U.P.B. Sci. Bull., Series C, Vol. 87, Iss. 4, 2025 ISSN 2286 – 3540

COOPERATION GAME-BASED RESOURCE ALLOCATION FOR TASK OFFLOADING IN MOBILE EDGE COMPUTING

Ruixia LI[1,2], Chia Sien LIM[2], Muhammad Ehsan RANA[2], Jinsong TU[3,*]

To improve the users' satisfaction and resource utilization of task offloading in Mobile Edge Computing (MEC), an incentive mechanism for resource allocation is considered. It motivates edge service providers to actively participate in task offloading by setting the resource price. A bargaining-based cooperative game model is proposed to charge terminal devices and reward edge servers. The Nash equilibrium is analyzed with complete information game theory. The result indicates that this strategy PSNCG can ensure that edge nodes participating in cooperative resource allocation can obtain maximum utility at an acceptable cost, thereby improving users' experience and resource utilization.

Keywords: Cooperative game, mobile edge computing, resource allocation, task offloading

1. Introduction

Cloud computing has become a key factor driving the development of social intelligence and the integration of cloud networks. However, with the popularity of mobile smart devices and continuous emergence of various network applications, more and more users are accessing cloud computing centers, which brings new challenges to the development of cloud computing. For example, new users such as the Internet of Things (IoT), Industry 4.0 and the smart agriculture have different requirements for computing, storage, and service performance, which makes it difficult for users far from cloud computing centers to receive timely and effective services. To overcome the limitations of cloud computing, further improve the quality and resource utilization of user service, MEC has received widespread attention [1]. As shown in Fig. 1, MEC can effectively shorten the distance between users and computing storage services, reducing possible network congestion and transmission delay [2].

* Corresponding author: jids8012@126.com
[1] School of Electronics and Information Engineering, West Anhui University, Lu'an, 237012, China
[2] Graduate School of Technology, Asia Pacific University of Technology and Innovation, Kuala Lumpur 57000, Malaysia
[3] School of Architecture and Civil Engineering, West Anhui University, Lu'an, 237012, China,

Task offloading in MEC means that users can offload tasks to adjacent edge nodes for processing and then save battery capacity and break through computing performance limitations [3]. Cloud computing centers and edge nodes collaborate or compete to provide support for user task offloading requests [4].

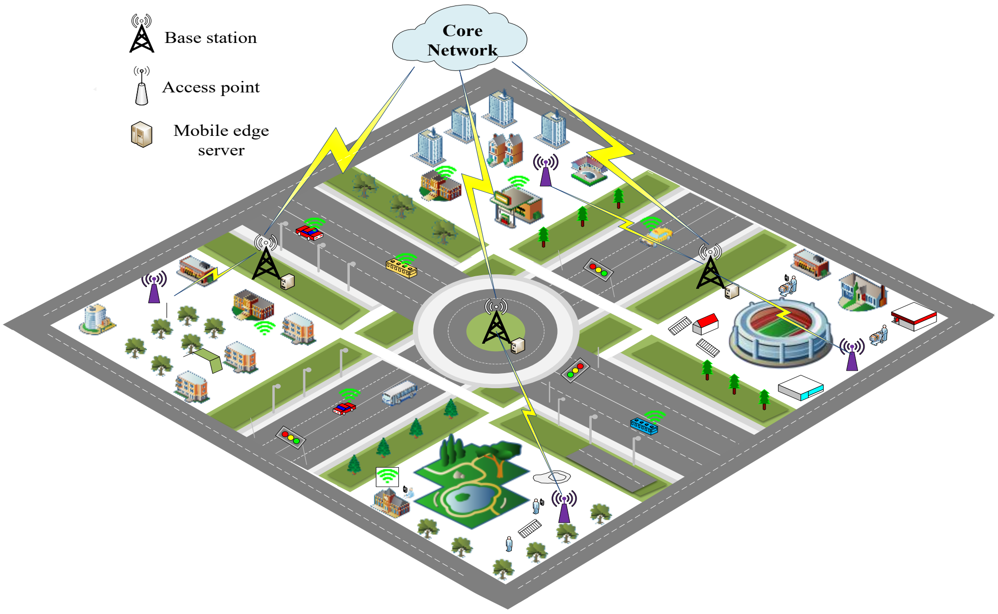

Fig. 1. The MEC architecture framework

Compared with traditional cloud computing, the resources of the edge servers are limited, which may lead to long processing time and low efficiency. It will affect the user's service experience if there is no reasonable mechanism of resource allocation.

Resource allocation is focused in MEC. The resource pricing and incentive mechanism in Section 3 is based on Nash bargaining game theory, encouraging service providers to actively engage in task offloading and improving system resource utilization. The main contributions can be summarized as follows:

- Nash bargaining process is introduced to determine the winning coalition for factors affecting resource prices; the final price and payment price are obtained to guarantee the quality of service and the enthusiasm of edge service providers;
- A cooperative game model is developed, in which a scientific utility function is designed to maximize the benefits of users and service providers;
- Nash equilibrium solution will be established to confirm the effectiveness of this strategy.

The structure is arranged as follows: Section 2 presents a review of relevant researches on task offloading and resource allocation in MEC; Section 3 analyzes the factors influencing pricing and the Nash bargaining process; Section 4 explains the cooperative game model; Section 5 proves the Nash equilibrium solutions in a complete information game theory and proposes an algorithm for the Nash bargaining process; Section 6 evaluates the proposed strategy, and Section 7 gives the conclusion.

2. Related works

From the perspective of computing resource allocation, an auction scheme was designed for computing resource trading to ensure the privacy of buyers as well as sellers while maximizing social welfare [5]. Guo et al. [6] considered blockchain and established an incentive mechanism based on a game model to ensure the acquisition of computing resources and the active participation of service providers.

Achieving peak performance of the entire system is challenging for communication or computing resources. Wei et al. [7] considered the coupling of calculation and transmission delay, as well as calculation and channel capacity, and optimized the completion time of discrete task by designing corresponding resource allocation strategies. Zhu et al. [8] used task offloading algorithms to allocate the resources and applying delay constraints to minimize the total energy consumption of the system.

The pricing strategy of bandwidth resources was studied for resource pricing [9]. Baek et al. [10] established a resource pricing mechanism in MEC, which can effectively allocate resources. Chen et al. [11] proposed a pricing resource allocation method that maximizes the profits of operators and uses Lyapunov optimization techniques to optimize utility maximization while ensure system stability.

Some researchers have proposed incentive mechanism based on economic theory and game theory to address the incentive challenges associated with task offloading in MEC. Based on a non-cooperative environment, the resource allocation of virtual machines was modeled as a graph matching problem [12]. The introduction of auction mechanism aims to optimize collective social welfare. Considering the utility of edge nodes and user profit constraints, Wang et al. [13] designed an incentive algorithm to maximize the profit of edge servers.

The impact of resource allocation of incentive mechanism on task offloading is seldom considered and conducted in non-collaborative environments. There are few incentive mechanisms for resource allocation considering cooperative games. In response to the above issues, we focus on the incentive mechanism for resource allocation in cooperative game scenarios. Then a complete information game model is established to investigate the Nash equilibrium solution with an edge node and multiple terminal devices.

Several works have investigated resource pricing in MEC economics. For example, Zhou et al. [9] studied time-dependent bandwidth pricing under information asymmetry, Baek et al. [10] proposed dynamic pricing schemes for IoT-edge environments, and Chen et al. [11] developed an online dynamic pricing framework for edge computing. Compared with these pricing methods, our PSNCG (Pure Strategy Nash Cooperative Game) focuses on bargaining-based coalition formation, which integrates both economic pricing and cooperative incentives.

3. Preliminary assumptions to resource allocation in MEC

3.1. Assumptions

The model relies on the following assumptions: (i) quasi-static Rayleigh fading channels, (ii) a trusted third party assists in resource allocation, and (iii) cooperation among edge nodes is disabled in the current setup. These assumptions are adopted for analytical tractability and are clearly stated here for completeness.

3.2. Factors affecting price setting of resource allocation

Due to limited resources for edge nodes, pending applications offloaded to edge nodes may exceed their service capabilities. A trusted third party is introduced to assist edge nodes to allocate resources [13]. To simply the analysis, only one edge node is used to demonstrate the proposed mechanism with multiple edge nodes. Cooperation between edge nodes is not considered.

Resource allocation is considered by setting the resource price, so that service providers are willing to actively participate in task offloading. Game theory is a widely adopted trading strategy for efficiently allocating sellers' resources to buyers in a competitive market with fair pricing [14]. Referring to the notation and definition rules [15], Table 1 shows the main notations.

Table 1

Description of notations

Notation	Definition
$node_0$	The edge node
$node_i$	Terminal device i
$Price$	The final agreeing price
$Price_i$	The payment price of terminal device $node_i$
r_0	The reservation price of edge node $node_0$
b_0	The bidding price of edge node $node_0$
r_i	The reservation price of terminal device $node_i$

s_i	The asking price of terminal device $node_i$
$u_i\{s_{node_i}, s_{node_{-i}}\}$	The utility function of terminal node $node_i$
$u_0\{s_{node_0}, s_{node_{-0}}\}$	The utility function of edge node $node_0$
s_{node_i}	The strategy of terminal node $node_i$
$s_{node_{-i}}$	The strategies of other nodes except for terminal node $node_i$
W_i	The task i
D_0	The cache resource of edge node $node_0$
C_0	The CPU cycles of edge node $node_0$
D_i	The data size of task W_i
C_i	The required number of CPU cycles to complete task W_i
$Delay_i^l$	The delay processed the task W_i locally
$Delay_i^m$	The delay offloading the task W_i to edge node
$Energy_i^l$	The constraint of energy consumption processing local task W_i
$Energy_i^m$	The required energy consumption offloading the task W_i to edge node
f_i	The CPU resource allocated by the edge node to the task W_i
$Bandwidth_0$	The bandwidth of edge node $node_0$
$Bandwidth_i$	The bandwidth to complete the task W_i

3.2.1. Delay

Let $W_i = \{D_i, C_i, Delay_i^l, Energy_i^l\}$ denote a task, where D_i and C_i represent the data size and the required number of CPU cycles of the task W_i respectively. $Delay_i^l$ and $Enegy_i^l$ denote the constraints of delay and energy consumption processing local task W_i . Delay is an important factor affecting the price of task offloading. The smaller the delay required to complete a task, the higher the price it should provide [16].

$$Delay_i^m = \frac{C_i}{f_i} + \frac{D_i}{rate_i} \tag{1}$$

$$rate_i = Bandwidth_i \log_2(1 + \frac{p_i^{up} h_i}{N_i}) \tag{2}$$

where f_i represents the CPU resource allocated to the task W_i by the edge node. $rate_i$ denotes the data transmission rate of offloading tasks to edge nodes, $h_i = d_n^{-s}$, $s = 4$, and $d_n = [0,50]$ [16]. $Bandwidth_i$ denotes the user bandwidth, that is, the bandwidth required to complete task W_i . P_i^{up} and N_i represent the transmission power consumption and the noise power respectively.

3.2.2. Energy consumption

Energy consumption is another important factor affecting task offloading, the lower the energy consumption required to complete a task, the higher the price. We focus on terminal devices to address the energy consumption by defining the energy consumption required to process task W_i [17].

$$Energy_i^m = (\frac{D_i}{rate_i} + \frac{C_i}{f_i}) \times P_i^{up} \tag{3}$$

3.2.3. Bandwidth

The channel is assumed to be quasi-static Rayleigh fading, which means that the bandwidth of terminal devices remains unchanged during each of the period [17]. Given a task offloading request of W_i , the smaller the delay required to complete a task, the higher the bandwidth it should be provided. The required bandwidth $Bandwidth_i$ can be obtained as follows:

$$Bandwidth_i = \frac{D_i}{Delay_i^m} \tag{4}$$

To simply the analysis, a single-edge-node model is adopted to investigate the pricing and allocation of limited edge resources. This modeling choice allows for a clear examination of the core interactions between edge resources and terminal devices without involving additional inter-node complexity.

It is worth noting that although this study uses a single-node setting, the underlying concepts and pricing mechanisms discussed here are also applicable to multi-node MEC environments. Cooperation among edge nodes, such as distributed coordination or inter-node negotiation, is a promising direction but is beyond the scope of this work.

Although this study adopts a single-edge-node setting for tractability and clarity of theoretical analysis, we acknowledge that inter-edge coordination is highly relevant in realistic MEC scenarios. Future extensions of PSNCG will explicitly consider multi-edge cooperation through distributed or hierarchical game-theoretic frameworks.

3.3. Bidding price

The reservation price r_0 indicates the minimum price that the seller is willing to accept. If the seller $node_0$ accepts the task offloading, a bidding price b_0 should be provided based on the reservation price r_0. For a given edge node $node_0$, the value of reservation price r_0 is typically determined by the factors such as cache resources D_0, the number of CPU cycles C_0, and the maximum available bandwidth $Bandwidth_0$. The reservation price r_0 is calculated as follows:

$$r_0 = w_{r0} \times D_0 \times C_0 \times Bandwidth_0 \tag{5}$$

where w_{r0} denotes the weighting, $w_{ro} > 0$. r_0 is positively correlated with D_0, C_0, and $Bandwidth_0$; b_0 is an increasing function of r_0, and $b_0 \geq r_0$.

3.4. Asking price

Terminal device $node_i$ must provide an asking price s_i for each node before participating in task offloading, which usually depends on its reservation price r_i. r_i refers to the highest price that terminal device $node_i$ can pay for task offloading. Given the task W_i that needs to be offloaded, reservation price r_i will be affected by two aspects: 1) the task, including D_i and C_i; 2) the constraints of delay and energy consumption. Then, reservation price r_i of terminal device $node_i$ is expressed as follows:

$$r_i = w_{s_{i0}} \times D_i \times C_i \times (w_{s_{i1}} \times Energy_i^l + (1 - w_{s_{i1}}) \times Delay_i^l) \tag{6}$$

where $w_{s_{i0}}$ and $w_{s_{i1}}$ represent the weight, and $w_{s_{i0}} > 0$, $0 \leq w_{s_{i1}} \leq 1$; $w_{s_{i1}}$ represent the weight of energy consumption. If there is no requirement for energy consumption, $w_{s_{i1}} = 0$; on the contrary, if only energy consumption is concerned, $w_{si1} = 1$, and here $w_{s_{i1}} = 0.5$ [18]. For buyers $node_i$, the asking price s_i is proportional to its reservation price r_i. Without loss of generality, we will regard the asking price s_i as an increasing function of its reservation price r_i, $r_i \geq s_i$.

3.5. Nash bargaining process

When the terminal device $node_i$ sends a task offloading request to the edge node, a Nash bargaining game between the terminal devices and the edge node begins. The specific process is described as follows:

First, edge node $node_0$ calculates a reservation price r_0, chooses its bidding price $b_0(b_0 \geq r_0)$ and sends its relevant resource parameters (D_0, C_0 and $Bandwidth_0$) to the trusted third party [15]. The sent message does not include r_0 and b_0;

Then, the trusted third party broadcasts the information to all possible terminal devices, and all the terminal devices within the coverage area of the edge node $node_0$ with task offloading request will calculate their reservation price r_i according to (6) and provide their own asking price $s_i(s_i \leq r_i)$.

Then the bargaining process begins, with the edge node $node_0$ and all terminal devices $node_i$ requiring resources to handle their offloading tasks submitting their sealed prices b_0 and s_i respectively. Let $\mathrm{C} = \{C \in 2^M \mid \sum_{node_i \in C} s_i \geq b_0\}$ be a set of coalitions, where C is coalition. If there is a unique coalition C in $\mathbf{C}$ (that is $|\mathrm{C}|=1$), then the coalition C is selected to bargain (that is all members of C participate in task offloading). If $|\mathrm{C}|=0$, the bargain fails. If $|\mathrm{C}|>1$, one coalition in $\max MN = \arg\max_{C^j \in \mathrm{C}} |C^j|$ is selected. After a coalition C is selected, it is referred to as the C-Coalition. Once the bargain is finalized, the game will conclude with an agreeing price $Price$, expressed as follows:

$$Price = \varepsilon \times b_0 + (1-\varepsilon) \times \sum_{node_i \in C} s_i,\ 0 < \varepsilon < 1 \tag{7}$$

where $0<\varepsilon<1$. ε is a weighting factor used to determine the final transaction price between the buyers (terminal device) and the seller (edge node). A higher value of ε makes the final price closer to the buyers' quote, while a lower value favors the seller's price.

Here ε is set to 0.5 to reflect a balanced bargaining outcome and ensure fairness between participants. This choice aligns with the principle of equitable negotiation found in Nash bargaining solutions.

Different ε can significantly impact the allocation results: increasing ε can enhance user satisfaction, but may reduce edge server incentives; decreasing ε does the opposite. Future work may consider dynamically adjusting ε based on user priority, service level, or real-time resource demand.

The payment price $Price_i$ of terminal devices $node_i$ is obtained as follows:

$$Price_i = s_i - \frac{-Price + \sum_{node_i \in C} s_i}{|C|} \tag{8}$$

If a node outside the coalition fails, its utility is equal to zero.

4. Cooperative game model

Game theory is introduced in this scheme, and the model for task offloading and resource allocation can be considered as a cooperative game. A key issue in the final decision-making of game theory is to consider its costs and benefits.

Definition 1. The resource allocation game G is denoted by a triple $(NODE,S,U)$,

- $NODE = \{node_0\} \cup \{node_i\}$ is the set of participants.
- $S = \{s_{node_i}\}_{i=0}^{N}$ is the strategy set, when the terminal device has a task offloading request, it has two options: $Cooperation(CP)$ or $Defect(D)$. For the edge node $node_0$, when it chooses CP, it will allocate resources to the terminal devices; on the contrary, if it chooses D, any task offloading request will be rejected. Therefore, the strategy set s_{node_i} of terminal device $node_i$ is $\{CP,D\}$. We use s_{node_i} denote the strategy set of terminal device $node_i$, $s_{node_{-i}}$ represents the strategy of other nodes, then the strategy of all nodes can be expressed as $\{s_{node_i}, s_{node_{-i}}\}$.
- $U = \{u_i\}_{i=0}^{N}$ is the utility function set, and $u_i\{s_{node_i}, s_{node_{-i}}\}$ denotes the utility function of $node_i$. The utility function $u_0\{s_{node_0}, s_{node_{-0}}\}$ of edge node $node_0$ is defined as follows:

$$u_0\{s_{node_0}, s_{node_{-0}}\} = \begin{cases} Price - r_0 & \left(\begin{array}{l} if\ s_{node_0} = CP \quad and \quad \exists C \in 2^N \quad such\ \ that \\ \sum_{node_i \in C} s_i \ge b_0 \quad and \quad |C| = \arg\max_{C^j \in \mathrm{C}} |C^j| \end{array} \right) \\ 0 & \qquad\qquad else \end{cases} \tag{9}$$

When the transaction is successful, the profit obtained by $node_0$ is measured by the difference between its bidding price and the final agreed price. If the transaction fails, $node_0$ will not be able to obtain any profit.

The utility function $u_i\{s_{node_i}, s_{node_{-i}}\}$ of terminal device $node_i$ is defined as follows:

$$u_i\{s_{node_i}, s_{node_{-i}}\} = \begin{cases} -r_i, \ if\ s_{node_i} = CP\ and\ node_i \notin C \\ r_i - s_i + \dfrac{\text{-}Price + \sum_{node_i \in C} s_i}{|C|}, \ if\ s_{node_i} = CP\ and\ node_i \in C \\ 0, \ if\ s_{node_i} = D \end{cases} \tag{10}$$

- where the utility of any terminal device outside the coalition is negative because it consumes resources to participate in task offloading;

- The utility of the terminal device within any coalition is $r_i - s_i + \frac{-Price + \sum_{node_i \in C} s_i}{|C|}$;
- The terminal device refuses task offloading, and its utility is equal to 0.

According to game theory, the behavior of each node is rational, and choose a strategy that maximizes its utility when the strategies of other nodes are given. In this case, it represents the optimal response of $node_i$ to the strategies of other nodes, denoted as $s^*_{node_i} = \arg\max_{s_{node_i}} u_i(s_{node_i}, s_{node_{-i}})$.

Definition 2. S^* is a Nash equilibrium strategy if and only if $u_i(s^*_{node_i}, s^*_{node_{-i}}) \geq u_i(s_{node_i}, s^*_{node_{-i}}),\ 1 \leq i \leq N, s_{node_i} \in S_{node_i}$. In the Nash equilibrium condition, no participant can increase its utility by unilaterally changing the strategy, and the Nash equilibrium can ensure the stability of the game.

Definition 3. $v(T)$ is used to denote the collective Pareto-optimal utility, and $T \subseteq N$ is a subset of total nodes. If the utility function of a coalition with only one member is represented as $v(i)$. The utility function of $node_i$ in the coalition T is expressed as x_i. If $x_i \geq v(i)$, and $\sum_{i=1}^{|T|} x_i = v(T)$, the vector $\bar{x} = (x_1, \ldots x_{|T|},)$ is a reasonable utility distribution.

Lemma 1. Given a coalition C defined in subsection 3.5, the utility distribution of (10) is rational.

Proof. Suppose $node_i \in C$, its utility function is expressed as follows:

$$u_i = r_i - s_i + \frac{-Price + \sum_{node_i \in C} s_i}{|C|} = r_i - s_i + \frac{(1-\varepsilon)(\sum_{node_i \in C} s_i - b_0)}{|C|} \tag{11}$$

$r_i - s_i \geq 0$ and $\sum_{node_i \in C} s_i - b_0 \geq 0$, then $u_i \geq 0$. The utility of the coalition of a single member is either 0 or $-r_i$, and Lemma 1 is proved.

5. Cooperative game with complete information

Each player has common knowledge about the strategy space and utility of all other players in the game and has complete information.

Lemma 2. For a $N-player$ game, if $\sum_{node_i \in C} s_i \geq b_0$, then $(CP_0, CP_1, \ldots CP_N)$ represents a pure strategy Nash equilibrium, where $N = \max MN$, $node_1 \ldots node_N$ is a member of C-Coalition and CP_i represents the strategy of $node_i$ is CP.

Proof. For the edge node $node_0$, its utility is $Price - r_0$ when its strategy is cooperating. On the contrary, if it chooses to defect instead of cooperating unilaterally, its utility becomes zero, which is lower than the cooperation utility. Similarly, for the terminal device $node_i$, if it unilaterally deviates from cooperation to defect, its utility consistently remains lower than $r_i - s_i + \frac{-Price + \sum_{node_i \in C} s_i}{|C|} \geq 0$.

Therefore, when $\sum_{node_i \in C} s_i \geq b_0$, $(CP_0, CP_1, ...CP_N)$ is a pure strategy Nash equilibrium, Lemma 2 is proved.

Lemma 3. For a $N-player$ game, when $\sum_{node_i \in C} s_i < b_0$, all defect strategies are a pure-strategy Nash equilibrium.

Proof. When the terminal device $node_i$ chooses defect, its utility is equal to 0. If it chooses cooperation, its utility is less than 0, and Lemma 3 is proved.

Theorem 1. For a $N-player$ game, when coalition C exists, $\sum_{node_i \in C} s_i \geq b_0$ and $|C| \leq \max MN$, there exists at least one pure strategy Nash equilibrium.

Proof. Given a coalition C, there must be a coalition $C^* = \arg\max_{C^j \subseteq C} |C^j|$, where C^j represents the coalition. Therefore, the best response of the edge node $node_0$ is CP, i.e. $S^*_{node_0} = CP$. Let $S^*_{node_i} = \begin{cases} CP & if \quad node_i \in C^* \\ D & else \end{cases}$, then $(S^*_{node_0}, S^*_{node_1}, ...S^*_{node_N})$ be a pure strategy Nash equilibrium.

Lemma 2 indicates that for any $node_i \in C^*$ that deviates unilaterally from cooperation to defect, its utility is always lower than the cooperation utility, which means its utility does not increase. Similarly, for any $node_i \notin C^*$ that deviates unilaterally from defect to cooperation, its utility will change from 0 to negative. Therefore, all players are unwilling to change their strategies.

When the terminal devices request the edge node for computation offloading, a Nash bargaining game in the terminal devices and the edge node starts. The detailed process is shown in Algorithm 1.

Algorithm 1: Pure Strategy Nash Cooperative Game (PSNCG)	
1:	Edge node $node_0$ selects a proper weight w_{r0}; each terminal device $node_i$ $(1 \leq i \leq N)$ selects two proper weight w_{si0} and w_{si1}.
2:	Edge node $node_0$ calculates its cache resource D_0, CPU cycles C_0 and the required bandwidth $Bandwidth_0$, reservation price r_0 according to (5), choose a bidding price $b_0 (b_0 \geq r_0)$ and then broadcasts its relevant parameters (D_0, C_0 and $Bandwidth_0$) to terminal devices within its communication range through the trusted third party;
3:	FOR each terminal device (buyer) $node_i \in M$ DO

4:	Receive D_0 , C_0 and $Bandwidth_0$;
5:	Collect the related parameter ($Delay_i^l$, $Energy_i^l$ and $Bandwidth_i$), then calculate its reservation price r_i and choose an asking price s_i according to (6);
6:	END FOR
7:	Edge node $node_0$ calculates its bidding price b_0 and each terminal device $node_i \in M$ calculates its asking price s_i by adopting the equilibrium excursion method as in [18];
8:	Edge node $node_0$ submits b_0 , and each terminal device $node_i \in M$ submits s_i ; let $\mathrm{C} = \{C \in 2^N \mid \sum_{node_i \in C} s_i \geq b_0 \text{ and } \lvert C \rvert \leq \max MN\}$. IF $\lvert \mathrm{C} \rvert \geq 1$, a coalition C in $\arg \max_{C^j \in \mathrm{C}} \lvert C^j \rvert$ is chosen to participate in resource allocation;
9:	Finally, the utilities are allocated to nodes in the coalition C according to (10).

The computational complexity of the algorithm 1 is analyzed to evaluate its feasibility during runtime. The algorithm consists of four main stages:

Initialization involves the allocation of weights and basic parameters, resulting in a constant computational cost of *O(1)*.

Price setting requires each terminal device to calculate its reservation price independently. For *N* devices, this results in a total complexity of *O(N)*. The most computationally intensive part is the coalition formation and Nash bargaining stage. This process enumerates all possible subsets of terminal devices to identify feasible coalitions whose total offers meet or exceed the bidding price of edge node. The worst-case complexity of this exhaustive search is $O(2^N)$.

After determining the coalition, the stages of benefit distribution and resource allocation only involve basic arithmetic operations, maintaining the complexity of *O(1)*.

Therefore, the total running time of algorithm 1 is mainly influenced by the coalition selection step. Despite exponential growth in the worst-case scenario, this complexity is still acceptable for small and medium-sized MEC systems, as the number of active terminal nodes per edge server is relatively limited.

To enhance scalability, future implementations may incorporate heuristic or greedy coalition formation methods, threshold-based pruning, or parallel subset evaluations. These approaches could significantly reduce the exponential cost of coalition selection while maintaining near-optimal performance.

6. Performance evaluation

The PSNCG method is compared with RCFL [19] and CPFL [20], we note that RCFL and CPFL are not direct resource-pricing baselines but are widely used cooperative benchmarks in MEC and federated settings. They are included to demonstrate the comparative advantages of PSNCG in system-level metrics. Meanwhile, related MEC pricing schemes [9-11] are conceptually compared in Section 2 to provide readers with a broader perspective on pricing-based approaches.

Based on gradient-descent federated learning, RCFL optimizes the trade-off between local model updates and global aggregation under constrained resource budgets, minimizing the loss function in distributed learning tasks. CPFL introduces a synergistic cloud-edge framework for personalized federated learning, aiming to address device heterogeneity and reduce communication and computation, meeting the latency demands of IoT applications. In contrast, the PSNCG does not focus on joint learning optimization, but solves the incentive-compatible resource pricing and allocation in MEC through a bargaining-based cooperative game model. PSNCG enables interactive negotiation between terminal devices and edge servers for resource trading, incorporating Nash bargaining theory to achieve fairness and system-wide utility maximization. This theoretical design allows PSNCG to dynamically balance resource supply and demand in a decentralized MEC environment, which is beyond the scope of RCFL and CPFL.

The simulation environment consists of a MEC system with 10 edge nodes and 100 terminal devices uniformly distributed. Each edge node is connected to 10 terminal devices, which generate tasks with various service levels [21]. Task generation rate in individual devices follows a Poisson process, and the task load ranges from 1 to 5. 100 simulations are conducted to obtain system performance metrics, which are plotted as a function of task request load. The performance metrics include normalized device benefits, bandwidth utilization, and task success rate in the MEC system.

Figures 2-4 summarize the performance of the proposed MEC system under different scenarios. Fig. 2 shows the average task completion time under different offloading strategies, where the X-axis represents task generation rate (requests per unit time) and the Y-axis shows completion time (ms), illustrating the impact of offloading strategies on performance. Fig. 3 presents resource utilization across various MEC scenarios, with the X-axis representing the number of edge servers and the Y-axis indicating resource utilization (%), highlighting system efficiency under different configurations. Fig. 4 depicts system throughput versus task arrival rate, where the X-axis corresponds to task arrival rate (requests per unit time) and the Y-axis shows throughput (tasks per second), demonstrating how varying task loads affect overall system performance. Error bars and confidence intervals are not shown in the current figures but will be reported in future work.

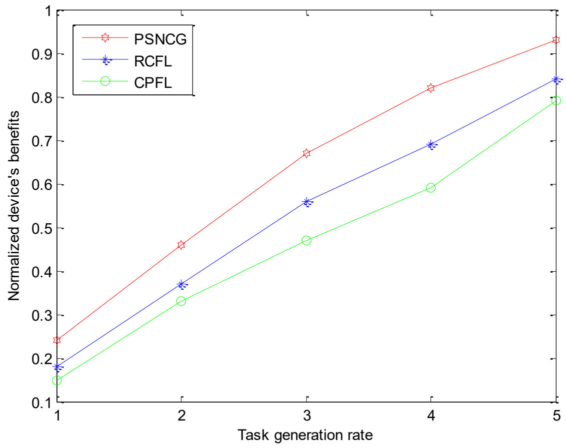

Fig. 2. Normalized device benefits and task generation rate under PSNCG, RCFL, and CPFL

In Fig. 2, the normalized benefits of devices are plotted against the task generation rate. As the number of task requests increases per device, more service is applied. The results demonstrate that the PSNCG can effectively manage resources under different task load from light to heavy, yielding higher device benefits within the MEC infrastructure. Notably, PSNCG operates interactively and strengthens the influence of the cooperative game model.

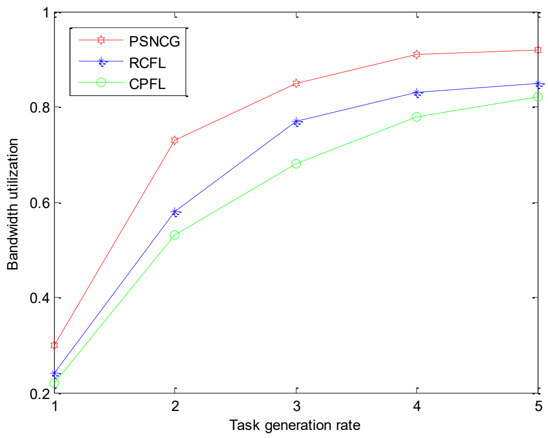

Fig. 3. Bandwidth utilization and task generation rate under PSNCG, RCFL, and CPFL

Fig. 3 shows the bandwidth utilization of task generation rate under different solutions. Compared with RCFL and CPFL, the PSNCG can ensures more stable and higher bandwidth utilization, achieving the bandwidth allocation process within the MEC system.

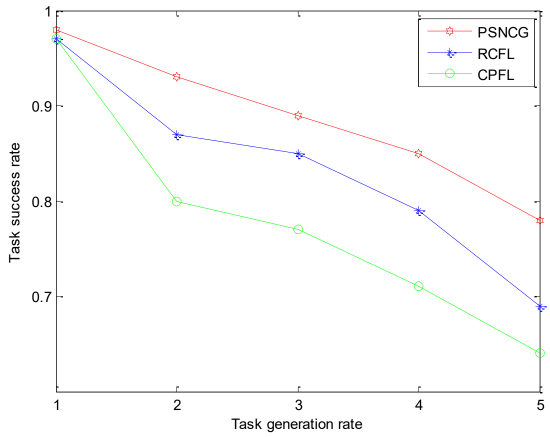

Fig. 4. Task success rate and task generation rate under PSNCG, RCFL, and CPFL

Fig. 4 shows the task success rate in these three schemes. Since the tasks are generated with their time constrained, system entities should fine tune the limited computation and communication resources to improve task success rate. The PSNCG can provide the optimal tradeoff for current system until the best solution has been found during the bargaining process. As shown in in Fig. 4, PSNCG can share system resources of different devices and obtain the maximum benefits while maintaining a rather higher task success rate than RCFL and CPFL.

From Figs. 2 to Fig. 4, numerical analysis is conducted to draw insights for validation. The bargaining approach PSNCG can achieve an appropriate performance balance in the MEC infrastructure. The results in Figs. 2 to 4 were obtained by averaging the results of 100 independent simulations for each setting. Although the observed performance gains of PSNCG relative to RCFL and CPFL are consistent in all experiments, statistical significance testing and confidence interval reporting will be incorporated in future work to provide more rigorous validation.

Current performance evaluation is based on a simulated MEC environment with uniformly distributed edge and terminal nodes. Although this setting provides a controlled baseline for comparing algorithms, real-world deployments involve varied and typically dynamic topologies. The current evaluation presents averaged results over 100 runs, which serve as proof-of-concept evidence. We acknowledge that future studies should incorporate confidence intervals, statistical significance tests, and sensitivity analyses of key parameters such as ε, reservation weights, and traffic heterogeneity to further substantiate robustness. In future work, the PSNCG scheme will be evaluated under diverse network structures such as clustered, grid-based, and random graph topologies, as well as using real-world traces from mobile networks or MEC benchmarks. In this way, the robustness and adaptability of the scheme under more realistic and heterogeneous scenarios will be validated.

7. Conclusion

An incentive mechanism that encourages edge service providers to participate in MEC task offloading by setting resource prices. A cooperative game model is proposed for resource allocation, aiming to maximize the profits of edge service providers while ensuring the quality of service for terminal devices. The Nash equilibrium with complete information game indicates that this model is reasonable. Compared with RCFL and CPFL, extensive simulations validate the performance of the proposed approach in normalized device advantages, bandwidth utilization, and task success rate. Future work will focus on designing a bargaining process that considers incentive-based interaction under incomplete information games. In this case, the resource allocation model can be formulated as a Bayesian game, where each participant has private information and forms beliefs about others. Under standard assumptions (finite type sets, compact strategy spaces, and continuous utility functions), the Bayesian Nash equilibrium can be established. This provides a theoretical basis for extending PSNCG to more realistic MEC scenarios with asymmetric information.

Beyond the single-node setting studied here, extending PSNCG to multi-edge cooperative MEC environments remains an important direction. Distributed bargaining and inter-node coordination will be incorporated in future research to enhance external validity.

Acknowledgement

This work was supported by the Project of West Anhui University High-level Talent Research Project (Grant No. WGKQ2025007), the Key Research and Development Technology Project of Anhui Province (Grant No. 2022l07020005), the Research and Demonstration of Key Technologies for Low Carbon Prefabricated Building Structures Based on 5G and BIM Technologies, and the Green Building and Digital Construction Integration Innovation Research Team (No. WXZR202402).

REFERENCES

[1] *X. Huang, L. He, X. Chen, G. Liu, F. Li,* A More Refined Mobile Edge Cache Replacement Scheme for Adaptive Video Streaming With Mutual Cooperation In Multi-Mec Servers, 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, pp.1-6, 2020.

[2] *M. Mukherjee, L. Shu, D. Wang,* Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges, IEEE Communications Surveys & Tutorials, Vol. **20**, Iss. 3, pp. 1826-1857, 2018.

[3] *Y. Y. Jiang, Y. X. Mao, G. X. Wu, Z. H. Cai, Y. X. Hao,* A collaborative optimization strategy for computing offloading and resource allocation based on multi-agent deep reinforcement learning, Computers and Electrical Engineering, Vol. **103**, pp. 108278, 2022.

[4] *C. Liang, Y. He, F. R. Yu, N. Zhao,* Enhancing video rate adaptation with mobile edge computing and caching in software-defifined mobile networks, IEEE Transactions on Wireless Communications, Vol. **17**, Iss. 10, pp. 7013-7026, 2018.

[5] *Z. Li, Z. Yang, S. Xie,* Computing Resource Trading for Edge-Cloud-Assisted Internet of Things, IEEE Transactions on Industrial Informatics, Vol. **15**, Iss. 6, pp. 3661-3669, 2019.

[6] *Z. Chang, W. Guo, X. Guo, Z. Zhou, T. Ristaniemi,* Incentive Mechanism for Edge-Computing-Based Blockchain, IEEE Transactions on Industrial Informatics, Vol. **16**, Iss. 11, pp. 7105-7114, 2020.

[7] *F. Wei, S. Chen, W. Zou,* A greedy algorithm for task offloading in mobile edge computing system, China Communications, Vol. **15**, Iss. 11, pp. 149-157, 2018.

[8] *Z. Zhu, J. Peng, X. Gu, H. Li, K. Liu, Z. Zhou, W. Liu,* Fair Resource Allocation for System Throughput Maximization in Mobile Edge Computing, IEEE Access, Vol. **6**, pp. 5332-5340, 2018.

[9] *Z. Zhou, B. Wang, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, M. Guizani,* Time-dependent pricing for bandwidth slicing under information asymmetry and price discrimination, IEEE Transactions on Communications, Vol. **68**, Iss. 11, pp. 6975-6989, 2020.

[10] *B. Baek, J. Lee, Y. Peng, S. Park,* Three dynamic pricing schemes for resource allocation of edge computing for IoT environment, IEEE Internet of Things Journal, Vol. **7**, Iss. 5, pp. 4292-4303, 2020.

[11] *S. Chen, B. C. Chen, X. Y. Tao, X. X, K. Q. Li,* An online dynamic pricing framework for resource allocation in edge computing, Journal of Systems Architecture, Vol. **133**, Iss. 102759, 2022.

[12] *G. Gao, M. Xiao, J. Wu, H. Huang, S. Wang, G. Chen,* Auction-Based VM Allocation for Deadline-Sensitive Tasks in Distributed Edge Cloud, IEEE Transactions on Services Computing, Vol. **14**, Iss. 6, pp. 1702-1716, 2021.

[13] *Q. Wang, S. Guo, J. Liu, C. Pan, L. Yang,* Profit Maximization Incentive Mechanism for Resource Providers in Mobile Edge Computing, IEEE Transactions on Services Computing, Vol. **15**, Iss. 1, pp. 138-149, 2022.

[14] *B. Dutta, A. Van den Nouweland, S. Tijs,* Link formation in cooperative situations, International Journal of Game Theory, Vol. **27**, Iss. 2, pp. 245-256, 1998.

[15] *Y. C. Guo, L.H. Yin, L.C. Liu, B.X. Fang,* Utility-based cooperative decision in cooperative authentication, IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 2014, pp. 1006-1014, 2014.

[16] *M. L. Dai, Z. B. Liu, S. Y. Guo, S. J. Shao, X. S. Qiu,* A computation offloading and resource allocation mechanism based on minimizing devices energy consumption and system delay, Journal of electronics & information technology, Vol. **41**, Iss. 11, pp. 2684-2690, 2019.

[17] *Q. V. Pham, H. T. Nguyen, Z. Han, W. J. Hwang,* Coalitional games for computation offloading in NOMA-enabled multi-access edge computing, IEEE Transactions on Vehicular Technology, Vol. **69**, Iss. 2, pp. 1982-1993, 2020.

[18] *C. Rao, Y. Zhao, J. Zheng, M. Goh, C. Wang,* Bidding Behavior and Equilibrium Excursion of Uniform Price Auction Mechanism, Asia Pacific Journal of Operational Research, Vol. **34**, Iss. 06, pp. 1750028, 2017.

[19] *S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, K. Chan,* Adaptive federated learning in resource constrained edge computing systems, IEEE Journal on Selected Areas in Communications, Vol. **37**, Iss. 3, pp. 1205-1221, 2019.

[20] *Q. Wu, K. He, X. Chen,* Personalized federated learning for intelligent IoT applications: A cloud-edge based framework, IEEE Open Journal of the Computer Society, Vol. **1**, pp. 35-44, 2020.

ITSG 12. Network Performance Objectives for IP-based Services [S]. ITU-T Y.1541, 2011.