Multi-User Cooperative Computation Framework Based on Bertrand Game

Nan Zhang, Guopeng Zhang, Kezhi Wang and Kun Yang

***Abstract*—In this paper, a multi-user cooperative computing framework is applied to enable mobile users to utilize available computing resources from other neighboring users via direct communication links. An incentive scheme based on Bertrand game is proposed for the user to determine *who* and *how* to cooperate. We model the resource demand users as *buyers* who aim to use minimal payments to maximize energy savings, whereas resource supply users as *sellers* who aim to earn payments for their computing resource provision. A Bertrand game against *buyer's market* is formulated. When the users have *complete information* of their opponents, the Nash equilibrium (NE) of the game is obtained in closed form, while in the case of *incomplete information*, a distributed iterative algorithm is proposed to find the NE. The simulation results verify the effectiveness of the proposed scheme.**

***Index Terms*—Cooperative computation framework, Task offloading, Resource allocation, Bertrand game, Nash equilibrium.**

I. Introduction

With the development of Artificial Intelligence (AI) applications, Virtual Reality (VR) services and others, the computational complexity of mobile applications is increasing rapidly. Although cloud computing can alleviate the shortage of computing resources for user equipments (UEs), it may incur longer transmission delay. To reduce latency and increase user experience, the framework of multi-user cooperative computing has been proposed to enable UEs to offload their computational-intensive tasks to other neighbouring UEs via direct communication links [1].

In the cooperative computing framework, UEs are generally divided into resource demand UEs (DUs) and resource supply UEs (SUs). DUs have heavy computing tasks and may require other devices to assist, while SUs have available resources and may help others. In [2], the authors considered that a DU can partition a computing task into multiple parts and offload the task parts to a set of SUs. The purpose was to maximize the completion rate. The authors in [3] proposed an energy-efficient task offloading method for two cooperative UEs and the aim was to minimize the long-term energy consumption of both users. In [1], the authors proposed a hybrid task offloading framework for *fog computing*, where UEs can choose flexibly from local computing, another device, or the central cloud. In [4], the authors proposed that a SU can make a decision on whether to provide computing service according to the social relationship with a DU.

The provision of computing services to others incurs resource consumption for *rational* providers, who may be only willing to use resources for profit purposes. However, the above works either consider UEs to be fully cooperative [1]-[3][5] or the cooperation depends on their social relations [4]. In this paper, we propose an incentive scheme based on Bertrand game [6] to address the above-mentioned issue. Bertrand game is widely used to analyze price competition behavior and product sales share of *buyer's market*. Compared to existing works, the main contribution of this paper includes:

1) By modeling DUs as *resource buyers* and SUs as *resource sellers*, a Bertrand game is proposed to solve the incentive problem in cooperative computing framework.
2) When the states and actions of all users are observable (known as the *complete information* hypothesis), the NE of the game is solved in closed-form. In the case of *incomplete information*, an effective distributed iterative algorithm is proposed to search the NE of the game.

II. System Model

Consider that there are a set $\mathcal{M} = \{1, \cdots, M\}$ of M UEs and each pair of the UEs can communicate with each other directly. A discrete time model is considered and let $t = 1, 2, \cdots$ denote the consecutive time slots. The duration of a time slot is denoted by T. In any slot t, UE m ($\forall m \in \mathcal{M}$) has a computing task to be executed, which is characterized by a tuple of two parameters $\varphi_m = (L_m, C_m)$, where L_m (in Mb) is the amount of the task input data and C_m (in CPU cycles/Mb) is the required number of CPU cycles to complete each megabit data [3]. Let $f_m^{\max}$ denote the maximum operating frequency of the CPU of UE m. In order to complete task φ_m within a time slot, the CPU frequency is set to $f_m = \frac{C_m L_m}{T}$ ($0 < f_m \leq f_m^{\max}$) by UE m. Let κ_m denote the effective capacitance coefficient of the CPU of UE m. The energy spent by UE m to complete task φ_m is given by [3]

$$E_m^{\text{exe}} = \kappa_m f_m^2 C_m L_m = \frac{\kappa_m \left(C_m L_m\right)^3}{T^2}, \ \forall m \in \mathcal{M}. \tag{1}$$

Nan Zhang and Guopeng Zhang are with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China (e-mail: NanZhang@cumt.edu.cn; gpzhang@cumt.edu.cn).

Kezhi Wang is with the Department of Computer and Information Science, Northumbria University, Newcastle NE2 1XE, U.K. (e-mail: kezhi.wang@northumbria.ac.uk).

Kun Yang is with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, and also affiliated with School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun, China. kunyang@uestc.edu.cn

ACK: Natural Science Foundation of China (Grant No. 61620106011, U1705263), UESTC Yangtze Delta Region Research Grant (Grant No.: 2020D002).

We consider the situation of *buyer's market*. Let $m = 0$ denote the unique DU in the system. The other UEs in set $\mathcal{M} - \{0\}$ are the potential SUs for the DU. We also define a new set $\mathcal{N} = \{1, \cdots, N\} \subseteq \mathcal{M} - \{0\}$ to represent the set of SUs that actually provide computing service to the DU.

A. Energy consumption of the DU

The energy consumption of the DU consists of the following two parts. The first part of energy is used to transmit the task data to the SUs in set $\mathcal{N}$ in slot $t-1$, and the second part of energy is used to execute the remaining task locally in slot t. Note that the above system settings allows task data transmission and task computing to be carried out in parallel.

In slot $t-1$, we assume that any SU i in set $\mathcal{N}$ is allocated equal time for receiving the task data, which is given by

$$t_n = T/\left|\mathcal{N}\right|, \ \forall n \in \mathcal{N}. \tag{2}$$

Let p_0^n denote the transmit power of the DU for uploading task data to SU n. Let g_0^n denote the channel gain from the DU to SU n. Then, the achievable data rate is given by

$$r_0^n = B \log_2 \left(1 + \frac{p_0^n g_0^n}{\sigma^2}\right), \ \forall n \in \mathcal{N}, \tag{3}$$

where B is the channel bandwidth of the system, and σ^2 is the noise power at the receiver of the SU.

The unique DU in the system is denoted by $m = 0$. Then L_0 denotes the amount of task input data of the DU. Let l_0^n $(0 \le l_0^n \le L_0)$ denote the amount of task data that the DU decides to offload to SU n. To ensure correct reception, condition $r_0^n t_n \ge l_0^n$ should be satisfied. By substituting eq. (3) into this condition, the lower bound of p_0^n is obtained as

$$p_0^n = \left(2^{\frac{l_0^n}{BT/|\mathcal{N}|}} - 1\right) \frac{\sigma^2}{g_0^n}, \ \forall n \in \mathcal{N}. \tag{4}$$

Then, the energy spent by the DU for task offloading in slot $t-1$ is given by

$$E_0^{\text{off}} = \sum\nolimits_{n \in \mathcal{N}} p_0^n t_n. \tag{5}$$

Assume that the CPU of the DU always works at the highest frequency $f_0^{\max}$, although the task may be offloaded, it still needs to complete other new tasks. Hence, the energy spent by the DU for executing the remaining task in slot t is

$$E_0^{\text{com}} = \kappa_0 (f_0^{\max})^2 C_0 \left(L_0 - \sum\nolimits_{n \in \mathcal{N}} l_0^n\right). \tag{6}$$

B. Energy consumption of each SU

The energy consumed by any SU n in set $\mathcal{N}$ consists of the following two parts. The first part is used by SU n for receiving the task data of the DU in slot $t-1$. Let p_n^{rec} denote the power of the receiver circuit of SU n. This part of energy consumption is given by

$$E_n^{\text{rec}} = p_n^{\text{rec}} t_n, \ \forall n \in \mathcal{N}. \tag{7}$$

After decoding the task data of the DU, SU n has to raise the CPU frequency in slot t to complete its own task φ_n and the DU's task of l_0^n Mb within slot t. By using eq. (1), we can derive the energy consumption of SU n in slot t as

$$E_n^{\text{com}} = \frac{\kappa_n C_n^3 \left(L_n + l_0^n\right)^3}{T^2}, \ \forall n \in \mathcal{N}. \tag{8}$$

Since the size of the execution result is small, the energy spent by SUs for feeding back the result is negligible.

III. Game Formulation

Our objective is to motivate *rational* UEs to participate in cooperative computing framework. Therefore, the following questions should be answered [7]: 1) *who to cooperate*, i.e., to determine which SUs are in set $\mathcal{N}$, and 2) *how to cooperate*, i.e., to determine how much resource a SU in set $\mathcal{N}$ provides to the DU and how much it can benefit from the provision. Next, we propose a Bertrand game based incentive scheme to address these issues.

In the proposed game, the DU is modeled as a *buyer* who aims to use minimal payment to maximize energy savings. Generally, the benefits of a player in a game is quantified by utilities. Let q_n represent the energy pricing of SU n for providing computing resources to perform 1 Mb computing tasks of the DU. According to [6], the following function is used to quantify the utility of the DU.

$$\begin{aligned} U_0 = \left(E_0^{\text{exe}} - E_0^{\text{com}} - E_0^{\text{off}}\right) - \sum\nolimits_{n \in \mathcal{N}} q_n l_0^n - \\ \frac{1}{2}\left(\sum\nolimits_{n \in \mathcal{N}} \left(l_0^n\right)^2 + 2v \sum\nolimits_{n \ne k} l_0^n l_0^k\right), \end{aligned} \tag{9}$$

where the *first* term represents the energy saving for the DU when it performs cooperative computing (offloading partial computing task to the SUs) rather than performing the entire task locally, the *second* term represents the total payment of the DU to all the selected cooperative SUs, and, thus the difference between the *first* term (the income of the DU) and the *second* term (the cost of the DU) just represents the profit that the DU can obtain under the proposed cooperative computing framework. Additionally, the utility function also takes the resource substitutability (RS) into account in the *third* term as in [6]. RS is an ability of a resource demander to substitute one resource (hold by some oligopolies) with other resources (hold by other oligopolies) of similar functionality [8]. Parameter $v \in [0, 1]$ represents the RS. When $v = 0$, there is no substitutability between resources sold by different oligopolies, when $v = 1$, the resource sold by different oligopolies are completely homogeneous with each other. For example, when a DU is faced with multiple SUs to offload the computing task, if special hardware or software are needed to perform the task, the RS of different SUs is weak, namely v tends to 0; otherwise, the computing task of the DU can be performed in any configured hardware or software system, namely, the computing resources of different SUs have a strong RS, so v tends to 1. The RS has also been used by the authors in [7] to address the spectrum allocation problem in multi-user cognitive radio networks.

The SUs in the game are modeled as *sellers* who aim to not only earn the payment to offset the energy consumption

for resource provision but also gain as much extra profits as possible. Then, the utility function for SU n is defined as

$$U_n = q_n l_0^n - \left(E_n^{\text{rec}} + \left(E_n^{\text{com}} - E_n^{\text{exe}}\right)\right),\ \forall n \in \mathcal{N}, \tag{10}$$

where the *first* term represents the revenue that SU n can receive from the DU, the *second* term represents the extra energy consumed by SU n when providing computing and communication resources to the DU rather than solely performing its own task, and thus, the difference between the *first* term (the income of SU n) and the *second* term (the cost of SU n) just represents the profit that SU n can obtain under the proposed cooperative computing framework.

Let $\pi = \left(l_0^1, \cdots, l_0^{|\mathcal{N}|}\right)$ and $\rho = \left(q_1, \cdots, q_{|\mathcal{N}|}\right)$ denote the strategy profile of the DU and the SUs in the game, respectively. The DU and each of the SUs aim to maximize their utilities in the game by choosing the optimal strategy. Therefore, the objective of the DU is formulated as

$$\max_{\pi}\ U_0 \tag{11}$$
$$\text{s.t.}\quad 0 \le l_0^n \le L_0,\ \forall n \in \mathcal{N}, \tag{11.1}$$
$$\sum\nolimits_{n\in\mathcal{N}} l_0^n \le L_0, \tag{11.2}$$
$$p_0^n \le P,\ \forall n \in \mathcal{N}, \tag{11.3}$$

where constraint (11.2) ensures that the amount of the offloaded data does not exceed the total amount of task data of the DU in any slot t, and constraint (11.3) ensures that the transmit power of the DU does not exceed the maximum allowable power P. The objective of SU n is formulated as

$$\max_{q_n}\ U_n,\ \forall n \in \mathcal{N} \tag{12}$$
$$\text{s.t.}\quad q_n \ge 0,\ \forall n \in \mathcal{N}, \tag{12.1}$$
$$C_n(L_n + l_0^n)/T \le f_n^{\max},\ \forall n \in \mathcal{N}, \tag{12.2}$$

where constraint (12.1) ensures a positive unit-price, and constraint (12.2) is the maximum computational power constraint.

Obviously, the DU and the SUs have conflict objectives in the game. The objective is to find the NE of the game, at which no user can achieve better utility by unilaterally violating the NE strategy profile $\{\hat{\pi}, \hat{\rho}\}$.

IV. Solution of the Game

In the formulated Bertrand game, the amount of resource that the DU buys is determined by the pricing of the SUs, and vice versa. Additionally, the pricing strategy q_n of SU n is affected not only by its own available resource but also by the pricing of the other SUs in set $\mathcal{N}$, which is represented by $\rho_{-n} = \left(q_1, \cdots, q_{n-1}, q_{n+1}, \cdots, q_{|\mathcal{N}|}\right)$. Next, with the *complete information* hypothesis, that is, the strategies of all the UEs are observable, we try to solve the NE strategy profile $\{\hat{\pi}, \hat{\rho}\}$ of the game. The main methods are given below.

1) Given the pricing strategy ρ of the SUs, the DU can obtain the optimal strtegy $\hat{\pi}$ by solving problem (11).
2) After obtaining $\hat{\pi}$, the SUs in set $\mathcal{N}$ can obtain their optimal pricing strategy $\hat{\rho}$ by solving problem (12).

A. Optimal Resource Purchase of the DU

In this step, we solve problem (11) by using given pricing ρ of the SUs. Note that constraint (11.2) is not considered in this step, so $\sum_{n\in\mathcal{N}} l_0^n > L_0$ is allowed, which means that the DU can buy additional computing resources beyond the demand L_0 from the SUs. This constraint will be dealt with in Sec. IV. C, where the active SU set $\mathcal{N}$ is determined.

In order to obtain the analytic solution of problem (11), one can simplify eq. (9) into

$$U_0 = -\frac{1}{2}\left(\sum_{n\in\mathcal{N}}\left(\frac{H_2}{g_0^n}+1\right)(l_0^n)^2 + 2v\sum\nolimits_{n\ne k} l_0^n l_0^k\right) + \sum\nolimits_{n\in\mathcal{N}}\left(A - \frac{H_1}{g_0^n} - q_n\right) l_0^n, \tag{13}$$

where $A = \kappa_0 (f_0^{\max})^2 C_0$, $H_1 = \ln 2^{\frac{1}{BT/|\mathcal{N}|}}\sigma^2 T/|\mathcal{N}|$ and $H_2 = \left(\ln 2^{\frac{1}{BT/|\mathcal{N}|}}\right)^2 \sigma^2 T/|\mathcal{N}|$ are constants. The derivation is detailed in Appendix A.

Next, one can differentiate U_0 with respect to l_0^n and solve equation $\frac{\partial U_0}{\partial l_0^n} = 0$. Then, the optimal solution of problem (11) is obtained as

$$\hat{l_0^n} = \left[\alpha_n - \beta_n q_n\right]_0^{Q_n^1},\ \forall n \in \mathcal{N}, \tag{14}$$

where $Q_n^1 = \min\left(L_0, \log_2\left(\frac{P g_0^n}{\sigma_0^2}+1\right)\frac{BT}{|\mathcal{N}|}\right)$,

$$\alpha_n = \frac{A - \frac{H_1}{g_0^n}\left(v\left(K - \frac{1}{H_2/g_0^n - v + 1}\right) + 1\right) + v\sum_{n\ne k}\left(\frac{H_1/g_0^k + q_k}{H_2/g_0^k - v + 1}\right)}{\left(H_2/g_0^n - v + 1\right)(vK+1)},$$

$$\beta_n = \frac{v\left(K - \frac{1}{H_2/g_0^n - v + 1}\right) + 1}{\left(H_2/g_0^n - v + 1\right)(vK+1)} \text{ and } K = \sum_{n\in\mathcal{N}} \frac{1}{H_2/g_0^n - v + 1}$$

are constants when all q_k are known for $\forall k \in \mathcal{N}$ and $k \ne n$, and $[x]_a^b = \max\left(\min(x, b),\ a\right)$.

B. Optimal Pricing Strategies of the SUs

By substituting the solution of problem (11) given in eq. (14) into problem (12), problem (12) is rewritten as

$$\max_{q_n}\ U_n = q_n \hat{l_0^n} - p_n^{\text{rec}} t_n - \frac{\kappa_n C_n^3\left((L_n + \hat{l_0^n})^3 - L_0^3\right)}{T^2},$$
$$\forall n \in \mathcal{N} \tag{15}$$
$$\text{s.t. } q_n \ge 0,\ \forall n \in \mathcal{N}, \tag{15.1}$$
$$C_n(L_n + \hat{l_0^n})/T \le f_n^{\max},\ \forall n \in \mathcal{N}. \tag{15.2}$$

By combining eq. (14) and constraint (15.2), we can get the more accurate range of $\hat{l_0^n}$ as

$$\hat{l_0^n} = \left[\alpha_n - \beta_n q_n\right]_0^{Q_n}, \tag{16}$$

where $Q_n = \min\left(Q_n^1, Q_n^2\right)$, and $Q_n^2 = \frac{T f_n^{\max}}{C_n} - L_n$. By using eq. (16), the pricing range of SU n is obtained as

$$q_n \in \left[\frac{\alpha_n - Q_n}{\beta_n}, \frac{\alpha_n}{\beta_n}\right], \tag{17}$$

which means that SU n cannot benefit by choosing prices higher than $\frac{\alpha_n}{\beta_n}$ or lower than $\frac{\alpha_n - Q_n}{\beta_n}$. Next, in order to solve problem (12), we give the following Lemma.

Lemma 1. *When $\hat{l_0^n} = \alpha_n - \beta_n q_n$ and q_n satisfies condition* (17)*, U_n is concave with respect to q_n.*

Proof. Please refer to Appendix B. □

According to **Lemma 1**, one can differentiate U_n with respect to q_n and let $\frac{\partial U_n}{\partial q_n} = 0$. Then, the optimal solution to problem (15) is obtained as

$$\hat{q_n} = \begin{cases} \frac{\alpha_n - Q_n}{\beta_n} & 0 \le \mu_n < \frac{\alpha_n - Q_n}{\beta_n} \\ \mu_n & \frac{\alpha_n - Q_n}{\beta_n} \le \mu_n \le \frac{\alpha_n}{\beta_n} \\ \frac{\alpha_n}{\beta_n} & \mu_n > \frac{\alpha_n}{\beta_n} \end{cases} \tag{18}$$

where $\mu_n = \frac{-\sqrt{6F_nL_n\beta_n + 3F_n\alpha_n\beta_n + 1} + 3L_nF_n\beta_n + 3F_n\alpha_n\beta_n + 1}{3F_n\beta_n^2}$, and $F_n = \frac{\kappa_n C_n^3}{T^2}$. The derivation is in Appendix C.

C. Algorithm Implementation

In this section, we first consider the *complete information* game (CIG), in which the channel state information (CSI) of the UEs, the pricing strategy of the SUs, and the resource purchase of the DU are all observable to the UEs. It is noted that the pricing q_n of each SU n is the function of g_0^n, i.e., the CSI between it and the DU. Therefore, the SU selection and scheduling algorithm requires to obtain the CSI via the dedicated feedback channel. According to the hierarchical structure of the game, we propose an iterative algorithm to find the NE strategy profile $\{\hat{\pi}, \hat{\rho}\}$ of the CIG which is given in **Algorithm 1**. We use $i = 1, 2, \cdots$ to represent an index sequence of the number of iterations. In the ith iteration, we use $q_n[i]$, $\rho_{-n}[i]$ and $\pi[i]$ to represent the pricing of SU n, the pricing of the SUs in set $\mathcal{N}$ other than SU n, and the resource purchase of the DU, respectively.

Algorithm 1 Solving CIG

1: Let $i = 1$. Initialize the pricing of the SUs in set $\mathcal{N}$ as $\rho[i]$.
2: The SUs broadcast their CPU parameters $f_n^{\max}$ to the DU.
3: With the given $\rho[i]$, the DU obtain the optimal resource purchase $\pi[i]$ by using eq. (16).
4: **repeat**
5: For each SU n in set $\mathcal{N}$, after collecting $\pi[i]$ from the DU and $\rho_{-n}[i]$ from the other SUs in set $\mathcal{N}$, SU n obtains its optimal pricing $q_n[i]$ by using eq. (18).
6: For the DU, after collecting $\rho[i]$ from the SUs in set $\mathcal{N}$, it obtains the optimal resource purchase $\pi[i]$ by using eq. (16).
7: Each SU can obtain the gradient of its utility $\nabla U_n[i](q_n[i])$
8: Update $i = i + 1$.
9: **until** $\|\nabla U_n[i](q_n[i])\| \le \epsilon \|\nabla U_n[i-1](q_n[i-1])\|, \ \forall n \in \mathcal{N}$

Considering the system is composed of one DU and two SUs, the convergence condition of **Algorithm 1** is analyzed in the following **Lemma 2**.

Lemma 2. ***Algorithm 1*** *can converge to a stable point $\{\pi[i] = \hat{\pi}, \rho[i] = \hat{\rho}\}$ as the number of iterations i increases.*

Proof. Please refer to Appendix D. □

The convergence speed of **Algorithm 1** mainly depends on the choice of the convergence threshold ϵ. Since $U_n(q_n)$ is strictly concave with respect to q_n, according to [9], the general upper bound on the number of iterations of **Algorithm 1** to reach a certain convergence threshold ϵ is $\mathcal{O}(\log(1/\epsilon))$. For example, when the initial price is chosen arbitrarily from the feasible domain and the threshold ϵ is set to 10^{-3}, **Algorithm 1** converges to a stable point after 10 iterations. It agrees to the simulation results as shown in Fig. 1. In addition to the CSI, the implementation of **Algorithm 1** also requires the DU and SUs to exchange $\{\pi[i], \rho[i]\}$ in any ith iteration. However, this information is private and difficult to obtain in practical application. Next, we consider a more realistic *incomplete information* game (ICIG), in which the only information available for any SU n to make decision is $l_0^n(i)$, that is the amount of resource purchased by the DU from SU n in any ith iteration.

Next, we propose a distributed iterative algorithm based on projected gradient descent (PGD) [10] to find the NE of the ICIG. Firstly, we define

$$\frac{\partial U_n}{\partial q_n} \approx \frac{U_n(q_n[i] + \delta) - U_n(q_n[i] - \delta)}{2\delta}, \ \forall n \in \mathcal{N}, \tag{19}$$

where δ is a sufficiently small positive number (e.g., $\delta = 10^{-5}$). According to the rules of PGD, any SU n should adjust its strategy in the direction that maximizes its own utility but not beyond the feasible domain. According to constraint (12.1), we define the feasible domain of the pricing strategies of the SUs as $\mathcal{X} = \{x | x \ge 0\}$. Then, the price update strategy for SU n in any ith iteration is designed as

$$q_n[i+1] = \arg\min_{x \in \mathcal{X}} \left\| x - \left(q_n[i] + a_n \left(\frac{\partial U_n}{\partial q_n} \right) \right) \right\|_2^2, \ \forall n \in \mathcal{N}, \tag{20}$$

where a_n is the adjustment speed (i.e., learning rate). The convergence of the PGD is analyzed in [10], thus omitted here.

The last issue to be addressed is to apply constraint (11.2) to the game, that is, to determine which SUs are in set $\mathcal{N}$ and solve the problem of *who to cooperate*. One simple and straightforward approach is to initialize $\mathcal{N} = \mathcal{M} - \{0\}$. After performing the game, one can make the following choices according to the obtained NE. If the NE satisfies constraint (11.2) and $l_0^n > 0$ for $\forall n \in \mathcal{N}$, the algorithm terminates and the NE is taken as the final solution of the game. Otherwise, the SU with the highest price is removed from set $\mathcal{N}$, and the game is performed again until constraint (11.2) is satisfied or set $\mathcal{N}$ is empty. Since the the elements of the potential SU set $\mathcal{M} - \{0\}$ is limited, this algorithm is bound to terminates. The detail of the algorithm is given in **Algorithm 2**.

Algorithm 2 SU selection algorithm

Input: The potential SUs set $\mathcal{M} - \{0\}$ for the DU.
Initialization: Let $\mathcal{N} = \mathcal{M} - \{0\}$.
2: **repeat**
Perform the game and obtain the NE $\{\hat{\pi}, \hat{\rho}\}$ of the game.
4: Remove any SU n with $\hat{l_0^n} = 0$ from set $\mathcal{N}$.
if $\sum_{n \in \mathcal{N}} l_0^n > L_0$ **then**
6: Remove the SU with the highest price from set $\mathcal{N}$.
end if
8: **until** $\sum_{n \in \mathcal{N}} l_0^n \le L_0$ and $l_0^n > 0$ for $\forall n \in \mathcal{N}$, or $\mathcal{N} = \varnothing$.
Output: The active SU set $\mathcal{N}$ for the DU.

V. Simulation results

In the simulations, the system parameters are set as below. The duration of a time slot is $T = 0.2$ s. The system bandwidth

is $B = 1$ MHz. All the UEs are with the same switch capacitance coefficient $\kappa_m = 10^{-28}$. The number of CPU cycles required to execute megabit data is $C_m = 8 \times 10^8$. To complete the computing task in slot t, the maximum CPU frequencies allocated by the DU and SU n are $f_0^{\max} = 2.4$ GHz and $f_n^{\max} = 1.5$ GHz, respectively. The maximum transmit power of the DU is $P = 0.1$ W. The power of the receiving circuit of SU n is $p_n^{\text{rec}} = 0.01$ W. The path loss gain is set to $0.001/d^3$ (where d is the distance between the transmitter and the receiver (in meters)). The noise power at the receiver of a SU is $\sigma^2 = 10^{-9}$. The substitutability factor for the DU's utility function is set to $v = 0.5$.

To testify the convergence of the proposed algorithms, we place one DU at coordinate $(0, 0)$, and two SUs at coordinates $(-20, 20)$ and $(20, 20)$, respectively. The amount of task data of the DU and the SUs are set to $L_0 = 0.6$ Mb, $L_1 = 0.15$Mb, and $L_2 = 0$ Mb, respectively. Fig. 1 shows the convergence of the price of the SUs with the increase of iteration numbers.

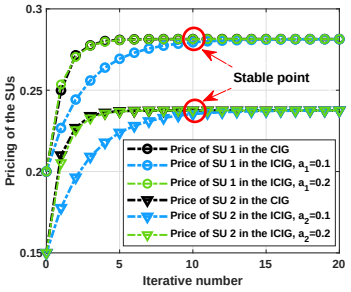

Fig. 1. The convergence to the NE price.

From Fig. 1, one can see that **Algorithm 1** requires only a few iterations to converge to the NE in the CIG. Whereas, in the ICIG, the convergence speed depends largely on the learning rate a_i in eq. (20). When the learning rate is properly set, e.g., $a_1 = a_2 = 0.2$, the ICIG converges to the NE as fast as the CIG. In addition, we note that SU 2 has a price advantage over SU 1, because SU 2 is with more idle computing resources than SU 1 in the current slot.

Next, Fig. 2 and Fig. 3 respectively show the convergence of the amount of offloaded data from the DU to the SUs and the convergence of their utilities in the ICIG.

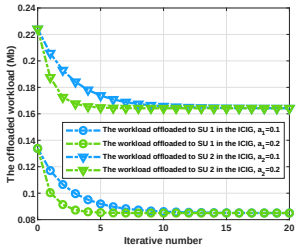

Fig. 2. The convergence of the amount of offloaded workload.

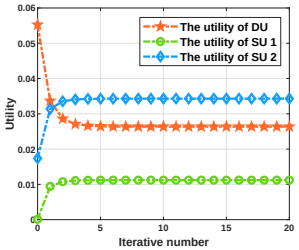

Fig. 3. The convergence of user utilities.

From Fig. 2, one can see that SU 2 accepts more computing tasks of the DU than SU 1 in the game. This indicates that the proposed pricing game can coordinate the resource supply of the SUs according to their current available resources. From Fig. 3, one can see that all the DU and SUs obtain positive utilities from the game. Since SU 2 sells more resources than SU 1, it obtains a higher utility than SU 1. It implies that the proposed game provides sufficient motivation to rational users to participate in cooperative computation.

Finally, we simulate a system consisting of one DU and three SUs. The coordinates of the DU and SUs are $(0, 0)$, $(-20, 20)$, $(20, 20)$ and $(20, -20)$, respectively. The amount of task data of SUs 1 and 2 are $L_1 = 0.15$ Mb and $L_2 = 0.1$ Mb, respectively. We increase the workload of SU 3 from 0 Mb to 0.15 Mb at a step of 0.05 Mb. Fig. 4 shows the variation of the amount of task data offloaded from the DU to the SUs.

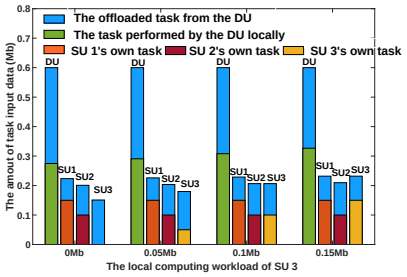

Fig. 4. The workload distribution on the users.

From Fig. 4, one can see that with the increase of the workload of SU 3, the amount of task data offloaded from the DU to SU 3 decreases gradually. At the same time, the workloads offloaded from the DU to SU 1 and SU 2 have a small increase. It indicates that the proposed game can coordinate the resource provision of the SUs according to their current available resources.

VI. Conclusion

In this paper, an incentive scheme based on Bertrand game has been proposed to stimulate rational users to participate in cooperative computation framework. When the game is with *complete information*, the NE of the game is obtained in closed-form, while in the case of *incomplete information*, a distributed iterative algorithm has been developed to find the NE. Simulation results have verified the effectiveness of the proposed scheme.

Appendix A
Derivation of Function (13)

A Maclaurin series is a function which has expansion series that gives the sum of derivatives of that function. In order to obtain the analytic solution of problem (11), we derive the Maclaurin series of the second term of function (9) up to order $n = 2$. Then function (9) is rewritten as

$$U_0 = \kappa_0 (f_0^{\max})^2 C_0 \sum_{n\in\mathcal{N}} l_0^n - \frac{\sigma^2 T/|\mathcal{N}|}{g_0^n} \sum_{n\in\mathcal{N}} \left(l_0^n \ln 2^{\frac{1}{BT/|\mathcal{N}|}} + \frac{1}{2}\left(l_0^n \ln 2^{\frac{1}{BT/|\mathcal{N}|}} \right)^2 \right) - \frac{1}{2}\left(\sum_{n\in\mathcal{N}} (l_0^n)^2 + 2v \sum_{n\neq k} l_0^n l_0^k \right) - \sum_{n\in\mathcal{N}} q_n l_0^n. \tag{S.1}$$

After some algebraic manipulation, we can further rewrite (S.1) as

$$U_0 = -\frac{1}{2}\left(\sum_{n\in\mathcal{N}}\left(\frac{H_2}{g_0^n}+1\right)(l_0^n)^2 + 2v\sum\nolimits_{n\neq k} l_0^n l_0^k\right) + \sum\nolimits_{n\in\mathcal{N}}\left(A - \frac{H_1}{g_0^n} - q_n\right) l_0^n, \quad \text{(S.2)}$$

where $A = \kappa_0 (f_0^{\max})^2 C_0$, $H_1 = \ln 2^{\frac{1}{BT/|\mathcal{N}|}}\sigma^2 T/|\mathcal{N}|$, $H_2 = (\ln 2^{\frac{1}{BT/|\mathcal{N}|}})^2\sigma^2 T/|\mathcal{N}|$ are constants.

Appendix B
Proof of Lemma 1

We take the first order derivative of U_n in eq. (15) with respect to q_n, and we have

$$\frac{\partial U_n}{\partial q_n} = \hat{l_0^n} + q_n\frac{\partial \hat{l_0^n}}{\partial q_n} - 3F_n\left(L_n + \hat{l_0^n}\right)^2\frac{\partial \hat{l_0^n}}{\partial q_n}, \quad \text{(S.3)}$$

where $F_n = \frac{\kappa_n C_n^3}{T^2}$.

Since $\hat{l_0^n} = \alpha_n - \beta_n q_n$ is considered, we have

$$\frac{\partial U_n}{\partial q_n} = \hat{l_0^n} - q_n\beta_n + 3F_n\beta_n\left(L_n + \hat{l_0^n}\right)^2. \quad \text{(S.4)}$$

Next, we further derive the second order derivative of eq. (15) with respect to q_n as

$$\frac{\partial^2 U_n}{\partial q_n^2} = -2\beta_n - 6F_n\beta_n^2\left(L_n + \hat{l_0^n}\right). \quad \text{(S.5)}$$

Since $F_n > 0$, $\frac{\partial \hat{l_0^n}}{\partial q_n} = -\beta_n < 0$, $\frac{\partial^2 \hat{l_0^n}}{\partial q_n^2} = 0$, and $\hat{l_0^n} \geq 0$, we can obtain $\frac{\partial^2 U_n}{\partial q_n^2} < 0$. Therefore, we conclude that the utility function U_n of SU n is concave with respect to q_n.

Appendix C
Derivation of Equation (18)

Solving function $\frac{\partial U_n}{\partial q_n} = 0$ is equivalent to solving the following quadratic equation

$$\alpha_n - 2\beta_n q_n + 3F_n\beta_n(L_n + \alpha_n - \beta_n q_n)^2 = 0. \quad \text{(S.6)}$$

By solving eq. (S.6), one can get the following two solutions, i.e.,

$$q_n = \mu_{n,1} = \frac{3L_nF_n\beta_n + 3F_n\alpha_n\beta_n + 1 + \sqrt{6F_nL_n\beta_n + 3F_n\alpha_n\beta_n + 1}}{3F_n\beta_n^2} \quad \text{(S.7)}$$

and

$$q_n = \mu_{n,2} = \frac{3L_nF_n\beta_n + 3F_n\alpha_n\beta_n + 1 - \sqrt{6F_nL_n\beta_n + 3F_n\alpha_n\beta_n + 1}}{3F_n\beta_n^2} \quad \text{(S.8)}$$

From the first solution, we can derive $\mu_{n,1} > \frac{\alpha_n}{\beta_n}$. It means that this solution is beyond the range of feasible prices, resulting in the amount of input data of the offloaded task being 0 and, hence, $U_n = 0$. So equation (S.6) has the unique solution $q_n = \mu_{n,2}$. For easy expression, we define $\mu_n = \mu_{n,2}$. Then we have $q_n = \mu_n$.

Now, we determine the solution of problem (15) according to the feasible price range where μ_n is located.

- If $\mu_n < \frac{\alpha_n - Q_n}{\beta_n}$, U_n is monotonically decreasing with respect to the q_n over the feasible price range. So the optimal price is $\hat{q_n} = \frac{\alpha_n - Q_n}{\beta_n}$.
- If $\frac{\alpha_n - Q_n}{\beta_n} \leq \mu_n \leq \frac{\alpha_n}{\beta_n}$, the optimal price is $\hat{q_n} = \mu_n$.
- If $\mu_n > \frac{\alpha_n}{\beta_n}$, U_n is monotonically increasing respect to q_n over the feasible price range. So the optimal price is $\hat{q_n} = \frac{\alpha_n}{\beta_n}$.

Finally, the solution of problem (15) is given as

$$\hat{q_n} = \begin{cases} \frac{\alpha_n - Q_n}{\beta_n} & 0 \leq \mu_n < \frac{\alpha_n - Q_n}{\beta_n} \\ \mu_n & \frac{\alpha_n - Q_n}{\beta_n} \leq \mu_n \leq \frac{\alpha_n}{\beta_n} \\ \frac{\alpha_n}{\beta_n} & \mu_n > \frac{\alpha_n}{\beta_n} \end{cases} \quad \text{(S.9)}$$

Appendix D
Proof of Lemma 2

According to eq. (18), we can obtain the self-mapping function of SU n as

$$q_n[i+1] = \begin{cases} \frac{\alpha_n[i] - Q_n}{\beta_n} & 0 \leq \mu_n[i] < \frac{\alpha_n[i] - Q_n}{\beta_n} \\ \mu_n[i] & \frac{\alpha_n[i] - Q_n}{\beta_n} \leq \mu_n[i] \leq \frac{\alpha_n[i]}{\beta_n} \\ \frac{\alpha_n[i]}{\beta_n} & \mu_n[i] > \frac{\alpha_n[i]}{\beta_n} \end{cases} \quad \text{(S.10)}$$

The Jacobian matrix of eq. (S.10) is given by

$$\mathbf{J} = \begin{bmatrix} J_{1,1} & J_{1,2} \\ J_{2,1} & J_{2,2} \end{bmatrix} = \begin{bmatrix} \frac{\partial q_1[i+1]}{\partial q_1[i]} & \frac{\partial q_1[i+1]}{\partial q_2[i]} \\ \frac{\partial q_2[i+1]}{\partial q_1[i]} & \frac{\partial q_2[i+1]}{\partial q_2[i]} \end{bmatrix}. \quad \text{(S.11)}$$

By definition, the self-mapping function (S.10) is stable if and only if the eigenvalues λ_n of $\mathbf{J}$ are all inside the unit circle of the complex plane, i.e., $|\lambda_n| < 1$.

Next, we derive the expression of the elements of $\mathbf{J}$ as

$$J_{1,1} = J_{2,2} = \frac{\partial q_n[i+1]}{\partial q_n[i]} = 0, \quad \text{(S.12)}$$

$$J_{1,2} = \frac{\partial q_1[i+1]}{\partial q_2[i]} = \begin{cases} \frac{1}{\beta_1}\frac{\partial \alpha_1}{\partial q_2} & 0 \leq \mu_1[i] < \frac{\alpha_1[i] - Q_1}{\beta_1} \\ \left(1 - \frac{1}{2\sqrt{\zeta_1}}\right)\frac{1}{\beta_1}\frac{\partial \alpha_1}{\partial q_2} & \frac{\alpha_1[i] - Q_1}{\beta_1} \leq \mu_1[i] \leq \frac{\alpha_1[i]}{\beta_1} \\ \frac{1}{\beta_1}\frac{\partial \alpha_1}{\partial q_2} & \mu_1[i] \geq \frac{\alpha_1[i]}{\beta_1} \end{cases} \quad \text{(S.13)}$$

and

$$J_{2,1} = \frac{\partial q_2[i+1]}{\partial q_1[i]} = \begin{cases} \frac{1}{\beta_2}\frac{\partial \alpha_2}{\partial q_1} & 0 \leq \mu_2[i] < \frac{\alpha_2[i] - Q_2}{\beta_2} \\ \left(1 - \frac{1}{2\sqrt{\zeta_2}}\right)\frac{1}{\beta_2}\frac{\partial \alpha_2}{\partial q_1} & \frac{\alpha_2[i] - Q_2}{\beta_2} \leq \mu_2[i] \leq \frac{\alpha_2[i]}{\beta_2} \\ \frac{1}{\beta_2}\frac{\partial \alpha_2}{\partial q_1} & \mu_2[i] \geq \frac{\alpha_2[i]}{\beta_2} \end{cases} \quad \text{(S.14)}$$

In eq. (S.13) and eq. (S.14), we have

$$\zeta_1 = 6L_1F_1\beta_1 + 3F_1\alpha_1\beta_1 + 1, \quad \text{(S.15)}$$

$$\zeta_2 = 6L_2F_2\beta_2 + 3F_2\alpha_2\beta_2 + 1, \quad \text{(S.16)}$$

$$\frac{1}{\beta_1}\frac{\partial \alpha_1}{\partial q_2} = \frac{\frac{v}{H_2/g_0^2 - v + 1}}{\frac{v}{H_2/g_0^2 - v + 1} + 1} < 1 \quad \text{(S.17)}$$

and

$$\frac{1}{\beta_2}\frac{\partial \alpha_2}{\partial q_1} = \frac{\frac{v}{\overline{H_2/g_0^1 - v + 1}}}{\frac{v}{\overline{H_2/g_0^1 - v + 1}} + 1} < 1. \tag{S.18}$$

Then, we know $J_{1,2} < 1$ and $J_{2,1} < 1$.

Since all the elements of $\mathbf{J}$ are less than one, the eigenvalues of $\mathbf{J}$ are all less than 1, which is given by

$$(\lambda_1, \lambda_2) = \frac{(J_{1,1} + J_{2,2}) \pm \sqrt{4J_{1,2}J_{2,1} + (J_{1,1} - J_{2,2})^2}}{2}, \tag{S.19}$$

which indicates that the eigenvalues λ_n are all inside the unit circle of the complex plane. Therefore, **Algorithm 1** can converge to a stable point.

REFERENCES

[1] X. Chen and J. Zhang, "When d2d meets cloud: Hybrid mobile task offloadings in fog computing," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[2] D. Wu, F. Wang, X. Cao, and J. Xu, "Wireless powered user cooperative computation in mobile edge computing systems," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–7.

[3] Q. Lin, F. Wang, and J. Xu, "Optimal task offloading scheduling for energy efficient d2d cooperative computing," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1816–1820, 2019.

[4] X. Chen, Z. Zhou, W. Wu, D. Wu, and J. Zhang, "Socially-motivated cooperative mobile edge computing," *IEEE Network*, vol. 32, no. 6, pp. 177–183, 2018.

[5] Z. Sheng, C. Mahapatra, V. C. M. Leung *et al.*, "Energy efficient cooperative computing in mobile wireless sensor networks," *IEEE Trans. Cloud Computing*, vol. 6, no. 1, pp. 114–126, 2018.

[6] N. Singh and X. Vives, "Price and quantity competition in a differentiated duopoly," *The RAND Journal of Economics*, vol. 15, no. 4, pp. 546–554, 1984.

[7] D. Niyato and E. Hossain, "Market-equilibrium, competitive, and cooperative pricing for spectrum sharing in cognitive radio networks: Analysis and comparison," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4273–4283, 2008.

[8] W. S. Oh and R. Muneepeerakul, "How do substitutability and effort asymmetry change resource management in coupled natural-human systems?" *Palgrave Communications*, vol. 5, 2019.

[9] C. Ma, J. Konečný, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, 2017.

[10] A. Ang, "Projected gradient algorithm," https://angms.science/doc/CVX/CVX_PGD.pdf.