Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7444916, 13 pages
https://doi.org/10.1155/2022/7444916

Research Article

Private Computing Offloading in Edge Cloud via Collaborative Online Learning

Lin Wang,[1] **Lei Yang**,[2] **Mingchuan Zhang**,[1] **Jianxin Zhang**,[3] **Zhibin Cao**,[2] **and Qingtao Wu**[1]

[1]*School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China*
[2]*CITIC Heavy Industries Corporation Limited, Information Technology Management Center, Luoyang 471003, China*
[3]*Academy of Industrial Internet Security, Beijing Qihu Technology Company Ltd., Beijing 100088, China*

Correspondence should be addressed to Mingchuan Zhang; zhang_mch@haust.edu.cn

Received 14 April 2022; Revised 16 June 2022; Accepted 27 June 2022; Published 14 July 2022

Academic Editor: Qiang Ye

Computing offloading based on mobile edge computing (MEC) for mobile devices (MDs) has received great attentions in recent years. Strategy selection is an extremely important part of computing offloading, so how to make an optimal decision quickly and accurately during the computing offloading is a difficult point. Furthermore, MDs are likely to leak personal privacy when interacting with edge cloud, and there is also an issue about commercial privacy leakage between different cloud service suppliers. In this paper, we propose the privacy-protected edge cloud computing offloading (EPCO) algorithm based on online learning to improve the efficiency of computing offloading while ensuring the privacy of system users. Simultaneously, EPCO also supports different MDs customize their privacy level. We prove that adding privacy protection mechanism is almost no effect on the convergence of the algorithm. The simulation results validate our conclusion using a real-world dataset.

1. Introduction

Mobile devices (MDs) have become extremely popular in recent years due to their mobility and convenience [1, 3]. Meanwhile, the functionality of the application for MDs becomes increasingly powerful [3], which leads to lack of local resources of MDs, such as computing resources, storage, and energy [4, 5]. To this end, computing offloading for MDs has emerged. Researchers proposed mobile cloud computing (MCC) that source starvation can be resolved by sending computing tasks of MDs to remote cloud for execution [6, 7]. However, since cloud servers are often far away from MDs, the data needs to be transmitted for a long distance, which results in a long response time. To this end, much research in recent years has focused on mobile edge computing (MEC) [8], which sends computing tasks to edge cloud servers (ECSs) [9]. ECSs are typically deployed around MDs, which enables a short physical distance between MDs and servers, resulting a shorter latency [10]. The work of this paper is based on the edge cloud network.

Strategy selection is an important part of computing offloading between ECSs and MDs [1]. When a MD decides to offload its computing tasks to an ECS, it must first make a decision to select an optimal server for computing offloading. Researchers have used game theory in past research to solve the problem of selecting servers for computing offloading [11], which was also significantly effective at the time. However, with the increasing demands of users on the quality of network services and the challenges of big data [12, 13], most of the previous studies are outdated. In recent years, online learning algorithms have been greatly developed and used in various fields to help improve system's performance [14–18]. Therefore, we consider using online learning algorithms to solve strategy selection problem of computing offloading. Furthermore, not only the efficiency of computing offloading should be considered, but the privacy of system users should also be concerned. However, few researches involve the above two aspects.

Privacy protection is an important part of the computing offloading [19]. The privacy issues we consider include the following two parts: the privacy of the MDs and the service suppliers. On the one hand, a MD's privacy may be exposed during data transmission or forwarding if there is a malicious third party involved. The malicious third party can infer the characteristics of the MD by accessing the computing offloading records. For example, a large amount of computation indicates the importance of the MD and the distance item exposes the location of the MD. With multiple of these side information, it is possible to identify a user in real world. For instance, He et al. [19] proposed a privacy-aware task offloading algorithm, which enabled low delay and energy consumption while maintaining an appropriate level of privacy. Min et al. [20] proposed a privacy-aware offloading algorithm that can improve the offloading performance, save energy, and enable privacy of healthcare IoT devices. Although these studies focus on the privacy issue in offloading, they only avoid the possibility of privacy leakage through some transmission method, so the privacy protection effect of these method is limited. On the other hand, since there is commercial competition between service suppliers, the privacy between them should also be considered. Therefore, privacy protection is another important part of the computing offloading. However, the research that considers both strategy selection and privacy protection is barely known.

To overcome above challenge, we consider introducing differential privacy into our computing offloading scenario. Differential privacy which proposed by Dwork et al. [21] has received great attention in the field of privacy protection in recent years. Differential privacy uses random noise to ensure that the private information of the individual will not be disclosed when the result of a query requests to disclose visible information. Zhang et al. and Hassan et al. [22, 23] use differential privacy techniques to address the risk of privacy leakage in their systems. This paper is motivated by the unresolved privacy risks in computing offloading scenarios. The privacy of mobile device users, such as location and device usage, may be leaked through data exchange during the offloading process, which is a potential privacy breach risk for users. According to the research of He et al. [19] and Min et al. [20], privacy risk has indeed become an important issue in computing offloading. Although they paid attention to the privacy issues in computing offloading, these algorithms only avoided the possibility of privacy leakage through a certain transmission method and did not fundamentally solve the problem of privacy risks. Therefore, we introduce differential privacy technology and propose an algorithm EPCO that protects the privacy of multiparty users in computing offloading. In this paper, the system structure is shown in Figure 1. For instance, a healthcare device needs to compute a large amount of monitoring data, so part of the computing tasks should be offloaded to the edge cloud for computing. The device sends the encrypted offloading data to the edge cloud, and then, each service supplier in the edge cloud gives an optimal offloading plan through online learning algorithms, and finally, the device makes a decision. Our main contributions are as follows:

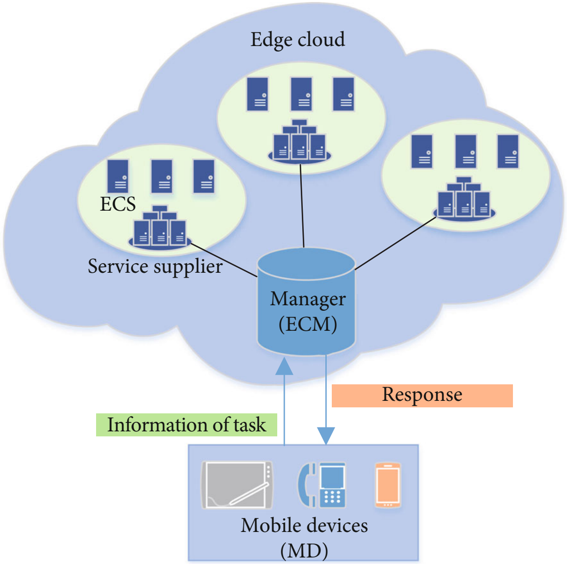


Figure 1: The structure of edge cloud in this paper.

(i) In this paper, we propose EPCO algorithm for MD and edge cloud to perform computing offloading based on online learning and differential privacy technology

(ii) EPCO preserves the privacy of both MDs and service suppliers. Moreover, we support different MDs to customize their own privacy protection levels. We proved it through theoretical derivation

(iii) We verified the theory through simulation experiments. The results show that EPCO guarantees the efficiency of computing offloading while protecting the privacy of system users

2. Related Work

In this section, we introduce related work from two aspects: optimal offloading and privacy management.

2.1. Optimal Offloading. Computing offloading alleviates the limitation of MDs' resources by sending computing tasks to the remote cloud for execution [24]. There has been a lot of research work on computing offloading in the past two decades [25–27]. Selecting ECSs for MDs is an indispensable part of computing offloading [28]. Previous decisions about server selection usually used the game theory method, which mainly concerned about energy conservation, network environment perception, and so on. Jošilo and Dán [29] made the decision to choose the wireless access points for mobile users during the computing offloading process. They proved that there is a Nash equilibrium in the model they propose, which can maximize the benefits of all users. However, this solution requires multiple interactions between users and computing resources to obtain better results, which is very unfriendly to time-sensitive applications. Barrameda and Samaan [30] considered to use tree execution dependency trees to enhance the accuracy, which is an important technical indicator in computing offloading. However, this solution is designed for a central cloud with a large number of computing resources. First, it needs to use a large amount of additional computing to run the algorithm, and secondly, the service response time cannot be guaranteed. Although these studies have solved some specific problems, they are gradually unable to adapt to the more complicated situation such as the challenges of big data and the personalized needs of users.

In recent years, online learning has been used in various fields to help improve system performance. Shahrampour et al. [31] used online learning in object recognition to help identify the current object through historical information from other modes. Sakulkar and Krishnamachari [32] proposed two online learning algorithms to help them solve the power allocation problem modelled as a Markov decision process. We consider applying online learning to the scenario of computing offloading, which can help us further improve the efficiency of computing offloading. Cao and Cai [33] used machine learning technology to solve the decision problems of MD for achieving Nash equilibrium points in the noncooperative game that they proposed, which shows the prospect of machine learning in computing offloading research. Therefore, these studies bring us new ideas for strategy selection of computing offloading. We consider that if we try to apply online learning theory and technology to the scenario of computing offloading, this can help us further improve the efficiency of computing offloading.

2.2. Privacy Management. As more and more people pay attention to personal privacy, the issue of user privacy protection should also be considered in computing offloading. Differential privacy is a popular research in recent years in terms of privacy protection. Differential privacy was first proposed by Dwork [34] and gave provable differential privacy protection. In recent years, people have realized the importance of privacy, and privacy has been used in various research fields to protect users' privacy [35, 36]. Shin et al. [37] proposed a novel matrix factorization algorithm that guarantees per-user privacy under local differential privacy. In addition, they reduced communication overhead between the server and users by dimensionality reduction. Piao et al. [38] proposed an algorithm that can reduce query sensitivity and improved the effectiveness of published data. The above researches on differential privacy only provide same level of privacy protection, which is not practical in many applications. Dobbe et al. [39] proposed a customized local differential privacy mechanism to solve the privacy protection problem in multiagent distributed optimization problems. They proposed an approach for determining sensitivity, and they derived analytical bounds for some quadratic problems. The customizable ideas mentioned here have been adopted by us. In this paper, we allow different MDs to customize their privacy protection levels.

Since few researchers have paid attention to the privacy protection of computing offloading before, there are not many related research contributions, but we will continue to pay attention to the research progress in this area.

3. System Model

3.1. Computing Offloading. Consider an edge cloud network with a set $\mathscr{M} = \{1, \cdots, M\}$ of mobile device and an edge cloud manager (ECM) that manages a set $\mathscr{V} = \{1, \cdots, V\}$ of service suppliers. Each supplier has a set $\mathscr{S} = \{1, \cdots, S\}$ of servers that can provide computing service. Each mobile device $i \in \mathscr{M}$ has a task that has been determined to perform computing offloading. As shown in Figure 1, at each slot t, a MD sends a d-dimensional context in $\mathscr{X} \coloneqq [0, 1]^d$ denoted by x_t to all suppliers, where x_t is added to Laplace noise based on the privacy protection requirements of different MDs. By receiving x_t, the ECM first broadcasts it to all suppliers. Each supplier then selects an optimal ECS and sends the information of this ECS to the ECM. The ECM provides the MD with the optimal ECS in this network that denoted by s_t. Subsequently, the MD decides whether to perform computing offloading.

Each $s_t \in \mathscr{S}$ has a two-dimensional vector denoted by $\mathbf{w}_t = (w_t^p, w_t^q)$, where w_t^p and w_t^q denote the reward in the price and reliability for performing computing offloading, respectively. w_t^p and w_t^q are given by $w_t^p = y_{s_t}^p(x_t) + \pi_t^p$ and $w_t^q = y_{s_t}^q(x_t) + \pi_t^q$, respectively, where $y_{s_t}^k(x_t)$, $k \in \{p, q\}$ denotes the expected reward of selecting ECS s in k given context x. π_t^k, $k \in \{p, q\}$ is a random noise, which satisfies $\mathbb{E}[\pi_t^k | s_{1:t}, x_{1:t}, \pi_{1:t-1}^p, \pi_{1:t-1}^q] = 0$. We assume that π_t^k, $k \in \{p, q$ $\}$ is conditionally 1-sub-Gaussian. Formally, this means that

$$\forall \lambda \in \mathbb{R} \mathbb{E}\left[\exp^{\lambda \pi_t^k} | s_{1:t}, x_{1:t}, \pi_{1:t-1}^p, \pi_{1:t-1}^q\right] \leq \exp\left(\frac{\lambda^2}{2}\right). \quad (1)$$

Let $y_*^p(x) \coloneqq \max_{s \in \mathscr{S}} y_s^p(x)$ and $y_*^q(x) \coloneqq \max_{s \in \mathscr{S}} y_s^q(x)$ denote the expected reward of a ECS s in the price and the reliability for context x, respectively. Let $s^*(x)$ denote the optimal ECS for the context x.

Assumption 1. We assume that for all $k \in \{p, q\}$, $s \in \mathscr{S}$ and x, $x' \in \mathscr{X}$, $y_s^k(x)$ satisfy the following condition:

$$\left| y_s^k(x) - y_s^k\left(x'\right) \right| \leq L \| x - x' \|^{\alpha}, \quad (2)$$

where $L > 0$, $0 < \alpha \leq 1$. Assumption 1 means that if the offloading price and reliability of two ESCs are similar, it is expected that the cost of their offloading are similar.

Initially, the MD does not know any reward of ECSs. The MD learns the reward of ECSs over time. In order to evaluate the performance of our method, we define the 2D regret of the ECS as the tuple $(R^p(T), R^q(T))$, where

$$R^k(T) \coloneqq \sum_{t=1}^{T} \quad y_*^k(x_t) - \sum_{t=1}^{T} \quad y_{s_t}^k(x_t), \quad k \in \{p, q\}. \quad (3)$$

When $R^p(T) = O(T^{\beta 1})$ and $R^q(T) = O(T^{\beta 2})$, we consider that the 2D regret is $O(T^{\max(\beta 1, \beta 2)})$.

3.2. Differential Privacy

Definition 2 (differential privacy). An algorithm $\mathcal{G}$ has ε differential privacy if there is only one entry different in all pairs $\mathcal{D}, \mathcal{D}' \in \mathbb{R}^d$, and all set of outcomes $\mathcal{R} \in \text{Range}(\mathcal{G})$.

$$\frac{\mathbf{P}[\mathcal{G}(\mathcal{D}) \in \mathcal{R}]}{\mathbf{P}\left[\mathcal{G}\left(\mathcal{D}'\right) \in \mathcal{R}\right]} \leq \exp(\varepsilon). \tag{4}$$

This definition mentioned above applies only to the identical level of privacy protection used by all suppliers. We now consider that each supplier i in our system specifies its own privacy ε_i.

Definition 3 (local differential privacy). An algorithm $\tilde{\mathcal{G}}$ has M nodes in the system, and we say that the algorithm $\tilde{\mathcal{G}}$ is ε_i locally private for node $i, i = 1, \cdots, M$ if for any $\mathcal{R}_i \in \text{Range}(\tilde{\mathcal{G}}_i)$ it satisfies that

$$\frac{\mathbf{P}\left\{\tilde{\mathcal{G}}_i(\mathcal{D}_1, \cdots, \mathcal{D}_i, \cdots, \mathcal{D}_M) \in \mathcal{R}_i\right\}}{\mathbf{P}\left\{\tilde{\mathcal{G}}_i\left(\mathcal{D}_1, \cdots, \mathcal{D}'_i, \cdots, \mathcal{D}_M\right) \in \mathcal{R}_i\right\}} \leq \exp(\varepsilon_i). \tag{5}$$

And we say that the algorithm $\tilde{\mathcal{G}}$ is $(\varepsilon_1, \cdots, \varepsilon_M)$-differentially private, if $\tilde{\mathcal{G}}$ is ε_i-differentially locally private for all suppliers, where $s = 1, \cdots, M$.

Definition 4 (sensitivity). The sensitivity of the function $y : X \mapsto R^d$ is as follows:

$$\Delta f = \max_{X, X'_{\|X - X'\|_1 = 1}} \left\|f(X) - f\left(X'\right)\right\|_1. \tag{6}$$

Definition 5 (sensitivity). The sensitivity of the function $h : G \mapsto R^d$ is as follows:

$$\Phi_i = \max_{G_i, G'_{i\|G_i - G'_i\|_1 = 1}} \left\|g(G_i) - g\left(G'_i\right)\right\|_1. \tag{7}$$

3.3. The Learning Algorithm. In this section, we detail our proposed EPCO as shown in Algorithm 1 (EPCO(1)), Algorithm 2 (EPCO(2)), and Algorithm 3 (EPCO(3)). Since the computing offloading decision of each ECS for different MDs has stochastic distributions, we decide to let our proposed system learn an ECS's performance by online learning method. According to the sample mean reward of each ECS for the same context vector update, service suppliers learn the performance of each ECS. In order to understand the EPCO, we divide it into three algorithms which are named as EPCO(1), EPCO(2), and EPCO(3), respectively. As shown in Figure 2, the MDs run EPCO(1) to customize their privacy protection level. ECM runs EPCO(2) to interact with the MD and send the best option among all agents to the MD for decisions. Service suppliers run EPCO(3) to select the optimal ECS and interact with the ECM.

```
Input: $\dot{x}$.
Output: $x$.
MD is ready for computing offloading;
$f(\dot{x}) = \dot{x} + Lap(\Delta f / \varepsilon)$;
Set $x = f(\dot{x})$;
Send $x$ to the edge cloud;
```

Algorithm 1: EPCO(1).

First, we analyse the privacy problem of MDs. Since different MDs have different requirements for privacy protection, MDs are allowed to customize the privacy level of each user. In order to protect personal privacy when MDs send computing information, the information is added with a noise which is drawn from the Laplace distribution in EPCO(1). Then, we discuss the privacy issues of the suppliers. When suppliers have selected optimal ECSs for MDs, they send the information to ECM. Since any supplier can access to this information in ECM, the Laplace mechanism is used in EPCO(2) to protect the privacy of service suppliers.

In this paper, the context space $\mathcal{X}$ is divided into m^d identical hypercubes with side length n^{-l}. Let C denote the subspace context space of $\mathcal{X}$. For ECS s and each $c \in \mathcal{C}$, EPCO maintains a counter $N_{s,c}$ recording the number of times that s was selected for the context that belongs to c. When a MD needs to perform a computing offloading, it first sends a message to the ECSs containing information about the computing task. In order to protect the privacy of the MD, it adds Laplace noise to this information in EPCO(1). Upon each context data of a MD arrival, the suppliers first identify to which subspace c the context belongs. Then, each service supplier first calculates the indices for the rewards (line 5 in EPCO(3)), which is given as follows:

$$h_{s,c}^k := \hat{y}_{s,c}^k + \mu_{s,c} + \eta_s^k, \quad k \in \{p, q\}, \tag{8}$$

where $\hat{y}_{s,c}^k = y_{s,c}^k / N_{s,c}$ estimates the sample mean of the reward for the selection of s in subspace c. $\hat{y}_{s,c}^p$ and $\hat{y}_{s,c}^q$ denote the price objective and reliability objective, respectively. $\mu_{s,c} = \sqrt{2S_{n,T}/N_{s,c}}$, where $S_{n,T} = (1 + 2\log(4|\mathcal{S}|n^d T^{3/2}))$ denotes the uncertainty of the reward estimate, which is commonly used to tradeoff exploration and exploitation in online learning [40]. η_s^k is a random noise obeying the Gaussian distribution. Then, the Upper Confidence Bound (UCB) for $y_s^k(x)$ is $h_{s,c}^k + b$ for context x in subspace c, where $b := Ld^{\alpha/2} n^{-\alpha}$ denotes the uncertainty due to context partition. Its main purpose is to inflate the reward of ECSs that are seldomly selected, which is more conducive to exploring more suitable servers than just servers with high estimated reward.

We add Laplace noise η_s^k to the index function to protect the privacy of ECSs. When $\mu_{s_p^*,c} + \eta_s^k \leq \gamma b$, it means that the

```
Input: $x_i$.
Output: $s^*$.
1: Receive $x_i$ from the MD;
2: Broadcast $x_i$ to each service suppliers;
3: Receive optimal ECSs from all service suppliers;
4: Set $s_i^* = \arg\ \max_{\{v \in \mathscr{V}\}}\{J_{v,s}\}$;
5: Send $s_i^*$ to the MD;
6: Observe the decision of the MD and send it to the service suppliers
```

Algorithm 2: EPCO(2).

```
Input: $x_i, \gamma$.
Output: $s^*$.
1: Initialize: $\mathscr{C} \subseteq \mathscr{X}, N_s = 0, \forall s \in \mathcal{S}_i$;
2: Receive $x_i$ from the ECM;
3: for$t = 1, 2, \cdots, T, x_t \in c$do
4:   Compute $h^k_{s,c}, k \in \{p, q\}$ via (4);
5:   $I^k_{s,c} := h^k_{s,c} + \eta^k_s, k \in \{p, q\}$
6:   if$\mu_{s_p^*,c} + \eta^k_s > \gamma b$then
7:     Set $s_p^* = s^{p^*}$;
8:   else
9:     Find the candidate optimal set of ECSs $\mathcal{S}^*$ via (5);
10:     Set $s_p^* = \arg\ \max_{s \in \mathcal{S}^*} g^q_{s,c}$;
11:   end if
12:   Send $s^*$ and $h_{s^*,c}$ to the ECM;
13:   Receive $\mathbf{w}_t = (w^p_t, w^q, t)$;
14:   $\widehat{y}^k_{s_t,c} \longleftarrow (\widehat{y}^k_{s_t,c} N^k_{s_t,c} + w^k_t)/(N^k_{s_t,c} + 1), k \in \{p, q\}$
15:   $N^k_{s_t,c} \longleftarrow N^k_{s_t,c} + 1$
16:   $t \longleftarrow t + 1$;
17: end for
```

Algorithm 3: EPCO(3).

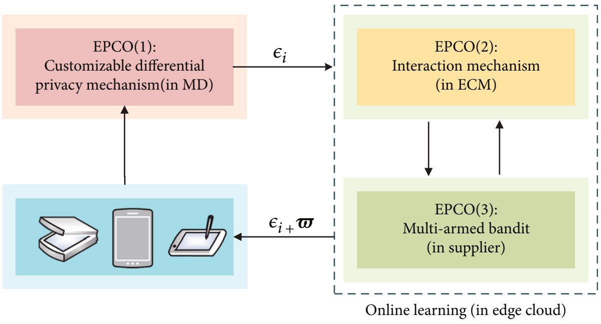


Figure 2: Algorithm mechanism.

confidence of s_p^* is high, and EPCO(3) calculates the candidate set of the optimal ECSs, which is given as follows:

$$\widehat{\mathcal{S}}^* := \left\{ s \in \mathcal{S} : h^p_{s,c} \geq \widehat{y}^p_{s_p^*,c} - \mu_{s_p^*,c} - \eta^k_s - 2b \right\} = \left\{ s \in \mathcal{S} : \widehat{y}^p_{s,c} \geq \widehat{y}^p_{s_p^*,c} - \mu_{s_p^*,c} - \eta^k_s - \mu_{s,c^*} - 2b \right\}. \tag{9}$$

When $\mu_{s_p^*,c} + \eta^k_s \geq \gamma b$, EPCO(3) just set $s_p^* = \arg\ \max_{s \in \mathcal{S}} h^p_{s,c}$ to improve the confidence of s_p^* (lines 6-7 in EPCO(3)). Simultaneously, an optimal ECS $\widehat{s}$ is selected by the exponential mechanism (lines 12-15 in EPCO(2)). We use $h_{s,c}$ to denote the total reward of ECS s that can be compared, which is given as follows:

$$J_{s,c} = \psi \widehat{y}^p_{s,c} + (1 - \psi) \widehat{y}^q_{s,c}, \quad \psi \in [0, 1], \tag{10}$$

where ψ represents a MD's preference and is adjusted according to the actual needs of a MD. For example, if a MD requires strict service payment, the value of ψ is larger. However, if a MD requires strict reliability, the value of ψ is relatively small. We select the ECS with the highest total reward and sent it to the MD (line 12 in EPCO(3)). Finally, according to the computing offloading decision of the MD (line 7 in EPCO(2)), the service suppliers update the sample mean reward and the counter (lines 14-16 in EPCO(3)).

4. Regret Analysis

In this section, we prove that the 2D regret of EPCO is sublinear functions of T. The regret $R^k(T)$ is due to selecting suboptimal ECSs from $\mathcal{S}_i$ by time T.

Let

$$R^k_c(T) := \sum_{t=1}^{T} y^k_*(x_c(t)) - \sum_{t=1}^{T} y^k_{\epsilon^\delta_s}(x_c(t)), \quad k \in \{p, q\} \tag{11}$$

denotes the regret for objective k in round $\mathscr{T}_c$. The best fixed ECS is denoted by $y^k_*(x)$, $y^k_*(x) = \max_{s \in \mathcal{S}} y^k_*(x)$, and $k \in \{p, q\}$. Then, we have the total regret for selecting suboptimal ECSs

$$R^k(T) = \sum_{c \in \mathscr{C}} R^k_c(T). \tag{12}$$

Then, the corresponding expected regret is given as follows:

$$\mathbb{E}\left[R^k(T)\right] = \sum_{c \in \mathscr{C}} \mathbb{E}\left[R^k_c(T)\right]. \tag{13}$$

Let $\underline{h}^k_{s,c}(t) := \widehat{y}^k_{s,c}(t) - \mu_{s,c}(t) - \eta^k_s(t)$ and $\bar{h}^k_{s,c}(t) := \widehat{y}^k_{s,c}(t) + \mu_{s,c}(t) + \eta^k_s(t)$ denote lower and upper bounds for $k \in \{p, q\}$, respectively. Then, $\underline{h}^k_{s,c}(t) - b$ and $\bar{h}^k_{s,c}(t) + b$ are the lower and upper confidence bounds, respectively. Let

$$Q_{s,c}^k \coloneqq \bigcup_{t=1}^{N_c(T)} \left\{ y_s^k(x_s(t)) \notin \left[\underline{h}_{s,c}^k(t) - b, \bar{h}_{s,c}^k(t) + b \right] \right\} \tag{14}$$

denote that the service supplier is not confident about the reward estimate by time T with the context x in subspace c. Then, we partition the regret into following and bound them, respectively.

$$\begin{aligned} \mathbb{E}\left[R_c^{k_1}(T)\right] &= \mathbb{E}\left[R_c^{k_1}(T)|Q\right]P(Q) + \mathbb{E}\left[R_c^{k_1}(T)\left|\tilde{Q}\right.\right]P\left(\tilde{Q}\right) \\ &\le A_{\max}^{k_1} N_c(T)P(Q) + \mathbb{E}\left[R_c^{k_1}(T)\left|\tilde{Q}\right.\right], \end{aligned} \tag{15}$$

where $\tilde{Q}$ denotes the complement of event Q and A^k is the maximum difference between the expected reward in optimal server and other server for objective k. We use s to denote the server selected in EPCO(3) algorithm, $s*$ to denote the optimal server, and $\hat{s}$ to denote the server whose index is highest. Next, we will bound the items in Equation (15). We first bound $P(Q)$.

Lemma 6. *For any $c \in \mathscr{C}$, we have the following:*

$$\mathbf{P}(Q_c) \le \frac{1}{n^d T}. \tag{16}$$

Proof. Let $W_{s,c}^k(t)$ denote the random reward of server s in objective k in round t. We know

$$\begin{aligned} W_s^k, c(t) &= y_s^k(x_c(t)) + \pi_c^k(t) + \eta_c^k, \\ y_s^k, c(t) &= \frac{\sum_{l=1}^{t-1} W_s^k, c(l)\mathbf{I}(s_c(l) = s)}{N_{s,c}(t)}. \end{aligned} \tag{17}$$

□

We define upper and lower bounds of the random reward as follows:

$$\begin{aligned} \widehat{W}_s^k, c(t) &= \widehat{y}_s^k(x_c(t)) + \pi_c^k(t) + \eta_c^k, \\ \check{W}_s^k, c(t) &= \check{y}_s^k(x_c(t)) + \pi_c^k(t) + \eta_c^k. \end{aligned} \tag{18}$$

Let

$$\begin{aligned} \widehat{y}_s^k(x_c(t)) &= \frac{\sum_{l=1}^{t-1} \widehat{W}_s^k, c(l)1(s_c(l) = s)}{N_{s,c}(t)}, \\ \check{y}_s^k(x_c(t)) &= \frac{\sum_{l=1}^{t-1} \check{W}_s^k, c(l)1(s_c(l) = s)}{N_{s,c}(t)}. \end{aligned} \tag{19}$$

Then, we have the following:

$$\begin{aligned} \widehat{L}_{s,c}^k(t) &\coloneqq \widehat{y}_s^k(x_c(t)) - \mu_{s,c}(t) - \eta_c^k, \\ \widehat{U}_{s,c}^k(t) &\coloneqq \widehat{y}_s^k(x_c(t)) + \mu_{s,c}(t) + \eta_c^k, \\ \check{L}_{s,c}^k(t) &\coloneqq \check{y}_s^k(x_c(t)) - \mu_{s,c}(t) - \eta_c^k, \\ \check{U}_{s,c}^k(t) &\coloneqq \check{y}_s^k(x_c(t)) + \mu_{s,c}(t) + \eta_c^k. \end{aligned} \tag{20}$$

Since when $N_{s.c}(t) = 0$, $\mathbf{P}(y_s^k(x_c(t)) \in [\widehat{L}_{s,c}^k(t) - b, \widehat{U}_{s,c}^k(t) + b]) = 0$, so below, we only focus on the case of $N_{s.c}(t) > 0$, which can be expressed as follows:

$$\begin{aligned} &\left\{ y_s^k(y_c(t)) \in \left[L_{s,c}^k(t) - b, U_{s,c}^k(t) + b \right] \right\} \subset \\ &\left\{ y_s^k(y_c(t)) \in \left[\widehat{L}_{s,c}^k(t) - b, \widehat{U}_{s,c}^k(t) + b \right] \right\} \cup \\ &\left\{ y_s^k(y_c(t)) \in \left[\check{L}_{s,c}^k(t) - b, \check{U}_{s,c}^k(t) + b \right] \right\}. \end{aligned} \tag{21}$$

From the Hölder continuity, we have the following derivation:

$$\begin{aligned} &\widehat{y}_s^k(x_c(t) = \sup_{x \in c} y_s^k(y) = y_s^k\left(x'\right), \\ &\widehat{y}_{s,c}^k - y_s^k(x_c(t)) \le L\left\|x' - x_c(t)\right\|^\alpha = L\left(\sqrt{d \cdot \frac{1}{n^2}}\right)^\alpha = L\left(\frac{\sqrt{d}}{n}\right)^\alpha. \end{aligned} \tag{22}$$

Then, integrating the above derivation, we have the following:

$$y_s^k(x_c(t)) \le \widehat{y}_{s,c}^k \le y_s^k(x_c(t)) + L\left(\frac{\sqrt{d}}{n}\right)^\alpha, \tag{23}$$

$$y_s^k(x_c(t)) - L\left(\frac{\sqrt{d}}{n}\right)^\alpha \le \check{y}_s^k(x_c(t)) \le y_s^k(x_c(t)). \tag{24}$$

Using Equation (23) and Equation (24), the question can be expressed as follows:

$$\begin{aligned} &\left\{ y_s^k(x_c(t)) \in \left[\widehat{L}_{s,c}^k(t) - b, \widehat{U}_{s,c}^k(t) + b \right] \right\} \subset \left\{ \widehat{y}_s^k(x_c(t)) \in \left[\widehat{L}_{s,c}^k(t), \widehat{U}_{s,c}^k(t) \right] \right\}, \\ &\left\{ y_s^k(x_c(t)) \in \left[\check{L}_{s,c}^k(t) - b, \check{U}_{s,c}^k(t) + b \right] \right\} \subset \left\{ \check{y}_s^k(x_c(t)) \in \left[\check{L}_{s,c}^k(t), \check{U}_{s,c}^k(t) \right] \right\}. \end{aligned} \tag{25}$$

Thus, plugging Equation (23) and Equation (24) into Equation (21), we obtain the following:

$$\begin{aligned} &\left\{ y_s^k(x_c(t)) \in \left[L_{s,c}^k(t) - b, U_{s,c}^k(t) + b \right] \right\} \subset \\ &\left\{ \widehat{y}_s^k \in \left[\widehat{L}_{s,c}^k(t), \widehat{U}_{s,c}^k(t) \right] \right\} \cup \left\{ \check{y}_s^k \in \left[\check{L}_{s,c}^k(t), \check{U}_{s,c}^k(t) \right] \right\} p. \end{aligned} \tag{26}$$

Using Equation (26), we have the following:

$$\mathbf{P}(\tilde{U}) \leq \mathbf{P}\left(\bigcup_{t=1}^{N_c(T)} \left\{\hat{y}_s^k \in \left[\hat{L}_{s,c}^k(t), \hat{U}_{s,c}^k(t)\right]\right\}\right) + \mathbf{P}\left(\bigcup_{t=1}^{N_c(T)} \left\{\check{y}_s^k \in \left[\check{L}_{s,c}^k(t), \check{U}_{s,c}^k(t)\right]\right\}\right). \tag{27}$$

Using the concentration inequality, the right side of the above inequality is bounded as follows:

$$\mathbf{P}(\tilde{U}) \leq \frac{1}{2|\mathcal{S}|n^d T}. \tag{28}$$

Using the union bound, we have the following:

$$\begin{aligned} \mathbf{P}(\tilde{U}) &\leq \frac{1}{2n^d T}, \\ \mathbf{P}(\tilde{U}) &\leq \frac{1}{pn^d T}. \end{aligned} \tag{29}$$

Using the result of Lemma 10, $\mathbf{P}(U)$ and $\mathbf{P}(\tilde{U})$ can be bounded as follows:

$$\begin{aligned} \mathbf{P}(U) = \mathbf{P}\left(\cup_{k\in\{p,q\}} U_c^k\right) &= m^d \cdot \mathbf{P}(U_c) \leq \frac{1}{T}, \\ \mathbf{P}(\tilde{U}) &\geq 1 - \frac{1}{T}. \end{aligned} \tag{30}$$

Lemma 7. *Under Assumption 1, $y_{s^*(t)}^p(x_c(t))$ and $y_{s(t)}^p(x_c(t))$ are generated by EPCO(3) algorithm. On event $\tilde{U}$, we have the following:*

$$y_{s^*(t)}^p(x_c(t)) - y_{s(t)}^p(x_c(t)) \leq U_{s,c}^p(t) - L_{s,c}^p(t) + 2(\gamma+2)b. \tag{31}$$

Proof. There are two cases here. When $\mu_{\widehat{s}_p^*,c} + \eta_s^k \leq \gamma b$, we have the following:

$$U_{s,c}^p(t) \geq L_{\widehat{s},c}^p - 2b \geq U_{\widehat{s},c}^p(t) - \mu_{\widehat{s}_p^*,c} - 2\eta_s^k - 2b \geq U_{\widehat{s},c}^p(t) - 2(\gamma+1)b. \tag{32}$$

When $\mu_{\widehat{s}_p^*,c} + \eta_s^k \geq \gamma b$, we have the following:

$$U_{s,c}^p(t) = U_{\widehat{s}^*,c}^p(t) \geq U_{\widehat{s}^*,c}^p(t) - 2(\gamma+1)b. \tag{33}$$

According to the above two cases, we obtain the following:

$$U_{s,c}^p(t) \geq U_{\widehat{s}^*,c}^p(t) - 2(\gamma+1)b \geq U_{s^*,c}^p(t) - 2(\gamma+1)b, \tag{34}$$

$$U_{\widehat{s}^*,c}^p(t) \geq U_{s^*,c}^p(t). \tag{35}$$

On event $\tilde{U}$, we have the following:

$$y_{s*,c}^p(x_c(t)) \leq U_{s^*,c}^p(t) + b \leq U_{\widehat{s}^*,c}^p(t) + 2(\gamma+1)b + b, \tag{36}$$

$$y_{s,c}^p(x_c(t)) \leq U_{s,c}^p(t) - b. \tag{37}$$

Combined with Equations (34)–(37), we obtain the following:

$$y_{s^*,c}^p(x_c(t)) - y_{s,c}^p(x_c(t)) \leq U_{s,c}^p(t) - L_{s,c}^p(t) + 2(\gamma+2)b. \tag{38}$$

□

Lemma 8. *Under Assumption 1, $y_{s^*(t)}^q(x_c(t))$ and $y_{s(t)}^q(x_c(t))$ are generated by EPCO(3) algorithm. On event $\tilde{U}$, we have the following:*

$$y_{s^*(t)}^q(x_c(t)) - y_{s(t)}^q(x_c(t)) \leq U_{s,c}^q(t) - L_{s,c}^q(t) + 2b. \tag{39}$$

Proof. We know that when $\mu_{\widehat{s}_p^*,c} + \eta_s^k \leq \gamma b$ holds, all servers $\widehat{\mathcal{S}}^*$ are in interval $[L_{\widehat{s}^*,c}^p(t) - 2b, U_{\widehat{s}^*,c}^p(t)]$. Then, we show that $U_{s^*,c}^p(t)$ also satisfies this condition. On event $\tilde{U}$, we have the following:

$$y_{s^*,c}^p(x_c(t)) \in \left[L_{s^*,c}^p(t) - b, U_{s^*,c}^p(t) + b\right], \tag{40}$$

$$y_{\widehat{s}^*,c}^p(x_c(t)) \in \left[L_{\widehat{s}^*,c}^p(t) - b, U_{\widehat{s}^*,c}^p(t) + b\right], \tag{41}$$

$$y_{s^*(t)}^q(x_c(t)) \geq y_{\widehat{s}^*,c}^p(x_c(t)). \tag{42}$$

By Equations (40)–(42), we obtain the following:

$$U_{s^*,c}^p \geq y_{s^*,c}^p(x_c(t)) - b \geq y_{\widehat{s}^*,c}^p(x_c(t)) - b \geq L_{\widehat{s}^*,c}^p(t) - 2b. \tag{43}$$

□

Since the selected server s is in $[L_{\widehat{s}^*,c}^p(t) - 2b, U_{\widehat{s}^*,c}^p]$, we have $U_{s,c}^q \geq U_{s^*,c}^q$. Using this result, we obtain the following:

$$y_{s,c}^q \geq U_{s,c}^q - b, \tag{44}$$

$$y_{s^*,c}^q \leq U_{s^*,c}^q + b \leq U_{s,c}^q + b. \tag{45}$$

From Equation (44) and Equation (45), we have the following:

$$y_{s^*,c}^q - y_{s,c}^q \leq U_{s,c}^q - L_{s,c}^q + 2b. \tag{46}$$

Lemma 9. *Under Assumption 1, $\mu_{\widehat{s}_p^*,c}$ are generated by EPCO(3) algorithm, and the upper limit of the number of rounds for $\mu_{\widehat{s}_p^*,c} + \eta_s^k > \gamma b$ is as follows:*

$$|\mathcal{S}|\left(\frac{2S_{n,T}}{\left(\gamma b - \eta_s^k\right)^2} + 1\right). \tag{47}$$

Proof. Since

$$\mu_{\widehat{s}_p^*,c} = \sqrt{\frac{2S_{n,T}}{N_{\widehat{s}^*,c}}},$$
$$N_{\widehat{s}^*,c} = \frac{2S_{n,T}}{\mu_{\widehat{s}_p^*,c}} < \frac{2S_{n,T}}{\left(\gamma b - \eta_s^k\right)^2}. \tag{48}$$

□

And each such event increases the value of $N_{\widehat{s}^*,c}$ by one. The number of rounds for $\mu_{\widehat{s}_p^*,c} + \eta_s^k > \gamma b$ is bounded by 2 $S_{n,T}/(\gamma b - \eta_s^k)^2 + 1$. Summing all the servers together obtains the final result.

Lemma 10. *Under Assumption 1, $R_c^p(t)$ is generated by EPCO(3) algorithm. On event $\tilde{U}$, we have the following:*

$$R_c^p(t) \le |\mathcal{S}|A_{\max}^p + 2B_{n,T}\sqrt{|\mathcal{S}|N_c(t)} + 2(\gamma+2)zN_c(t), \tag{49}$$

where $B_{n,T} \coloneqq 2\sqrt{2S_{n,T}}$.

Lemma 11. *Under Assumption 1, $R_c^q(t)$ is generated by EPCO(3) algorithm. On event $\tilde{U}$, we have for all $c \in \mathscr{C}$*

$$R_c^q(t) \le |\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zN_c(t) + 2B_{n,T}\sqrt{|\mathcal{S}|N_c}. \tag{50}$$

More detailed proof of Lemma 10 and Lemma 11 is presented in [41] (see Lemma 10 and Lemma 11 in [41]).

Theorem 12. *Under Assumption 1, $R^k(t)$ is generated by EPCO(3) algorithm, and we have for any $k \in p, q$ as follows:*

$$\mathbf{P}\left(R^k(t) < \xi_k(t)\right) \ge 1-, \quad \forall t \in \{1, \cdots, T\}, \tag{51}$$

where

$$\xi_p(t) = n^d|\mathcal{S}|A_{\max}^p + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t} + 2(\gamma+2)zt,$$
$$\xi_q(t) = n^d|\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zt + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t}. \tag{52}$$

Proof. Combining Equation (12), Lemma 10, and Lemma 11, we have the following:

$$R^p(t) \le n^d|\mathcal{S}|A_{\max}^p + 2B_{n,T}\sum_{c\in\mathscr{C}}\sqrt{|\mathcal{S}|N_c(t)} + 2(\gamma+2)zN_c(t),$$
$$\le n^d|\mathcal{S}|A_{\max}^p + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t} + 2(\gamma+2)zt, R^q(t)$$
$$\le n^d|\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zN_c(t) + 2B_{n,T}\sum_{c\in\mathscr{C}} \quad \sqrt{|\mathcal{S}|N_c}$$
$$\le n^d|\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zt + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t}. \tag{53}$$

□

Theorem 13. *Under Assumption 1, $R^k(t)$ is generated by EPCO(3) algorithm, and m and γ satisfy $n = \lceil T^{1/(3\rho+d)}\rceil$ and $\gamma > 0$, and we have the following:*

$$\mathbb{E}[R^p(T)] \le A_{\max}^p + 2^d|\mathcal{S}|A_{\max}^q T^{d/3\rho+d} + 2(\gamma+2)Ld^{\rho/2}T^{2\rho+d/3\rho+d}$$
$$+ 2^{d/2+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d}, \mathbb{E}[R^q(T)]$$
$$\le 2^{d/2+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d}$$
$$+ A_{\max}^q + \left(2Ld^{\rho/2} + \frac{A_{\max}^q|\mathcal{S}|2^{1+2\rho+d}B_{n,T}}{\gamma^2L^2d^\rho}\right)T^{2\rho+d/3\rho+d}$$
$$+ 2^dA_{\max}^q|\mathcal{S}|T^{d/3\rho+d}. \tag{54}$$

Proof. According to the Theorem 12 and Equation (15), $\mathbb{E}[R^k(T)]$ is bounded as follows:

$$\mathbb{E}\left[R^k(T)\right] \le \mathbb{E}\left[R^k(T)\middle|\tilde{U}\right] + \sum_{c\in\mathscr{C}} \quad A_{\max}^k N_c(t)\mathbf{P}(U) \le \mathbb{E}\left[R^k(T)\middle|\tilde{U}\right]$$
$$+ \frac{\sum_{c\in\mathscr{C}} \quad A_{\max}^k N_c(t)}{T = \mathbb{E}\left[R^k(T)\middle|\tilde{U}\right] + A_{\max}^k} \tag{55}$$

Then, we obtain the following:

$$\mathbb{E}[R^p(T)] \le \xi_p(t) + A_{\max}^p,$$
$$\mathbb{E}[R^q(T)] \le \xi_q(t) + A_{\max}^q. \tag{56}$$

When $n = \lceil T^{1/(3\rho+d)}\rceil$, we have the following:

$$\mathbb{E}[R^p(T)] \le A_{\max}^p + 2^d|\mathcal{S}|A_{\max}^q T^{d/3\rho+d} + 2(\gamma+2)Ld^{\rho/2}T^{2\rho+d/3\rho+d}$$
$$+ 2^{d/2+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d}, \mathbb{E}[R^q(T)]$$
$$\le 2^{(d/2)+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d} + A_{\max}^q$$
$$+ \left(2Ld^{\rho/2} + \frac{A_{\max}^q|\mathcal{S}|2^{1+2\rho+d}B_{n,T}}{\gamma^2L^2d^\rho}\right)T^{2\rho+d/3\rho+d}$$
$$+ 2^dA_{\max}^q|\mathcal{S}|T^{d/3\rho+d}. \tag{57}$$

The result shows that not only the regret of EPCO is sublinear, which is $O(T^{(2\rho+d)/(3\rho+d)})$, but also added privacy differential mechanism does not affect its convergence. □

5. Differentially Private

The privacy protection mechanism applied in this paper uses *differential privacy* mechanism, which is originally introduced by Dwork et al. [21].

Theorem 14. *EPCO(1) preserves $(\varepsilon_i, 0)$-differential privacy for MD i where*

$$\varepsilon_i = \Phi_i \sum_{t=1}^{T} \frac{1}{\rho_i^t}, \tag{58}$$

and each MD i of the noise η_i^t is independently selected according to the Laplace distribution, where the density function is $p(\eta_i^t) = (1/2\rho_i^t) \exp(-\|\eta_i^t\|/\rho_i^t)$ for $i = 1, \cdots, M$ and $t = 1, \cdots, T$.

Proof. We first show that $\tilde{\mathscr{G}}_i$ is locally $(\varepsilon_i, 0)$-differential privacy for MD i. Now we start by studying the quantity of interest

$$\frac{\mathbf{P}\left\{\tilde{\mathscr{G}}_i^T(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M) \in \mathscr{W}\right\}}{\mathbf{P}\left\{\tilde{\mathscr{G}}_i^T\left(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M\right) \in \mathscr{W}\right\}} \tag{59}$$

of MD i. We use a random variable W_i^t, for $t = 1, \cdots, T$, to denote the output of $\tilde{\mathscr{G}}_i^T$ with input $(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M)$ and W'^t_i to denote the output of $\tilde{\mathscr{G}}_i^T$ with input $(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M)$. □

According to the definition above, we can rewrite the issue as follows:

$$\frac{\mathbf{P}\left\{W_i^1 = w_i^1, \cdots, W_i^T = w_i^T\right\}}{\mathbf{P}\left\{W'^1_i = w_i^1, \cdots, W'^T_i = w_i^T\right\}}. \tag{60}$$

We use W^t to denote the tuple $(W_1^t, \cdots, W_M^t)$. Then, we have the following:

$$\begin{aligned}\frac{\mathbf{P}\left\{W_i^1 = w_i^1, \cdots, W_i^T = w_i^T\right\}}{\mathbf{P}\left\{W'^1_i = w_i^1, \cdots, W'^T_i = w_i^T\right\}} &= \frac{\mathbf{P}\left\{W^1 = w^1, \cdots, W^T = w^T\right\}}{\mathbf{P}\left\{W'^1 = w^1, \cdots, W'^T = w^T\right\}} \\ &= \prod_{t=1}^{T} \frac{\mathbf{P}\left\{W^t = w^t \mid W^\tau = w^\tau, \tau < t\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| W'^\tau = w^\tau, \tau < t\right\}}.\end{aligned} \tag{61}$$

We form this process into a Markov chain where the random vector a^t denotes the Lagrangian $(v_1^t, \cdots, v_M^t)$, which is presented in [39] (see Theorem 3.1 in [39]). Then, we have the following relationship:

$$\begin{aligned}\frac{\mathbf{P}\left\{W^t = w^t \mid W^\tau = w^\tau, \tau < t\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| W'^\tau = w^\tau, \tau < t\right\}} &= \frac{\mathbf{P}\left\{W^t = w^t \mid a^{t-1} = v^{t-1}\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| a'^{t-1} = v'^{t-1}\right\}} \\ &= \frac{\mathbf{P}\left\{W^t = w^t \mid a^{t-1} = v^{t-1}\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| a'^{t-1} = v^{t-1}\right\}} \le \exp\left(\frac{\left\|y\left(v_i^{t-1}\right) - y\left(v'^{t-1}_i\right)\right\|}{\rho_i^t}\right) \\ &\le \exp\left(\frac{\Phi_i}{\rho_i^t}\right).\end{aligned} \tag{62}$$

With Φ_i in Definition 5, we obtain the following:

$$\begin{aligned}&\frac{\mathbf{P}\left\{\mathscr{G}_i^T(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M) \in \mathscr{W}\right\}}{\mathbf{P}\left\{\mathscr{G}_i^T\left(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M\right) \in \mathscr{W}\right\}} \\ &= \prod_{t=1}^{T} \frac{\mathbf{P}\left\{W^t = w^t \mid W^\tau = w^\tau, \tau < t\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| W'^\tau = w^\tau, \tau < t\right\}} \le \prod_{t=1}^{T} \exp\left(\frac{\Phi_i}{\rho_i^t}\right) \\ &= \exp\left(\sum_{t=1}^{T}\left(\frac{\Phi_i}{\rho_i^t}\right)\right).\end{aligned} \tag{63}$$

This result proves the privacy guarantee in Equation (58).

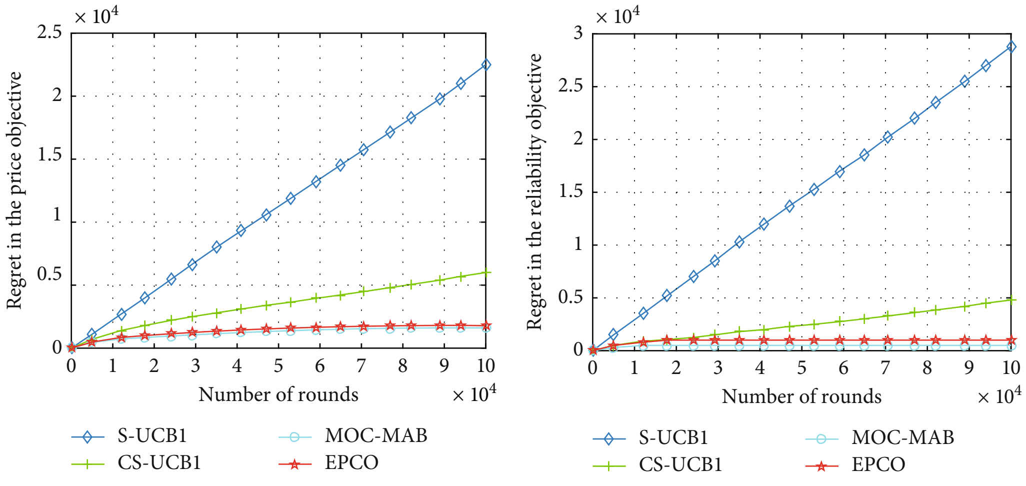
Thus, Theorem 14 proves that our proposed EPCO(1) can guarantee the MD's privacy and different MDs have different privacy protection levels.

Theorem 15. *EPCO(1) preserves $(\varepsilon, 0)$-differential privacy for the contextual information of ECSs.*

Proof. Let $D = (h_1, \cdots, h_T)$ denote true information of a supplier and $\mathscr{D}'$ denote a dataset which differs from $\mathscr{D}$ in only one data. We define the reward that adds noise as I_t. Then, for different suppliers s_1 and s_2, we have the following:

$$\begin{aligned}\frac{\mathbf{P}[\mathscr{M}(s_1, t) = I_t]}{\mathbf{P}[\mathscr{M}(s_2, t) = I_t]} &= \frac{\exp\left(-\left(\varepsilon'\left|h_{s_1,t} - I_t\right|/\Delta f\right)\right)}{\exp\left(-\left(\varepsilon'\left|h_{s_2,t} - I_t\right|/\Delta f\right)\right)} \\ &= \exp\left(\frac{\varepsilon'}{\Delta f}\left(\left|h_{s_2,t} - I_t\right| - \left|h_{s_1,t} - I_t\right|\right)\right) \\ &\le \exp\left(\frac{\varepsilon'}{\Delta f}\left|h_{s_1,t} - h_{s_2,t}\right|\right) \\ &= \exp\left(\frac{\varepsilon'}{\Delta f}\left\|h_{s_1,t} - h_{s_2,t}\right\|_1\right) \le \exp\left(\varepsilon'\right).\end{aligned} \tag{64}$$

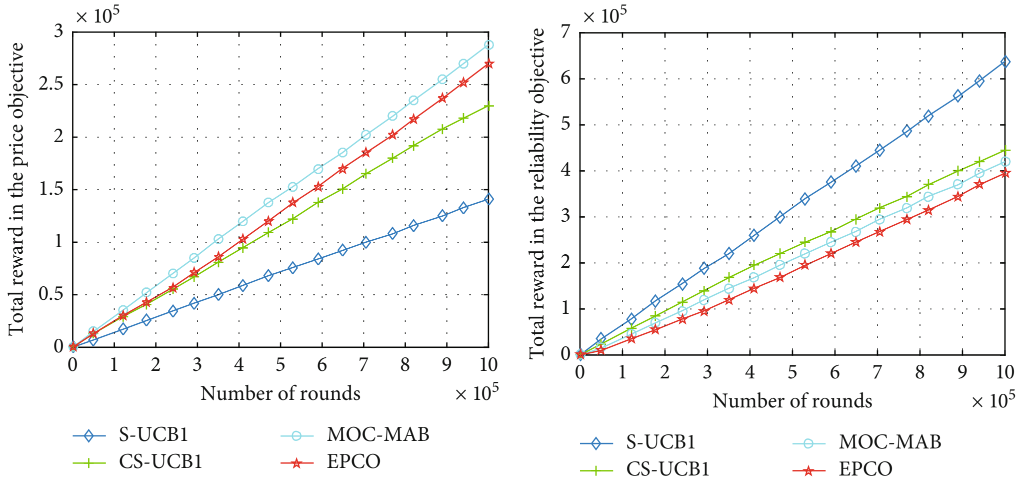
Employing Theorem 3.6 in [42] (see page 32 in [42]), Theorem 15 is proved. □



(a) Regret of price objective

(b) Regret of reliability objective

Figure 3: Regret of multiobjective.



(a) Reward of price objective

(b) Reward of reliability objective

Figure 4: Reward of multiobjective.

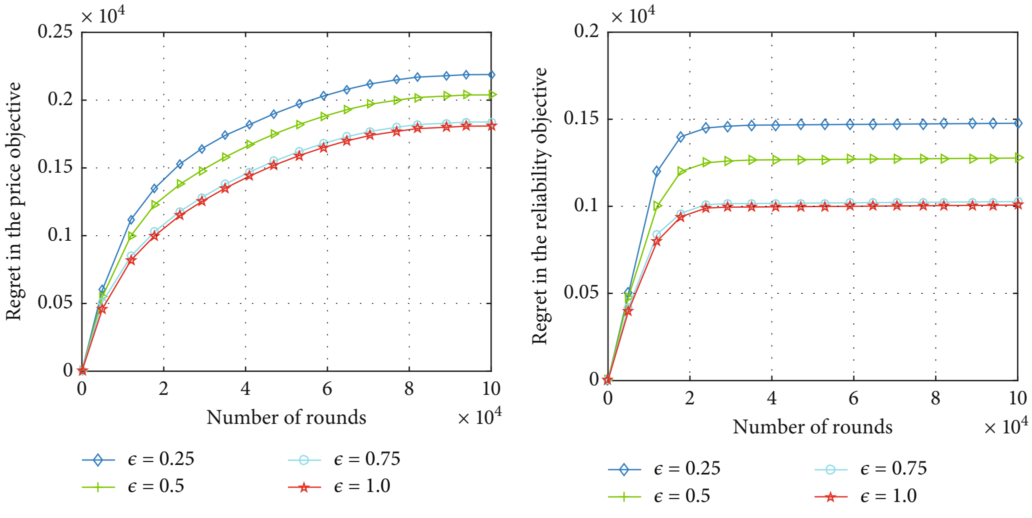
Therefore, Theorem 15 proves that service supplier fails to extract information about ECSs in ECM by the rewards. In summary, Theorem 14 and Theorem 15 prove that EPCO can preserve both privacy of MDs and service suppliers synchronously. In addition, EPCO also supports different MDs to customize their privacy protection levels.

6. Simulation Results

In order to verify the efficiency and privacy of our proposed algorithm, we conducted the following simulation experiments with real-world datasets [43]. We compare EPCO with P-UCB1, S-UCB1 [44], and MOC-MAB [41] algorithms. We use Python 3.6 version to implement these algorithms. We run these algorithms on an Acer computer with Intel(R) Core(TM) i5-4460 @ 3.2 GHz and 8 GB RAM. The operating system is Windows 10 Professional. We set L, α, and γ to 1. We give the sets $\mathscr{P} = \{0.2,0.4,0.6\}$ and $\mathscr{R} = \{0.1,0.25,0.5,1\}$. We let the time horizon $T = 10^5$ and $T = 10^6$ to satisfy different experiments. We take $\mathscr{X} = [0, 1]^2$, and the context is chosen randomly from $\mathscr{X}$ at each round. We use 6 arms in each algorithm and every algorithm runs 20 times. We take the average result of the simulation experiment.

6.1. Analysis of Regret. Figure 3 shows the change of the regret of the algorithm in price and reliability objectives over time. The simulation results show that the regret of EPCO is at a lower position to both price and reliability objectives. But its regret is not the lowest, which is slightly higher than the MOC-MAB algorithm. Because we added a privacy protection mechanism to EPCO, it affected the convergence of the algorithm, but this effect was almost negligible.

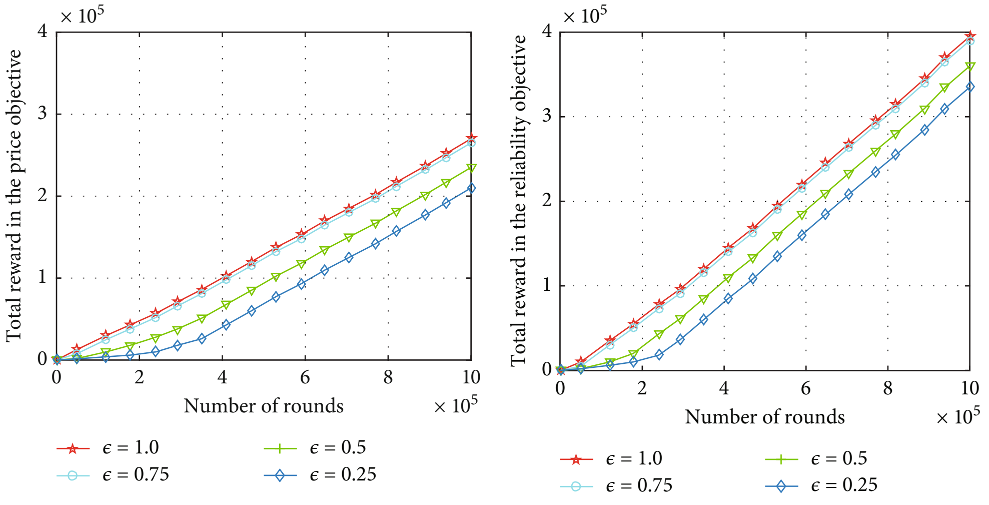
6.2. Analysis of Rewards. We compare the rewards of these algorithms in two objectives, and the results are shown in



(a) Impact on price objective

(b) Impact on reliability objective

Figure 5: The relationship between regret and privacy.



(a) Impact on price objective

(b) Impact on reliability objective

Figure 6: The relationship between reward and privacy.

Figure 4. Compared with other multiobjective optimization algorithms, EPCO performs relatively well. Figure 4(a) shows that the EPCO algorithm performs well in price objective, outperforming other algorithms, but slightly inferior to the MOC-MAB algorithm. Figure 4(b) shows that the EPCO algorithm also performs well in reliability objective. This is because although our algorithm has high efficiency, due to the addition of a privacy protection mechanism, a small part of the error is caused, which affects the reward of different objectives.

6.3. Analysis of Privacy Protection. The effect of different privacy protection levels on the performance of the EPCO algorithm is shown in Figures 5 and 6. We set the privacy parameter ε to 0.25, 0.5, 0.75, and 1, respectively, representing different levels of privacy protection. Figure 5 shows the relationship between different privacy levels and regret. Obviously, when the value of ε is larger, the regret of the EPCO algorithm is smaller. This is because as the value of ε increases, the availability of data increases, resulting in a smaller regret. Similarly, in Figure 6, as the value of ε increases, the availability of data also increases, resulting in a larger reward for the EPCO algorithm.

6.4. Discussion. The simulation results show that EPCO's regret is at a low level in P-UCB1, S-UCB1, and MOC-MAB, because P-UCB1 and S-UCB1 are not suitable for multiobjective optimization. The regret of MOC-MAB is lower than that of EPCO. This is due to the extra work done by EPCO to protect the privacy of users. With ε taking 0.25, 0.5, 0.75, and 1.0, respectively, the regret and reward EPCO also produced different consequences. We noticed that when the value of the privacy parameter ε is larger, the regret of EPCO is smaller, the reward and privacy leakage is larger.

This is because a larger ε has little effect on the calculation performance of the algorithm. Although there is a difference in privacy leakage values from the data point of view, this difference is almost impossible to detect in real scenarios; that is to say, regardless of whether the privacy parameter ε is set to 0.1 or 0.5, this means that user privacy is almost impossible to be leaked.

6.5. Lessons Learned. The purpose of our simulation experiment is to verify the performance of EPCO from three perspectives, namely, regret, reward, and privacy. We used the Python programming language to implement the EPCO algorithm and used real-world datasets for operations. By comparing with three multiobjective optimization algorithms from the perspectives of regret and reward, namely, P-UCB1, S-UCB1, and MOC-MAB, it is concluded that EPCO does a good job in these two aspects. Meanwhile, it is concluded that EPCO can also effectively protect the privacy of users in the system through the method of privacy leakage. This is because differential privacy mechanism integrates well with online learning algorithms. No matter whether ε is set to 0.1 or 1.0, there is no trending change in the performance of EPCO. That is to say, theoretically, the value of b can be adjusted as large as possible within a certain range, which can not only ensure the privacy of users but also minimize the impact of privacy protection mechanisms on the calculation performance of the algorithm.

7. Conclusion

We have proposed a privacy-protected algorithm and EPCO algorithm, for two types of users of computing offloading. One is for MDs whose privacy is protected by a customizable differential privacy mechanism. The other is for ECS suppliers whose privacy is protected by using a common differential privacy mechanism. In addition, we proved no significant impact on the performance of the EPCO algorithm with our privacy protection mechanisms. Simultaneously, an online learning algorithm is introduced to improve the computing offloading efficiency and a detailed theoretical proof of the method is given. The simulation results verify the effectiveness of the EPCO algorithm.

Data Availability

The data that support the findings of this study are found in "More Google cluster data" [43].

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. (62002102) and No. (62176113), in part by the Leading Talents of Science and Technology in the Central Plain of China under Grant No. (224200510004), and in part by the Luoyang Major Scientific and Technological Innovation Projects under Grant No. (2101017A).

References

[1] M. Li, Q. Wu, J. Zhu, R. Zheng, and M. Zhang, "A computing offloading game for mobile devices and edge cloud servers," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2179316, 10 pages, 2018.

[2] W. Zhang, D. Yang, W. Wu et al., "Optimizing federated learning in distributed industrial IoT: a multiagent approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, 2021.

[3] W. Wu, N. Chen, C. Zhou et al., "Dynamic ran slicing for service-oriented vehicular networks via constrained learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2076–2089, 2021.

[4] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: a survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[5] R. Zheng, J. Zhu, M. Zhang, R. Liu, Q. Wu, and Y. Li, "A novel resource deployment approach to mobile microlearning: from energy-saving perspective," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 7430860, 15 pages, 2019.

[6] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[7] Q. Wu, M. Zhang, R. Zheng, Y. Lou, and W. Wei, "A QoS-satisfied prediction model for cloud-service composition based on a hidden Markov model," *Mathematical Problems in Engineering*, vol. 2013, Article ID 387083, 7 pages, 2013.

[8] F. Song, M. Zhu, Y. Zhou, I. You, and H. Zhang, "Smart collaborative tracking for ubiquitous power IoT in edge-cloud interplay domain," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6046–6055, 2020.

[9] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.

[10] W. Shangguang, Z. Yali, X. Jinliang, Y. Jie, and H. Ching-Hsien, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[11] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: a computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[12] F. Song, Z. Ai, Y. Zhou, I. You, K.-K. R. Choo, and H. Zhang, "Smart collaborative automation for receive buffer control in multipath industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1385–1394, 2020.

[13] F. Song, Z. Ai, H. Zhang, I. You, and S. Li, "Smart collaborative balancing for dependable network components in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6916–6924, 2021.

[14] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017.

[15] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in

blockchain-empowered mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8050–8062, 2019.

[16] M. Zhang, Y. Zhou, Q. Ge, R. Zheng, and Q. Wu, "Decentralized randomized block-coordinate Frank–Wolfe algorithms for submodular maximization over networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2021.

[17] M. Zhang, B. Hao, Q. Ge, J. Zhu, R. Zheng, and Q. Wu, "Distributed adaptive subgradient algorithms for online learning over time-varying networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4518–4529, 2022.

[18] J. Zhu, Q. Wu, M. Zhang, R. Zheng, and K. Li, "Projection-free decentralized online learning for submodular maximization over time-varying networks," *Journal of Machine Learning Research*, vol. 22, pp. 1–42, 2021.

[19] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-aware offloading in mobile-edge computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, 2017.

[20] M. Min, X. Wan, L. Xiao et al., "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, 2019.

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds., pp. 265–284, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[22] Y. Zhang, Y. Mao, and S. Zhong, "Joint differentially private Gale-Shapley mechanisms for location privacy protection in mobile traffic offloading systems," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2738–2749, 2016.

[23] M. Ul Hassan, M. H. Rehmani, R. Kotagiri, J. Zhang, and J. Chen, "Differential privacy for renewable energy resources based smart metering," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 69–80, 2019.

[24] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[25] Y. Kim, H. Lee, and S. Chong, "Mobile computation offloading for application throughput fairness and energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 3–19, 2019.

[26] M. Zhang, M. Yang, Q. Wu, R. Zheng, and J. Zhu, "Smart perception and autonomic optimization: a novel bio-inspired hybrid routing protocol for MANETs," *Future Generation Computer Systems*, vol. 81, pp. 505–513, 2018.

[27] R. Zheng, J. Chen, M. Zhang, Q. Wu, J. Zhu, and H. Wang, "A collaborative analysis method of user abnormal behavior based on reputation voting in cloud environment," *Future Generation Computer Systems*, vol. 83, pp. 60–74, 2018.

[28] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4242–4251, 2019.

[29] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[30] J. Barrameda and N. Samaan, "A novel statistical cost model and an algorithm for efficient application offloading to clouds," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 598–611, 2018.

[31] S. Shahrampour, M. Noshad, J. Ding, and V. Tarokh, "Online learning for multimodal data fusion with application to object recognition," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, pp. 1259–1263, 2018.

[32] P. Sakulkar and B. Krishnamachari, "Online learning schemes for power allocation in energy harvesting communications," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4610–4628, 2018.

[33] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: a game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–764, 2018.

[34] C. Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of Lecture Notes in Computer Science, , pp. 1–12, Springer, 2006.

[35] F. Song, Y.-T. Zhou, Y. Wang, T. Zhao, I. You, and H.-K. Zhang, "Smart collaborative distribution for privacy enhancement in moving target defense," *Information Sciences*, vol. 479, pp. 593–606, 2019.

[36] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 4–17, 2018.

[37] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

[38] C. Piao, Y. Shi, J. Yan, C. Zhang, and L. Liu, "Privacy-preserving governmental data publishing: a fog-computing-based differential privacy approach," *Future Generation Computer Systems*, vol. 90, pp. 158–174, 2019.

[39] R. Dobbe, Y. Pu, J. Zhu, K. Ramchandran, and C. Tomlin, "Customized local differential privacy for multi-agent distributed optimization," 2018, https://arxiv.org/abs/1806.06035.

[40] C. Tekin, J. Yoon, and M. van der Schaar, "Adaptive ensemble learning with confidence bounds," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 888–903, 2017.

[41] C. Tekin and E. Turgay, "Multi-objective contextual multi-armed bandit with a dominant objective," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3799–3813, 2018.

[42] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[43] J. Wilkes, "More Google cluster data," 2011, http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.