2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

# Resource Sharing with Autonomous Agents in Cloud-Edge Computing Networks via Mechanism Design

Sajad Parvizi, Mina Montazeri and Hamed Kebriaei, *Senior Member, IEEE*

***Abstract*—Executing computation tasks through cloud–edge collaboration has emerged as a promising method to enhance the quality of service for applications. Typically, cloud servers and edge servers are selfish and rational. Therefore, it is crucial to develop incentive mechanisms that maximize cloud server profit and simultaneously motivate idle edge servers to participate in the task executing process while edge servers, as autonomous agents, choose their resource-sharing levels by themselves. This paper addresses the challenges of resource limitations and heterogeneity in edge computing by proposing a novel mechanism that integrates contract theory with Stackelberg game properties considering asymmetric information and the autonomous nature of edge servers. To propose an optimal mechanism, we design a linear form of reward function such that the mechanism's goals are met. The mechanism allows edge servers to autonomously decide their level of resource contribution while ensuring the maximization of the cloud server's utility. The proposed mechanism not only facilitates efficient resource utilization but also guarantees the truthful and rational participation of edge servers. Initially, the proposed mechanism is conceptualized as a non-convex functional optimization with a dual continuum of constraints. However, we illustrate that by deriving an equivalent representation of the constraints, it can be transformed into a convex optimal control problem. Simulation results demonstrate the efficiency of our proposed incentive mechanism approach.**

***Index Terms*—Cloud-Edge, resource sharing, autonomous edge servers, incentive mechanism, incomplete information.**

## I. Introduction

The advent of smart devices and the increasing demand for computationally intensive tasks have necessitated the development of more efficient computing solutions. Edge computing has emerged as a promising approach to address these challenges, offering a way to augment the computational capabilities of cloud servers, which often overload during peak demand periods and lead to a decrease in the Quality of Service (QoS) provided to users [1]. Recognizing the limitations inherent in cloud computing resources, the collaboration between cloud computing and edge computing has been identified as a critical strategy for significantly reducing computation latency and overall system costs [2], [3].

However, incorporating edge computing into existing cloud architectures presents its own set of challenges. Edge servers, unlike their cloud counterparts, are often limited by their computational resource and energy budgets [4]. These limitations are primarily due to the smaller processors used in edge devices and the diverse processor architectures they employ, making resource management a complex task [5]. The heterogeneity and resource limitations of edge servers necessitate innovative approaches to effectively utilize these idle resources to support cloud computing tasks. By doing so, it is possible to alleviate the computing overhead and reduce task latency on cloud servers, thereby optimizing the overall performance of the computing ecosystem [6].

Some studies have investigated the common scenario of cloud-edge collaboration, where the mobile users offload the computation tasks to the cloud server and the cloud server requests the assistance of the edge server when necessary [7], [8]. Most of them aim to improve the QoS of the system, i.e., reduce the computation latency [9], energy consumption [10], and more. However, considering the consumption of computation and energy resources, the edge servers as autonomous agents may be unwilling to share their limited resources without any incentive [11], [12]. Therefore, it becomes crucial to develop incentive mechanisms, which can encourage idle edge servers to participate in the resource-sharing process.

To design incentive mechanisms, game-theory-based research was developed to analyze the interactions between independent and selfish players aiming to maximize utilities for all players involved in the game [13]. Particularly, [14] and [15] model the interaction process between the cloud server and edge servers as a Stackelberg game, where the goal is to maximize the utility of both the edge servers and the cloud server. Although the cloud server, as the leader, provides the payment strategy and the edge servers, as the followers, determine the amount of computational work, it is assumed that the cloud server has complete information about the edge servers. The resource-sharing capability of an edge server depends on several factors, such as the residual computational resources and the execution cost based on hardware architecture. However, the above-mentioned factors

Sajad Parviziand Hamed Kebriaei are with the School of ECE, College of Engineering, University of Tehran, Tehran, Iran.
Mina Montazeri is with the School of ECE, College of Engineering, University of Tehran, Tehran, Iran, and also with the Urban Energy Systems Laboratory, Swiss Federal Laboratories for Materials Science and Technology, Dübendorf, Switzerland.

979-8-3503-1632-2/24/$31.00 ©2024 IEEE

are generally private information, available only to the edge server itself but not to the cloud server. In other words, the information is asymmetric. Contract theory, as a powerful theoretical tool, is employed to address incentive design problems in scenarios with asymmetric information [16]. The authors in [17], [18], and [19] investigated contract-based incentive mechanisms for computing resource sharing in edge computing networks. In these studies, the cloud server allocates a resource-sharing level and the corresponding incentive reward to each edge server as functions of their announced private information. This is done to motivate edge servers to participate in the resource-sharing process and ensure truthful announcing of the private information of edge servers. However, it is unlikely that edge servers, as autonomous agents, let the mechanism choose their resource-sharing levels for them. Motivated by the challenges mentioned in Stackelberg and contract theory methods, [20] develop a mechanism which ensures both truthful announcing of private information by the agent and their rational participation while the participation level is determined by the agent. Although [20] introduces a general mechanism, these two features make the mentioned mechanism an excellent choice for addressing resource-sharing problems in real-world scenarios.

In this study, inspired by [20], we develop a mechanism for addressing the resource-sharing problem. Specifically, we propose a computation latency minimization formulation by collaborating with edge servers and a cloud server to perform computational tasks. The cloud server, acting as the designer, designs only a reward function in the presence of information asymmetry to achieve the following goals: 1) Maximize the cloud server's utility function, 2) Allow edge servers to decide on their resource-sharing levels, and 3) Guarantee the participation of edges servers and also truthful announcing of edges servers' private information in the mechanism. We demonstrate that the cloud server can achieve these objectives by employing a linear form of the reward function, incorporating two decision functions. Following the design of the reward function by the cloud server, each edge server determines its optimal announced private information and resource-sharing level based on its actual private information and the incentive reward function communicated by the cloud server. Subsequently, each edge server receives a reward based on these two variables. To close to reality, we assume the private information of each edge server is drawn from a specific continuous distribution known to the cloud server. This scenario presents us with a non-convex double continuum of incentive constraints. We demonstrate that achieving the mechanism, which fulfills all previously mentioned properties, involves solving a constrained non-convex functional optimization problem. To solve this optimization, we derive a relationship between the decision functions of the cloud server, allowing us to reformulate the functional optimization as a convex optimal control problem.

The rest of this paper is structured as follows. The system model is described in Section II. Then, we formulate the mechanism for the resource-sharing problem in Section III. We introduce a special form of reward function in Section IV and illustrate that solving the optimal mechanism entails addressing a non-convex optimization problem. Furthermore, we demonstrate the transformation of this problem into an equivalent convex optimal control problem using mathematical reformulation techniques. Simulation results are demonstrated in Section V. Finally, Section VI presents the conclusion.

## II. SYSTEM MODEL

During peak times, computational tasks can be shared from the cloud server to edge servers and processed by edge servers with idle computing resources. These edge servers can include devices like smartphones or parked vehicles. To encourage these potential servers to participate and share their unused resources, cloud servers should create an efficient incentive mechanism that compensates for the cost of resource sharing by these edge servers.

The proposed cloud-edge framework is presented in Fig 1. We consider a scenario in which there exists one Cloud Server (CS) and $N$ Edge Servers (ESs). In order to obtain multiple resources of ESs, the CS will offer an incentive reward that can encourage the ESs to share computing resources to reduce the latency of computational tasks. However, the ESs who participate in edge computing networks decide on their computing resource sharing for doing the task. Furthermore, each ES has a private parameter in its utility function, referred to as the ES's "type", which is not known to the CS. Although it is feasible for ESs to share type, they can announce their type incorrectly to receive more incentive rewards, as the announced type may not necessarily match the actual type. Thus, the CS designs a reward function in which each ES receives a reward based on its announced type.

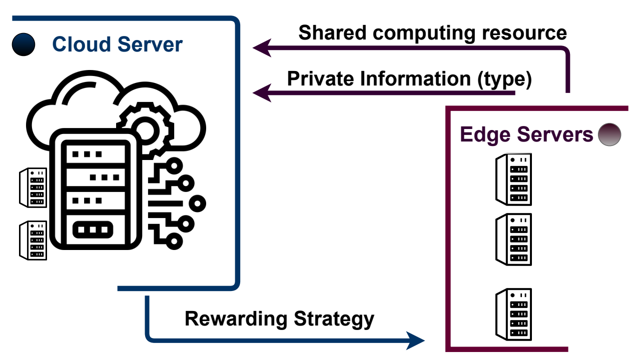


Fig. 1: The information flow in the Cloud-Edge framework

### A. Utility of ESs

Considering that the hardware architecture of the ESs's device to share the computing resource may be different from each other, they incur different costs in sharing their resources. With the different hardware architectures, the ESs can be considered to be heterogeneous. By defining $\theta$ as a parameter related to the hardware architecture of ESs' device, the ES utility function is formulated as follows [19]:

$$U(\theta, x, R(x, \hat{\theta})) = R(x, \hat{\theta}) - \theta\ c\ x^2, \tag{1}$$

where $x \in \mathbb{R}^+$ is the shared resources, $c$ is the unit cost of energy consumption, $\theta \in \Theta$ with $\Theta = [\underline{\theta}, \bar{\theta}]$ is the private information of the ES and is treated as its actual type, and $\hat{\theta} \in \Theta$ is the announced type which is not necessarily equal to the actual type, i.e. $\theta$. $R(x, \hat{\theta})$ is the incentive reward function that each ES receives from the CS. Although neither the CS nor other ESs don't know the ES's type, its cumulative distribution $F(\theta)$ is common knowledge.

### B. Utility of CS

By leveraging the amount of resources shared by ESs, the task processing delay can be reduced. The decreased processing delay that CS has for executing computational tasks is calculated as [19], [18]:

$$g(x) = \frac{D}{f_c} - \frac{D}{f_c + x}, \tag{2}$$

where $f_c$ is the amount of CS's computing resource, and $D$ represents the average total size of the computational task, which can be estimated based on the historical tasks. Finally, the CS's utility from the participation of each ES is as follows:

$$V(x, R(x, \hat{\theta})) = g(x) - R(x, \hat{\theta}) \tag{3}$$

## III. Problem Formulation

The objective of CS is to employ a suitable ES with sufficient computation resources to decrease task processing delay while maximizing the utility function defined in (3). Meanwhile, the goal of each ES is to optimize its utility defined in (1). However, by comparing equations (3) and (1), for the CS entity, $R(x, \hat{\theta})$ functions as a cost, impacting their utility negatively when maximizing it and conversely, for the ES entities, $R(x, \hat{\theta})$ represents a profit, positively influencing their utility when maximizing it. This discrepancy in the interpretation of the same variable as a cost for one party and a profit for the other intensifies the conflicting interests between the CS and ES entities, ultimately shaping their strategic decisions and interactions.

Meanwhile, since the private information (type) of each ES determines its cost and the amount of its resource sharing, it is essential for the CS to have accurate information to evaluate the performance of each ES. To motivate ESs to participate and guarantee ESs announce their types truthfully, the CS designs the incentive function $R(x, \hat{\theta})$ to ensure mutual benefit for both the CS and ES. Following is a definition of these constraints.

**Definition 1.** *A mechanism is Individually Rational (IR) if the ES's utility is non-negative by announcing its type truthfully:*

$$U(\theta, x, R(x, \theta)) \geq 0 \quad \forall \theta \in [\underline{\theta}, \bar{\theta}]. \tag{4}$$

Compared with the non-participate state in which the utility is always zero, each ES is willing to share resources in return for the incentive reward, as long as the IR constraint is satisfied.

**Definition 2.** *A mechanism is Incentive Compatible (IC) if the ES achieves equal or higher utility by announcing its type truthfully:*

$$U(\theta, x, R(x, \theta)) \geq U(\theta, x, R(x, \hat{\theta})) \qquad \forall \theta, \hat{\theta} \in [\underline{\theta}, \bar{\theta}]. \tag{5}$$

With IC inequality, for a self-interested ES with actual type $\theta$, truthful announcing of type is the best choice since this situation best fits into its actual type and brings it maximal utility. Therefore, the optimal profit maximization mechanism can be obtained by solving the following maximization problem [20]:

$$\max_{R(x, \hat{\theta})} \mathbb{E}_\theta[V(x, R(x, \hat{\theta}))], \tag{6a}$$

$$s.t. \quad x(\theta, r_1(\hat{\theta})) = \underset{\tilde{x}}{\operatorname{argmax}}\, U(\theta, \tilde{x}, R(\tilde{x}, \hat{\theta})) \tag{6b}$$

$$U(\theta, x, R(x, \theta)) \geq 0 \qquad \forall \theta \in [\underline{\theta}, \bar{\theta}], \tag{6c}$$

$$U(\theta, x, R(x, \theta)) \geq U(\theta, x, R(x, \hat{\theta})) \forall \theta, \hat{\theta} \in [\underline{\theta}, \bar{\theta}], \tag{6d}$$

where constraint (6b) represents the best response of ES to determine its sharing resource, and (6c),(6d) represent the IR and IC constraints, respectively.

## IV. Problem Solution

Solving optimization (6) is not straightforward due to the nonconvex double continuum constraint imposed by the (6d). Furthermore, unlike previous works in which CS had more flexibility in designing contract items and determining the ES's resource sharing, in the proposed work ESs decide on their participation level in sharing resources. For this reason, designing a typical optimal reward function identical to that found in the literature is not feasible, thus prompting us to consider a special form for the reward function. First, we consider a linear form for incentive reward function and then by presenting several lemmas and theorems obtain an equivalent simplified reformulation for optimization problem (6).

In the proposed profit maximization mechanism, the CS adopts the following form of the reward function [20]:

$$R(x, \hat{\theta}) \equiv r_1(\hat{\theta}) x + r_2(\hat{\theta}), \tag{7}$$

where $r_1(\hat{\theta})$ represents the reward factor of resource sharing subsidized by the CS and $r_2(\hat{\theta})$ represents the bias reward of CS to ES that only depends on ES's announced type.

**Proposition 1.** *Consider the constraint* (6b). *The optimal computing resource that each ES shared with CS to maximize its utility is given by:*

$$x(\theta, r_1(\hat{\theta})) = \frac{r_1(\hat{\theta})}{2c\theta}. \tag{8}$$

*Proof.* We can rewrite constraint (6b) as

$$x(\theta, r_1(\hat{\theta})) = \underset{\tilde{x}}{\text{argmax}}\ [r_1(\hat{\theta})\tilde{x} + r_2(\hat{\theta}) - \theta\ c\ \tilde{x}^2]. \quad (9)$$

For optimality, we set $\frac{\partial U(\theta,\tilde{x},R(\tilde{x},\hat{\theta}))}{\partial \tilde{x}} = 0$, which yields: $x(\theta, r_1(\hat{\theta})) = \frac{r_1(\hat{\theta})}{2c\theta}$. Moreover, since $\frac{\partial^2 U(\theta,\tilde{x},R(\tilde{x},\hat{\theta}))}{\partial \tilde{x}^2} = -2c\theta < 0$, the second derivative is negative. □

Replacing $R(x, \hat{\theta})$ from equation (7) and $x(\theta, r_1(\hat{\theta}))$ from equation (8), Optimization (6) can be rewritten as following optimization problem.

$$\max_{r_1(\hat{\theta}),r_2(\hat{\theta})} \int_{\underline{\theta}}^{\bar{\theta}} [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{2c\hat{\theta}} - r_2(\hat{\theta})]f(\hat{\theta})d\hat{\theta}, \quad (10a)$$

$$s.t. \quad \frac{r_1^2(\theta)}{4c\theta} + r_2(\theta) \geq 0, \qquad \forall\theta \in [\underline{\theta}, \bar{\theta}], \quad (10b)$$

$$\frac{r_1^2(\theta)}{4c\theta} + r_2(\theta) \geq \frac{r_1^2(\hat{\theta})}{4c\theta} + r_2(\hat{\theta}), \qquad \forall\theta, \hat{\theta} \in [\underline{\theta}, \bar{\theta}]. \quad (10c)$$

**Lemma 1.** *In optimization* (10)*, the IR constraint is satisfied for all* $\theta$ *if IR is binding for* $\theta = \bar{\theta}$*, which means* $U(\bar{\theta}, x, R(x, \bar{\theta})) = 0$.

*Proof.* Differentiating $U$ with respect to $\theta$, we have:

$$\frac{dU}{d\theta} = \frac{dU}{dx}\frac{dx}{d\theta} - cx^2. \quad (11)$$

The first term of equation (11) equals zero due to the first-order condition and hence, we have: $\frac{dU}{d\theta} \leq 0$. Thus, using IC constraint, we have:

$$\begin{aligned} U(\theta, x, r_1(\theta), r_2(\theta)) &\geq U(\theta, x, r_1(\bar{\theta}), r_2(\bar{\theta})) \\ &\geq U(\bar{\theta}, x, r_1(\bar{\theta}), r_2(\bar{\theta})). \end{aligned} \quad (12)$$

Furthermore, IR should be binding, otherwise, we could decrease $r_2(\theta)$ for all $\theta \in [\bar{\theta}, \bar{\theta}]$ by $\epsilon > 0$, which would satisfy all constraints of (10) and also increase the utility of CS. □

**Theorem 1.** *The solution of* $(r_1(\hat{\theta}), r_2(\hat{\theta}))$ *in optimization* (10) *is IC if and only if both of the following conditions hold:*

$$r_1'(\hat{\theta}) \leq 0 \qquad \forall\hat{\theta} \in [\underline{\theta}, \bar{\theta}], \quad (13)$$

$$r_2(\hat{\theta}) = \int_{\theta}^{\bar{\theta}} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y} dy - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}} \quad (14)$$

*Proof.* We divide the proof of this theorem into two parts, the forward direction "If", and the backward direction, "Only If". To show "If" part, according to relation (14), IC constraint (10c) can be re-written as:

$$\begin{aligned} &\frac{r_1^2(\theta)}{4c\theta} + \int_{\theta}^{\bar{\theta}} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}dy - \frac{r_1^2(\theta)}{4c\theta} \geq \\ &\frac{r_1^2(\hat{\theta})}{4c\theta} + \int_{\hat{\theta}}^{\bar{\theta}} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}dy - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}}. \end{aligned} \quad (15)$$

For $\theta > \hat{\theta}$, (15) can be re-written as:

$$\int_{\hat{\theta}}^{\theta} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}dy \leq \frac{r_1^2(\hat{\theta})}{4c}(\frac{1}{\hat{\theta}} - \frac{1}{\theta}). \quad (16)$$

Moreover, for $\theta < \hat{\theta}$, we have:

$$\int_{\theta}^{\hat{\theta}} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}dy \geq \frac{r_1^2(\hat{\theta})}{4c}(\frac{1}{\theta} - \frac{1}{\hat{\theta}}), \quad (17)$$

which both equations (16) and (17) hold true due to the monotonicity of $r_1(\theta)$.

To show the "Only If" part, first we prove that truthfulness implies monotonicity of $r_1(\theta)$. Considering two different mathematical interpretations of IC constraint; one where an ES of type $\theta$ announces its type as $\hat{\theta}$, denoted by $IC_{\theta,\hat{\theta}}$; and another where an ES of type $\hat{\theta}$ announces its type as $\theta$, denoted by $IC_{\hat{\theta},\theta}$, we have:

$$\frac{r_1^2(\theta)}{4c\theta} + r_2(\theta) \geq \frac{r_1^2(\hat{\theta})}{4c\theta} + r_2(\hat{\theta}) \qquad (IC_{\theta,\hat{\theta}}), \quad (18)$$

$$\frac{r_1^2(\hat{\theta})}{4c\hat{\theta}} + r_2(\hat{\theta}) \geq \frac{r_1^2(\theta)}{4c\hat{\theta}} + r_2(\theta) \qquad (IC_{\hat{\theta},\theta}). \quad (19)$$

By subtraction of equations (18) and (19), we get:

$$\frac{r_1^2(\theta) - r_1^2(\hat{\theta})}{4c}(\frac{1}{\theta} - \frac{1}{\hat{\theta}}) \geq 0,$$

where if $\theta > \hat{\theta}$ then $r_1(\theta) \leq r_1(\hat{\theta})$ , which implies monotonicity of $r_1(\theta)$. To derive equation (14), we can rearrange equations (18) and (19) as follows:

$$\frac{r_1^2(\hat{\theta}) - r_1^2(\theta)}{4c\theta} \leq r_2(\theta) - r_2(\hat{\theta}) \leq \frac{r_1^2(\hat{\theta}) - r_1^2(\theta)}{4c\hat{\theta}} \quad (20)$$

by considering $\hat{\theta} = \theta + \epsilon$ and dividing throughout equation (20) by $\epsilon$, and letting $\epsilon \to 0$, we have:

$$\frac{1}{4c\theta}\frac{d}{d\theta}[r_1^2(\theta)] \leq -\frac{d}{d\theta}r_2(\theta) \leq \frac{1}{4c\theta}\frac{d}{d\theta}[r_1^2(\theta)] \quad (21)$$

Thus,

$$\frac{d}{d\theta}r_2(\theta) = -\frac{1}{4c\theta}\frac{d}{d\theta}[r_1^2(\theta)]. \quad (22)$$

Integrating equation (22) with respect to $\theta$ from $\theta$ to $\bar{\theta}$ we have:

$$r_2(\bar{\theta}) - r_2(\theta) = \int_{\theta}^{\bar{\theta}} [-\frac{1}{4cy}\frac{d}{dy}(r_1^2(y))]dy. \quad (23)$$

So we have:

$$r_2(\theta) = \int_{\theta}^{\bar{\theta}} [\frac{1}{4cy}\frac{d}{dy}(r_1^2(y))]dy + r_2(\bar{\theta}). \quad (24)$$

Considering $U(\bar{\theta}, x(\bar{\theta}, r_1(\bar{\theta})), r_1(\bar{\theta}), r_2(\bar{\theta})) = 0$ from Lemma 1, the integration by parts of equation (24) gives us:

$$r_2(\theta) = \int_{\theta}^{\bar{\theta}} \frac{r_1^2(y)}{4cy^2}dy - \frac{r_1^2(\theta)}{4c\theta}. \quad (25)$$

□

Thus, optimization (10) can be rewritten as follows:

$$\max_{r_1(\hat{\theta}),r_2(\hat{\theta})} \int_{\underline{\theta}}^{\bar{\theta}} [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{2c\hat{\theta}} - r_2(\hat{\theta})]f(\hat{\theta})d\hat{\theta}, \quad (26)$$

$$s.t. \quad r_1'(\hat{\theta}) \leq 0 \qquad \forall\hat{\theta} \in [\underline{\theta}, \bar{\theta}],$$

$$r_2(\hat{\theta}) = \int_{\theta}^{\bar{\theta}} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}dy - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}}.$$

**Definition 3.** $h(t) \equiv \frac{f(t)}{1-F(t)}$ *is known as the hazard rate of* $t$ *in the statistics literature [21] that captures the probability at which an event is expected to occur at a time* $t$*, considering that it has not been taken place yet.*

**Proposition 2.** *The optimal solution to the optimization problem in* (26) *is the same as the following optimization solution.*

$$\max_{r_1(\hat{\theta})} \int_{\underline{\theta}}^{\bar{\theta}} [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}} + \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}^2}\frac{1}{h(\hat{\theta})}]f(\hat{\theta})d\hat{\theta}, \tag{27a}$$

$$s.t.\ r_1'(\hat{\theta}) \leq 0, \tag{27b}$$

*Proof.* Replacing $r_2(\hat{\theta})$ from equation (26) in the utility function of CS, we have:

$$V = \int_{\underline{\theta}}^{\bar{\theta}} [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{2c\hat{\theta}} - \int_{\hat{\theta}}^{\bar{\theta}} [\frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}]dy + \frac{r_1^2(\theta)}{4c\theta}]f(\hat{\theta})d\hat{\theta} \tag{28}$$

The integration by parts of the term $\int_{\underline{\theta}}^{\bar{\theta}} \int_{\underline{\theta}}^{\theta} [k_\theta(x(y, r_1(y)), y)dy]f(\hat{\theta}_i)d\hat{\theta}_i$ gives us:

$$\int_{\underline{\theta}}^{\bar{\theta}} \left[\int_{\hat{\theta}}^{\bar{\theta}} \frac{r_1^2(\hat{\theta})}{4c\theta^2}|_{\hat{\theta}=\theta=y}dy\right] f(\hat{\theta})d\hat{\theta} = \int_{\underline{\theta}}^{\bar{\theta}} \frac{r_1^2(\hat{\theta})}{4c(\hat{\theta})^2}\frac{1-F(\theta)}{f(\theta)}f(\hat{\theta})d\hat{\theta}. \tag{29}$$

Considering definition of $h(\hat{\theta})$ in Definition 3, equation (28) can be rewritten as follow:

$$V = \int_{\underline{\theta}}^{\bar{\theta}} [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}} + \frac{r_1^2(\hat{\theta})}{4c(\hat{\theta})^2}\frac{1}{h(\hat{\theta})}]f(\hat{\theta})d\hat{\theta}. \tag{30}$$

□

Notice that, optimization problem (27) is formulated as a calculus of variations problem [22]. By defining the variable $u(\hat{\theta}) = r_1'(\hat{\theta})$, optimization Problem (27) appears in the form of an optimal control problem in dynamic optimization, with state variables $r_1(\hat{\theta})$ and control variables $u(\hat{\theta})$. Subsequently, the optimal control problem can be formulated as follows:

$$\max_{r_1(\hat{\theta})} \int_{\underline{\theta}}^{\bar{\theta}} [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}} + \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}^2}\frac{1}{h(\hat{\theta})}]f(\hat{\theta})d\hat{\theta}, \tag{31a}$$

$$s.t.\ r_1'(\hat{\theta}) = u(\hat{\theta}), \tag{31b}$$

$$u(\hat{\theta}) \in \mathbb{U} \equiv [\underline{u}, 0], \tag{31c}$$

where $\underline{u} < 0$ is a large (finite but otherwise arbitrary) control constraint. We can solve it by calculating the Hamiltonian function as follows:

$$H(\hat{\theta}, r_1(\hat{\theta}), u(\hat{\theta}), \lambda(\hat{\theta})) = [\frac{D}{f_c} - \frac{2c\hat{\theta}D}{r_1(\hat{\theta}) + 2c\hat{\theta}f_c} - \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}} + \frac{r_1^2(\hat{\theta})}{4c\hat{\theta}^2}\frac{1}{h(\hat{\theta})}]f(\hat{\theta}) + \lambda u(\hat{\theta}), \tag{32}$$

where $\lambda$ is a Lagrange multiplier. As proved in [23], given that ${r_1}'(\hat{\theta})$ represents a linear function of $r_1(\hat{\theta})$ and $u(\hat{\theta})$, and considering that the cost function of optimization (31) is concave while $\mathbb{U}$ constitutes a convex set, $r_1(\hat{\theta})$ are the solution of the optimization (31) if and only if the prescribed conditions, recognized as the Minimum Principle, are satisfied.

TABLE I: Utility of cloud server for different total sizes of computational tasks

| | D = 200 | D = 250 | D = 300 | D = 350 | D = 400 |
|---|---|---|---|---|---|
| Utility of cloud server | 1.15 | 0.95 | 0.57 | 0.47 | 0.38 |

$$\dot{r_1}(\hat{\theta}) = \frac{\partial}{\partial \lambda}H(\hat{\theta}, r_1(\hat{\theta}), u(\hat{\theta}), \lambda(\hat{\theta})), \tag{33a}$$

$$\dot{\lambda}(\hat{\theta}) = \frac{\partial}{\partial r_1}H(\hat{\theta}, r_1(\hat{\theta}), u(\hat{\theta}), \lambda(\hat{\theta})), \tag{33b}$$

$$u(\hat{\theta}) = \arg\min_{u \in \mathbb{U}} H(\hat{\theta}, r_1(\hat{\theta}), u(\hat{\theta}), \lambda(\hat{\theta})), \tag{33c}$$

$$\lambda(\bar{\theta}) = 0. \tag{33d}$$

Numerical iterative algorithms can be utilized to solve this optimal control problem. In particular, we employ the Gradient Projection Algorithm to address problem (33) [24]. The solution of problem (33) concludes to the optimal reward factor $r_1(\hat{\theta})$. Also, by substituting optimal $r_1(\hat{\theta})$ in (14), the optimal bias reward $r_2(\hat{\theta})$ is obtained.

## V. Simulation

In this section, we provide numerical results to validate the proposed mechanism. The unit cost of energy consumption is c = 1. The computing resource of the CS is denoted as fc = 10GHz [18], [17]. Assume that the types of ESs, $\theta$ are distributed in range [4, 6] uniformly.

Figure 2 shows the optimal utility, reward, and shared resource of various edge server types, considering different total sizes of computational tasks. the task size of CS is set with different values from set D = [200 250 300 350 400]GB. As shown in Figure 2.(a), with the increase of task size D, the larger reduced execution latency will be achieved, leading to more utility of the CS. On the contrary, as shown in Table I, the utility of cloud server decreases by increasing of task size D. Figure 2.(a) also shows that $U(\bar{\theta}, x, R(x, \bar{\theta})) = 0$ which is consistent with Lemma 1. Figure 2 shows all the optimal utility, reward, and shared resource decrease with the edge servers' type in the proposed mechanism. Therefore, the edge servers can choose the hardware with lower $\theta$ to increase their utility. Figure 2(c) also presents that the first variable of the reward function (i.e. $r_1$) decreases with the increase of edge servers' type, which is consistent with $r_1'(\hat{\theta}) \leq 0$ in Proposition 2.

## VI. Conclusion

In this paper, we successfully address the challenges of resource limitations and heterogeneity in edge computing by developing an incentive mechanism that utilizes the principles of contract theory and Stackelberg game properties. This mechanism is designed to handle the asymmetric information and autonomous nature of edge servers, motivating their participation in computation task execution through a well-structured reward function. By allowing edge servers to decide their level of resource contribution autonomously, the mechanism ensures the maximization of the cloud server's utility while promoting efficient resource utilization and truthful participation from the edge servers. We overcame the inherent nonconvexity of the optimization problem derived from the proposed mechanism by employing a mathematical

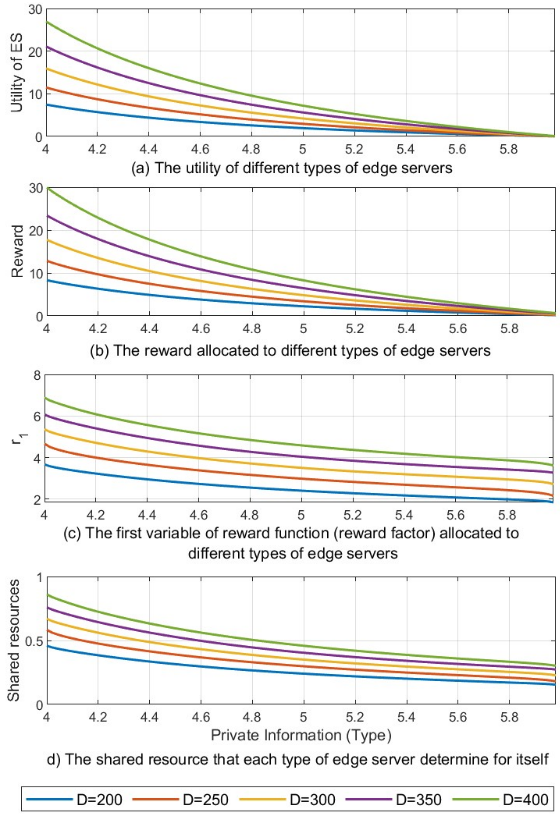


Fig. 2: Utility, reward, and shared resource across various edge server types, considering different total sizes of computational tasks.

reformulation technique, transforming the problem into an equivalently convex optimal control problem. The simulation results further validate the effectiveness of the proposed incentive mechanism, demonstrating its potential to enhance the quality of service in cloud-edge collaborative environments significantly.

## REFERENCES


[1] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2016.

[2] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[3] Y. Li, B. Yang, H. Wu, Q. Han, C. Chen, and X. Guan, "Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-stackelberg approach," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 969–15 982, 2022.

[4] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–37, 2019.

[5] H. Gu, L. Zhao, Z. Han, G. Zheng, and S. Song, "Ai-enhanced cloud-edge-terminal collaborative network: Survey, applications, and future directions," *IEEE Communications Surveys & Tutorials*, 2023.

[6] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1227–1240, 2020.

[7] H. Hu, D. Wu, F. Zhou, X. Zhu, R. Q. Hu, and H. Zhu, "Intelligent resource allocation for edge-cloud collaborative networks: A hybrid ddpg-d3qn approach," *IEEE Transactions on Vehicular Technology*, 2023.

[8] H. Yuan and M. Zhou, "Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1277–1287, 2020.

[9] T. Liu, L. Fang, Y. Zhu, W. Tong, and Y. Yang, "A near-optimal approach for online task offloading and resource allocation in edge-cloud orchestrated computing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2687–2700, 2020.

[10] J. Chen, Z. Chang, X. Guo, R. Li, Z. Han, and T. Hämäläinen, "Resource allocation and computation offloading for multi-access edge computing with fronthaul and backhaul constraints," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8037–8049, 2021.

[11] S. Luo, X. Chen, Z. Zhou, X. Chen, and W. Wu, "Incentive-aware micro computing cluster formation for cooperative fog computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2643–2657, 2020.

[12] G. Li and J. Cai, "An online incentive mechanism for collaborative task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 624–636, 2019.

[13] T. Mahn, M. Wirth, and A. Klein, "Game theoretic algorithm for energy efficient mobile edge computing with multiple access points," in *2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 2020, pp. 31–38.

[14] H. Zhou, Z. Wang, N. Cheng, D. Zeng, and P. Fan, "Stackelberg-game-based computation offloading method in cloud–edge computing networks," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 510–16 520, 2022.

[15] J. Lee, D. Kim, and D. Niyato, "Market analysis of distributed learning resource management for internet of things: a game-theoretic approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8430–8439, 2020.

[16] S. Parvizi, M. Montazeri, and H. Kebriaei, "Contract-based demand response mechanism for commercial and industrial customers," *IEEE Transactions on Automation Science and Engineering*, 2024.

[17] C. Su, F. Ye, T. Liu, Y. Tian, and Z. Han, "Computation offloading in hierarchical multi-access edge computing based on contract theory and bayesian matching game," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 686–13 701, 2020.

[18] Z. Zhou, H. Liao, B. Gu, S. Mumtaz, and J. Rodriguez, "Resource sharing and task offloading in iot fog computing: A contract-learning approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 227–240, 2019.

[19] N. Zhao, W. Du, F. Ren, Y. Pei, Y.-C. Liang, and D. Niyato, "Joint task offloading, resource sharing and computation incentive for edge computing networks," *IEEE Communications Letters*, vol. 27, no. 1, pp. 258–262, 2022.

[20] M. Montazeri, H. Kebriaei, and B. N. Araabi, "A tractable truthful profit maximization mechanism design with autonomous agents," *IEEE Transactions on Automatic Control*, 2024.

[21] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, "Algorithmic game theory, 2007," *Google Scholar Digital Library Digital Library*.

[22] A. Calogero *et al.*, "Notes on optimal control theory with economic models and exercises," 2013.

[23] D. Bertsekas, *Dynamic programming and optimal control: Volume I*. Athena scientific, 2012, vol. 1.

[24] J. Preininger and P. T. Vuong, "On the convergence of the gradient projection method for convex optimal control problems with bang–bang solutions," *Computational Optimization and Applications*, vol. 70, 2018.