Problem 2 by its lower bound. Specifically, by utilizing the Jensen inequality [12], there is

$$\begin{aligned}Y(a,b)=\textstyle\sum_{k\in\mathcal{K}}u_k^p(a,b) &\ge K\left(\textstyle\prod_{k\in\mathcal{K}}u_k^p(a,b)\right)^{\frac{1}{K}}\\ &=K\left(e^{\sum_{k\in\mathcal{K}}\ln\left(u_k^p(a,b)\right)}\right)^{\frac{1}{K}}.\end{aligned} \tag{12}$$

In the following, we will maximize the lower bound of $Y(a,b)$ as indicated in (12) instead. For the ease of presentation, define $Z_k(a,b)=\left[\frac{aC_kF_kl_k^*(a,b)}{C_kR_k-F_k\beta_kl_k^*(a,b)}+b\right]$, hence $u_k^p(a,b)=Z_k(a,b)l_k^*(a,b)C_k$. Then combing the constraint in (8), $Y(a,b)$'s lower bound maximization problem can be rewritten as [5]

Problem 4:

$$\begin{aligned}\max_{a,b}\quad & \sum_{k\in\mathcal{K}}\ln(Z_k(a,b))+\ln(l_k^*(a,b))\\ \text{s.t.}\quad & \sum_{k\in\mathcal{K}}C_kl_k^*(a,b)\le\bar{F}, && \text{(13a)}\\ & \sum_{k\in\mathcal{K}}f_k^c(l_k^*(a,b))\le f_{\mathrm{C}}, && \text{(13b)}\\ & 0\le l_k^*(a,b)\le\frac{C_kR_k}{F_k\beta_k},\forall k\in\mathcal{K}, && \text{(13c)}\\ & a\ge 0,b\ge 0. && \text{(13d)}\end{aligned}$$

[5]For Problem 2, both the objective function $Y(a,b)$ and the left-hand side functions of its constraints are non-convex, which impose a big challenge on solving it. By approximating $Y(a,b)$ by its lower bound, we can find a way to work out the global optimal solution for maximizing the lower bound of $Y(a,b)$ in this paper, which can also promise the performance of $Y(a,b)$. Moreover, numerical results in Section IV can verify that the achieved revenue of the edge server by optimizing the approximated problem can still outperform the one under linear pricing strategy, which already indicates the achievement of our design goal, i.e., to testify the advantage of nonlinear pricing over linear pricing.

For function $Z_k(a,b)$, there is the following property.

Lemma 4: $Z_k(a,b)$ is a monotonically non-decreasing function with a and a monotonically increasing function with b for $a\ge 0$, $b\ge 0$, and $k\in\mathcal{K}$.

Proof: We first consider the region that $(w_d-bw_pF_k)\ge 0$. For the monotonicity of $Z_k(a,b)$ with a, investigate the first-order and second-order partial derivatives of $Z_k(a,b)$ with a, there are $\frac{\partial Z_k(a,b)}{\partial a}=\frac{(w_d-bw_pF_k)\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}}{2aw_pF_k}+\frac{C_k\left[-1+\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}\right]}{\beta_k}$, and $\frac{\partial^2Z_k(a,b)}{\partial a^2}=-\frac{C_k\beta_k(w_d-bw_pF_k)^2}{4a\sqrt{aw_pC_kF_k}[aw_pC_kF_k+\beta_k(w_d-bw_pF_k)]^{\frac{3}{2}}}$. It holds that with $(w_d-bw_pF_k)\ge 0$ and $a\ge 0$, there always exists $\frac{\partial^2Z_k(a,b)}{\partial a^2}\le 0$ which indicates $\frac{\partial Z_k(a,b)}{\partial a}$ keep on decreasing with a. In addition, $\lim_{a\to 0^+}\frac{\partial Z_k(a,b)}{\partial a}=\infty$ and $\lim_{a\to\infty}\frac{\partial Z_k(a,b)}{\partial a}=0$. Hence, $\frac{\partial Z_k(a,b)}{\partial a}$ is alway larger than 0 for $a\ge 0$, which proves the increasing monotonicity of $Z_k(a,b)$ with a when $(w_d-bw_pF_k)\ge 0$. In terms of the monotonicity of $Z_k(a,b)$ with b, the first-order partial derivation of $Z_k(a,b)$ with respect to b is

$$\frac{\partial Z_k(a,b)}{\partial b}=1-\frac{1}{2}\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}, \tag{14}$$

which is no less than $1/2$ when $(w_d-bw_pF_k)\ge 0$.

When $(w_d-bw_pF_k)\le 0$, it can be checked that $Z_k(a,b)$ is a constant b and is continuous at $(w_d-bw_pF_k)=0$. In this case, $Z_k(a,b)$ is definitely non-decreasing with a and is increasing with b. Combing the discussion for the cases that $(w_d-bw_pF_k)\ge 0$ and $(w_d-bw_pF_k)\le 0$, $Z_k(a,b)$ is non-decreasing with a and increasing with b for $a\ge 0$, $b\ge 0$, and $k\in\mathcal{K}$. ■

Lemma 5: $l_k^*(a,b)$ is monotonically non-increasing with a and b for $a\ge 0$, $b\ge 0$, and $k\in\mathcal{K}$.

Proof: The first-order partial derivation of $l_k^*(a,b)$ with a and b, are given as follows:

$$\frac{\partial l_k^*(a,b)}{\partial a}=-\frac{R_k(w_d-bw_pF_k)}{2w_pa^2F_k^2}\left[\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}\right]^{\frac{3}{2}}, \tag{15}$$

$$\frac{\partial l_k^*(a,b)}{\partial b}=-\frac{R_k}{2aF_k}\left[\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}\right]^{\frac{3}{2}}. \tag{16}$$

When $(w_d-bw_pF_k)\ge 0$, $\sqrt{\frac{aw_pC_kF_k}{aw_pC_kF_k+\beta_k(w_d-bw_pF_k)}}\in(0,1]$ for $a\ge 0$, and both $\frac{\partial l_k^*(a,b)}{\partial a}$ and $\frac{\partial l_k^*(a,b)}{\partial a}$ are no greater than 0. When $(w_d-bw_pF_k)\le 0$, $l_k^*(a,b)=0$ according to (9) and $l_k^*(a,b)$ is continuous at $(w_d-bw_pF_k)=0$. Thus $l_k^*(a,b)$ is monotonically non-increasing with a and b for $a\ge 0$, $b\ge 0$, and $k\in\mathcal{K}$. ■

Defining $F(a,b)=\sum_{k\in K}\ln Z_k(a,b)$ and $G(a,b)=-\sum_{k\in K}\ln l_k^*(a,b)$, Problem 4 can be written as the following equivalent form.

Problem 5:

$$\max_{a,b} \quad F(a,b) - G(a,b)$$
$$\text{s.t.} \quad \text{Constraint } (13a) \text{ – Constraint } (13d),$$

which is further equivalent with the following one

Problem 6:

$$\max_{a,b} \quad F(a,b) + t$$
$$\text{s.t.} \quad t + G(a,b) \leq G(a_{\max}, b_{\max}), \tag{18a}$$
$$\text{Constraint } (13a) \text{ – Constraint } (13d), \tag{18b}$$
$$t \geq 0, \tag{18c}$$

where $a_{\max}$ and $b_{\max}$ represent a number large enough to make $Y(a,b)$ to be zeros [6]. The equivalence between Problem 5 and Problem 6 can be explained as follows: 1) With Lemma 5, $l_k^*(a,b)$ is non-increasing function with a and b for $k \in \mathcal{K}$. Hence, constraints (13a) and (13c) only impose lower bound on a and b [7]. 2) With the remark in Section III-A and Lemma 5, $f_k^c(l_k^*(a,b))$ is a non-increasing function with a and b for $k \in \mathcal{K}$. Hence, constraint (13b) imposes another lower bound on a and b. 3) With Lemma 4 and Lemma 5, both function $F(a,b)$ and function $G(a,b)$ are non-decreasing with a and b. Therefore, to achieve the maximal utility of Problem 6, a, b, and t should be as large as possible, in which case the constraint (18a) becomes active considering that the constraints (13a), (13b) and (13c) do not impose upper bound on a or b. The activeness of constraint (18a) indicates that $t = G(a_{\max}, b_{\max}) - G(a,b)$. Thus, the objective function of Problem 6 turns to be $F(a,b) - G(a,b) + G(a_{\max}, b_{\max})$, which is equivalent to maximizing $F(a,b) - G(a,b)$.

Problem 6 falls into the standard form of *monotonic optimization problem*, in which the objective function is monotonic and all the constraints can be written as the form that monotonic function is larger or smaller than zero [13]. *Polyblock algorithm* can be utilized to find the global optimal solution (which can guarantee the gap between the achieved utility to global optimal utility to be within predefined $\epsilon > 0$) of a monotonic optimization problem [14]. By following the polyblock algorithm, the detailed procedure for solving Problem 6 is given as in Algorithm 1.

By adopting the polyblock algorithm [8] , the three elements of π_{i^*} are the optimal a, b and t for Problem 6, respectively. To this end, Problem 4 has been solved optimally, whose solution can serve as the solution for Problem 2.

Last but not least, the implementation of the proposed resource allocation strategy is given as follows. The solution of a and b is worked out at the edge server, which is rich in computing capability and is capable of runing polyblock algorithm, while the $l_k^*(a,b)$ is calculated at kth MU for $k \in \mathcal{K}$.

[6]Specifically, $a_{\max}$ can be searched via bi-section method and $b_{\max}$ can be set as $\max_{k\in\mathcal{K}} \left(\frac{w_d}{w_p F_k}\right)$.

[7]Note that the constraint $l_k^*(a,b) \geq 0$ for $k \in \mathcal{K}$ does not impose an upper bound on a or b, since $l_k^*(a,b)$ would be zero even when $a = \infty$ or $b = \infty$.

[8]For the complexity of polyblock algorithm, no exact expression can be found in literature since it is hard to predict how many steps are required before the searching iteration stops. In worst case, the polyblock algorithm has exponential complexity [15]. On the other hand, it can be checked Problem 4, or equivalently Problem 6, is a non-convex optimization problem. No existing algorithm can solve a general non-convex problem in polynomial time and the polyblock algorithm is not an exception either [16].

Algorithm 1 Polyblock Algorithm.

1: Initialize a 3-dimensional point set $\mathcal{S} = \{\boldsymbol{s}_1 = (a_{\max}, b_{\max}, G(a_{\max}, b_{\max}) - G(0,0))^T\}$, where $a_{\max}$ and $b_{\max}$ satisfy the condition of (2) and (8). Select a $\epsilon > 0$.
2: **while** Set $\mathcal{S} \neq \emptyset$ **do**
3: **for** $i \in \{1, 2, ..., |\mathcal{S}|\}$ **do**
4: Calculate λ_i which satisfies $G(\lambda_i \boldsymbol{s}_i(1), \lambda_i \boldsymbol{s}_i(2)) + \lambda_i \boldsymbol{s}_i(3) = G(a_{\max}, b_{\max})$ by a bisection search, and set $\pi_i = \lambda_i \boldsymbol{s}_i$.
5: Find $i^* = \arg\max_{1\leq i\leq|\mathcal{S}|}(F(\pi_i(1), \pi_i(2)) + \pi_i(3))$.
6: For any $\boldsymbol{s}_i \in \mathcal{S}$, if $(F(\boldsymbol{s}_i(1), \boldsymbol{s}_i(2)) + \boldsymbol{s}_i(3)) \leq (F(\pi_{i^*}(1), \pi_{i^*}(2)) + \pi_{i^*}(3)) + \epsilon$, then delete the point from set $\mathcal{S}$.
7: **if** Set $\mathcal{S} \neq \emptyset$ **then**
8: Find $j^* = \arg\max_{1\leq j\leq|\mathcal{S}|}(F(\boldsymbol{s}_j(1), \boldsymbol{s}_j(2)) + \boldsymbol{s}_j(3))$.
9: Caculate λ_{j^*} that satisfies $(G(\lambda_{j^*} \boldsymbol{s}_{j^*}(1), \lambda_{j^*} \boldsymbol{s}_{j^*}(2)) + \lambda_{j^*} \boldsymbol{s}_{j^*}(3)) = G(a_{\max}, b_{\max})$ by a bisection search, and set $\pi_{j^*} = \lambda_{j^*} \boldsymbol{s}_{j^*}$.
10: Generate the point set $\mathcal{S}^\dagger = \{\boldsymbol{s}^\dagger | \boldsymbol{s}^\dagger = \boldsymbol{s}_{j^*} + (\pi_{j^*} - \boldsymbol{s}_{j^*}) \circ \mathbf{1}^n, \forall n \in \{1,2,3\}\}$, where $\circ$ means Hadamard product and $\mathbf{1}^n$ is the 3-dimensional vector with nth element being 1 and the rest elements being 0.
11: Delete $\boldsymbol{s}_{j^*}$ from set $\mathcal{S}$, and add set $\mathcal{S}^\dagger$ into set $\mathcal{S}$.
12: Output the last π_{i^*} before set $\mathcal{S}$ turns to be empty.

IV. Numerical Results

In this section, numerical results are presented to verify the effectiveness of our proposed method. Default system parameters are given as follows. There are 30 MUs in the system, whose C_k, R_k and F_k are uniformly distributed within $[500, 1500]$cycles/bit, $\left[8 \times 10^5, 4 \times 10^6\right]$bits, and $[0.1, 1]$GHz, respectively. $\bar{F} = 6 \times 10^9$ cycles/second and $f_C = 5 \times 10^8$Hz, and the ratio $\alpha_k = 0.2$ for $k \in \mathcal{K}$. The total system bandwidth B is 1MHz, and the channel gain h_k are exponentially distributed with mean being 10^{-6}, which corresponds to Rayleigh distribution. The N_0 is set as -174dBm/Hz, $p_k = 0.1$W and $p_{B,k} = 1$W for $k \in \mathcal{K}$. Both w_p and w_d are set as 1.

Total revenue collected by the edge server and average latency are taken as the main performance metrics. The average latency can be expressed as $\sum_{k\in\mathcal{K}} \frac{(R_k - l_k)C_k}{KF_k}$ according to Lemma 2, which is affected by the amount of offloading from each MU. As a comparison, the performance of the method in [7], which adopts linear pricing method by setting $a = 0$, is also investigated.

In Fig. 1, the total revenue and average latency are plotted versus $\bar{F}$ for our proposed method and linear pricing method. It can be observed that our proposed method can always outperform the linear pricing method, which verifies the effectiveness of our proposed nonlinear pricing strategy and our presented solution for optimizing a and b under nonlinear

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2020.3045473, IEEE Transactions on Vehicular Technology

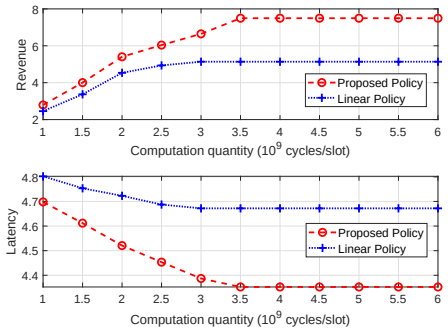

Fig. 1: Performance analysis and comparison versus $\bar{F}$.

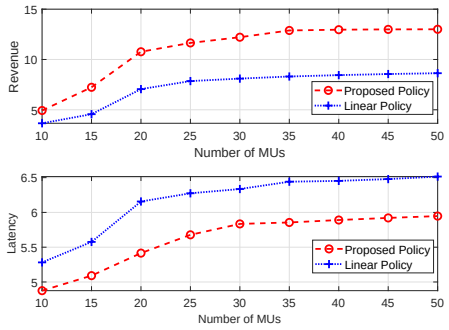

Fig. 2: Performance analysis and comparison versus K.

pricing mechanism. It can be also seen that the total revenue is non-decreasing with $\bar{F}$. This is because larger $\bar{F}$ indicates relaxed feasible region of l_k for $k \in \mathcal{K}$, which can further contribute to non-decreasing total revenue. In addition, it can be also found that the average latency is non-increasing with $\bar{F}$, since larger $\bar{F}$ may promote every MU to offload more data to the edge server, i.e., larger $l_k^*(a,b)$. Hence, the average latency is less according to its definition.

Fig. 2 plots the total revenue and the average latency versus the number of MUs K for our proposed method and the linear pricing method in [7]. It can be seen that our proposed method can always achieve better performance than the linear pricing method, which also verifies the effectiveness of our proposed strategy and associated solution. It can be also seen that as K increases, the total revenue tends to be non-decreasing with K. This can be explained by the fact that larger K permits more affordable MUs to offload data even when they are charged by a higher price, in which case higher total revenue will be achieved. Last but not least, it can be found that the average latency is also non-decreasing with K. The reason is as follows. The computing capability of the edge server is limited. Hence, the offloaded data l_k for $k \in \mathcal{K}$ would be non-increasing when K grows, which will lead to the non-decreasing of average latency according to its definition.

To inspect the complexity of polyblock algorithm, we evaluate the average running time of Algorithm 1. When the number of MUs N are set as 10, 20, 30, 40, 50, the associated average running time is 13.4338, 13.5328, 13.6478, 13.7619, 14.0225 seconds, respectively. It can be seen that the average running time of Algorithm 1 is within 15 seconds even when there are 50 MUs, which would be fully acceptable in real application when this algorithm is implemented on the edge server.

V. Conclusion

In this work, considering the super-linear feature of the cost with the computing capability, the nonlinear pricing policy based distributed offloading is investigated in the MEC system for the first time. A single-leader multi-follower Stackelberg game is formulated, in which the edge server performs as the leader and every single MU performs as a follower. The optimal closed-form solutions of the amount of data to offload and the amount of computing capability to lease are derived for every follower. The parameters of pricing function are optimized so as to maximize the total revenue for the leader through relaxation and monotonic optimization method, although it is non-convex. Numerical results demonstrate the effectiveness of our proposed strategy over traditional linear pricing policy.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322-2358, 4th Quart. 2017.

[2] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 260-288, 1st Quart. 2019.

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

[4] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506-5519, Aug. 2018.

[5] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571-3584, Aug. 2017.

[6] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435-3447, Apr. 2017.

[7] M. Liu and Y. Liu, "Price-based distributed offloading for mobile-edge computing with computation capacity constraints," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 420-423, Jun. 2018.

[8] S. Kim, S. Park, M. Chen, and C. Youn, "An optimal pricing scheme for the energy-efficient mobile edge computation offloading with OFDMA," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1922-1925, Sept. 2018.

[9] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585-4600, Jun. 2019.

[10] Available: https://www.cpubenchmark.net/cpu_value_available.html.

[11] 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, Oct. 2016.

[12] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta mathematica*, vol. 30, no. 1, pp. 175-193, Mar. 1906.

[13] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimization,* 2nd ed. New York: Springer-Verlag, 2009.

[14] R. Fan, H. Jiang, Q. Guo, and Z. Zhang, "Joint optimal cooperative sensing and resource allocation in multichannel cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 722-729, Feb. 2011.

[15] W. Utschick and J. Brehmer, "Monotonic Optimization Framework for Coordinated Beamforming in Multicell Networks," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1899-1909, Apr. 2012.

[16] Y. Zhang, L. Qian, and J. Huang, "Monotonic optimization in communication and networking systems," *Found. Trends Netw.*, vol. 7, no. 1, pp. 1-75, Oct. 2013.