EdgeDR: An Online Mechanism Design for Demand Response in Edge Clouds

Shutong Chen, Lei Jiao, *Member, IEEE*, Fangming Liu*, *Senior Member, IEEE*, and Lin Wang, *Member, IEEE*

Abstract—The computing frontier is moving from centralized mega datacenters towards distributed cloudlets at the network edge. We argue that cloudlets are well-suited for handling power demand response to help the grid maintain stability due to more flexible workload management attributed to their distributed nature. However, they also require computing demand response to avoid overload and maintain reliability. To this end, we propose a novel online market mechanism, EdgeDR, to achieve cost efficiency in edge demand response programs. At a high level, we observe that the cloudlet operator can dynamically switch on/off entire cloudlets to compensate for the energy reduction required by the power grid or provide enough computing resources to the edge service. We formulate a long-term social cost minimization problem and decompose it into a series of one-round procurement auctions. In each auction instance, we propose to let the cloudlet tenants bid with cost functions of their two-dimension service quality degradation tolerance, and let the cloudlet operator choose the service quality, manage the workload, and schedule the cloudlet activation status. In addition, we present a dynamic payment mechanism for the operator to balance the tradeoff between short-term profit and long-term benefit in more practical scenarios. Via rigorous analysis, we exhibit that our bidding policy is individually rational and truthful; our workload management algorithm has near-optimal performance in each auction; and our overall online algorithm achieves a provable competitive ratio. We further confirm the performance of our mechanism through extensive trace-driven simulations.

Index Terms—Edge demand response, energy saving, cloudlet control, online mechanism

1 Introduction

Cloudlets are the key infrastructures to realizing the promise of edge computing [1], [2]. Often in the forms of small data centers, machine rooms, and server clusters, cloudlets can provide low latency, service redundancy, and data privacy to end users from the Internet edge such as metro stations, enterprise premises, cellular towers, and WiFi neighborhoods. Cloudlets massively exist in metropolises, contributing to more flexibility in their workload management [3], [4]. These characteristics make cloudlets well-suited for handling power demand response. For example, the cloudlet can participate in emergency demand response (EDR) to prevent power blackouts at peak hours and help the power grid maintain stability and reliability [5], [6], [7]. On the other hand, since a cloudlet typically has limited computing resources, it is necessary to protect itself from resource depletion at computation-peak hours and this can be done by performing demand response on computing resources. We call this problem the edge demand response problem.

Similar to an Infrastructure-as-a-Service (IaaS) cloud, not all resources of an IaaS cloudlet are controlled by a single party—the cloudlet facilities/hardware (e.g., servers) and the cloudlet software (e.g., virtual machines) are often operated by the cloudlet operator and the tenants (i.e., service providers who provide services to end users), respectively. This creates the so-called "split incentives" hurdle [5], [8]. For the edge demand response to work, both the cloudlet operator and the tenants would need to participate. However, tenants often have no motivation to join, because their concerns are not about reducing energy or saving computing resources at peak hours, but about obtaining enough resources from cloudlets via paying usage fees to the cloudlet operator. This hurdle also exists in "colocation" data centers, where tenants own and run their servers, together with virtual machines and services.

Many efforts have been devoted to designing mechanisms to address split incentives for power demand response; however, existing solutions have limitations and are unsuitable for cloudlets. First, they lack flexibility in procurements and are restrictive in adapting to the changing market conditions. Auction-based mechanisms procure tenants' bids of fixed commitments [5] while rewards-based mechanisms set reward rates and accept tenants' offers obliviously [9], [10]. Second, they mostly assume tenants to reduce their workload to reduce power consumption [8], [11] rather than manage tenants' workload or schedule the resource allocation themselves. The only workload-aware mechanism known to us explores temporal flexibility for batch jobs [6], [7], which does not match the type of workload of cloudlets with spatial flexibility for workload dis-

- *S. Chen and F. Liu are with the National Engineering Research Center for Big Data Technology and System, the Services Computing Technology and System Lab, Cluster and Grid Computing Lab in the School of Computer Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, 430074, China. E-mail: {shutongcs, fmliu}@hust.edu.cn.*
- *L. Jiao is with the Department of Computer and Information Science, University of Oregon, Eugene, OR, USA. E-mail: jiao@cs.uoregon.edu.*
- *L. Wang is with the Department of Computer Science at Vrije Universiteit Amsterdam, The Netherlands and with the Department of Computer Science at Technische Universität Darmstadt, Germany. E-mail: lin.wang@vu.nl.*

Manuscript received 31 Dec. 2020; revised 25 Apr. 2021; accepted 25 May 2021.
(Corresponding author: Fangming Liu.)

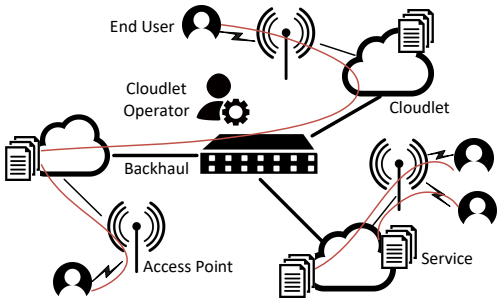


Fig. 1. An example of the cloudlet network.

tribution. Third, they primarily rely on costly, environment-unfriendly, and diesel-powered generators to compensate for the energy reduction required by the grid. As EDR becomes more frequent [6], the current reward from the grid may not suffice to cover the generation cost. And operators are actually cutting the power infrastructure investment by down-sizing the capacity of generators, which could compromise the EDR capability [11]. Thus, it is intriguing to seek other methods to compensate for energy reduction.

For cloudlets, we make two key observations. Our first observation is that a mechanism for cloudlets can exploit the flexibility of service quality degradation in two dimensions, i.e., propagation delay and throughput, avoiding the need for rejecting workload in peak periods. As shown in Fig. 1, cloudlets are often connected to one another via local area networks [12], with network delay usually one order of magnitude smaller than the remote clouds. Therefore, end-user requests can be moved and processed at locations different from where they originate with moderate additional delay. Moreover, the cloudlet operator can provide options for the tenants to reduce their application throughput, i.e., the number of requests per time slot that each service provider can process. A procurement-auction-based mechanism, for example, can let each tenant bid with a cost function of the delay degradation and throughput degradation tolerance, and let the operator choose the desired delay quality and throughput quality for each tenant and distribute the workload with limited computing resources in the cloudlet network accordingly. Inspired by divisible-goods auctions with finite-capacity suppliers [13], this design achieves adaptability in procurements; compared to other methods such as "supply functions", it is also function-based, but allows further adaptability by requiring no unified market-clearing price [11], [14].

Our next observation is that the cloudlet operator can switch off entire cloudlets to compensate for energy reduction [15]. Non-IT appliances (e.g., cooling and lighting) can consume $33\%\sim52\%$ of the total energy of a cloudlet of up to 500 servers [16]; thus, moving workloads around to empty entire cloudlets and shutting them down can eliminate the considerable non-IT energy. Obviously, shutting down idle cloudlets also reduces the amount of allocatable computing resources. At computation-peak hours, the cloudlet operator needs to activate closed cloudlets for more resources. However, switching on and off cloudlets frequently incurs considerable "switching" penalty such as start-up delay, system oscillation, and hardware wear-and-tear [15], [17], [18], which may hurt the power saving and resource preserving benefit. Thus, the cloudlet operator needs to carefully strike the balance among energy reduction, resource preservation, and the switching cost for conducting online auctions with no information about the future available.

In this paper, we propose, to the best of our knowledge, the first online auction mechanism — EdgeDR, specifically for performing power emergency demand response and computing demand response together at the edge. Our proposal features a set of unique designs for cloudlets to meet the split incentives, the adaptability, and the greenness requirements. Based on our observations, we make the following contributions:

We build models to capture the cloudlet operator's and the tenants' costs, design online procurement auctions, and formulate the long-term social cost minimization problem. Particularly, the cloudlet operator has the switching cost incurred by the dynamic control of cloudlets, in addition to cloudlets' maintenance cost and the possible cost of local power generation. Assuming truthfulness of bidding, our online auction mechanism at each time slot solicits a bid from each tenant in the form of a cost function based on the tenant's tolerance range for any additional delay and/or throughput degradation incurred and a per-unit quality degradation penalty within that range. Note that for ensuring the quality of service, the cloudlet operator should strictly follow the tenant's tolerance range for the service quality. We minimize the long-term social cost as a mixed-integer nonlinear program, subject to cloudlet capacities, EDR demand satisfaction, and service delay and throughput tolerance.

We design an online mechanism, EdgeDR, to solve our social cost minimization problem and also a procurement auction mechanism to ensure individual rationality and truthful bids. Our online mechanism solves the one-shot NP-hard problem at each time slot using a primal-dual-based, polynomial-time approximation algorithm, obtains a near-optimal cloudlets switching solution, and then postpones cloudlets switching as required by the near optimum for avoiding excessive switching cost. We rigorously prove that the social cost over time incurred by EdgeDR is no greater than a parameterized constant (i.e., the competitive ratio) times the offline optimal social cost in the worst case. Our procurement auction mechanism provably guarantees individual rationality and truthfulness by following the "bid-monotonic" and the "critical" sufficient conditions.

We further investigate a more practical situation where the cloudlet operator wishes to maximize her own profit through cutting the payment to tenants as much as possible. Moreover, despite that the individual rationality and truthfulness are guaranteed by the payment mechanism, the tenant may drop out of EdgeDR when her received payment cannot meet her own expectation on utility. This outcome reduces the potential of workload distribution and resource allocation in EdgeDR and hence hurts the long-term benefit of the operator. To balance the operator's short-term cost (i.e., payment for tenants) and long-term benefit, we propose a dynamic payment mechanism to adjust the payment and recruit the dropped volunteer according to the urgency of the edge demand response event.

We finally conduct extensive evaluations using real-

world data traces. We simulate a real EDR event in 2018 and use London's underground network to simulate the edge system consisting of APs and cloudlets. The results show that the auction-based EdgeDR has high effectiveness in improving long-term social welfare of both the cloudlet operator and tenants, and the utility of the tenants. A small-scale evaluation also shows a great empirical competitive performance of the proposed algorithm. Moreover, EdgeDR achieves local generation power free and more than 15% resource preserving compared to delay-only mechanism at peak hours. Evaluations on the dynamic payment mechanism confirm high performance on improving the operator's benefit, while preserving the potential of workload distribution and resource allocation in EdgeDR.

2 Model and Problem Formulation

System Settings. We consider a system that consists of a set of distributed heterogeneous cloudlets $\mathcal{N} = \{0, 1, ..., N\}$ which are connected to each other through a wireline backhaul, a set of tenants or service providers $\mathcal{M} = \{0, 1, ..., M\}$ who operate and provide services to end users, and a set of Access Points (APs) $\mathcal{S} = \{0, 1, ..., S\}$ via which end users can access the services deployed at any cloudlet in the system. We denote the capacity of cloudlet k as $R_k, \forall k \in \mathcal{N}$. We represent the time horizon with multiple continuous time slots as $\mathcal{T} = \{0, 1, ..., T\}$. We use a binary variable $x_k(t)$ to indicate whether the cloudlet k is active ($x_k(t) = 1$) or not ($x_k(t) = 0$) at time slot $t, \forall t \in \mathcal{T}$.

The cloudlet operator of this IaaS cloudlet system, as discussed in Sec. 1, determines the workload distribution and resource allocation of each service provider. Let $\lambda_{ij}^l(t)$ denote the workload which is originated via AP i to service provider j at time slot t. And $l \in \mathcal{L} = \{0, 1, ..., L\}$ represents the level of resource allocation: $l = 0$ indicates that the operator allocates the required computing resource and $\lambda_{ij}^l(t)$ has the maximum throughput (in units of requests per time slot); and $l \geqslant 1$ indicates that the operator compresses the allocated computing resource and $\lambda_{ij}^l(t)$ suffers from throughput degradation in this case. We use $y_{ij}^{kl}(t)$ to represent whether workload $\lambda_{ij}^l(t)$ is distributed to cloudlet k ($y_{ij}^{kl}(t) = 1$) or not ($y_{ij}^{kl}(t) = 0$). Without loss of generality, we assume each service provider operates only one service, as multiple services can be treated as multiple "virtual" service providers correspondingly; for the ease of management, we also assume the end users' workload from any AP i for any service j at one time slot is processed at one and only one cloudlet. Note that processing edge service across multiple cloudlets/servers [19], [20], [21] and service migration [22] are also realistic workload management approaches for edge computing. We leave designing edge demand response mechanism for these complicated workload management methods to the future work.

In edge demand response programs, the power EDR event and the computation-peak period may occur simultaneously or separately. During a power EDR event, the cloudlet operator should mandatorily cut down a certain level of its own power demand from the utility according to the signed agreement between the operator and the utility (e.g., PJM [23]). When a computation-peak period comes, the cloudlet operator should manage the resource utilization of each cloudlet to avoid computing resource depletion. Specifically, when receiving the demand response signal, the cloudlet operator can meet the demand in three ways, workload distribution and resource allocation, cloudlet activation status management, and local power generation management.

TABLE 1
List of Notations

Notations	Description
M	# of service providers
N	# of cloudlets
S	# of APs
T	# of time slots
L	# of resource allocation level
$\lambda_{ij}(t)$	workload originated via AP i to service provider j at t
$D_{ij}(t)$	propagation delay tolerance of $\lambda_{ij}(t)$
$Q_{ij}(t)$	throughput requirement of $\lambda_{ij}(t)$
$P_{EDR}(t)$	EDR requirement at time slot t
p	fuel cost of the local generator for one-unit power
$x_k(t)$	cloudlet k is active at time slot t or not
$y_{ij}^{kl}(t)$	$\lambda_{ij}(t)$ is distribution to cloudlet k with resource allocation l or not
$z_{ij}(t)$	propagation delay degradation of $\lambda_{ij}(t)$
$q_{ij}(t)$	throughput degradation of $\lambda_{ij}(t)$
$r_{ij}(t)$	payment to $\lambda_{ij}(t)$
$a_{ij}^{kl}(t)$	value evaluation of allocating $\lambda_{ij}(t)$ to cloudlet k with resource allocation l
$\theta_{ij}(t)$	maximum propagation delay degradation which $\lambda_{ij}(t)$ can tolerate
$\varrho_{ij}(t)$	maximum throughput degradation which $\lambda_{ij}(t)$ can tolerate
$c_{ij}(t)$	per-unit quality degradation penalty when the propagation delay exceeds $D_{ij}(t)$ and/or the throughput does not meet $Q_{ij}(t)$
$u_g(t)$	the amount of local generation
α_k	start-up cost of activating the cloudlet k
ς	maintaining cost of one active cloudlet
$e_k(t)$	power consumption of cloudlet k at time slot t
$\omega_{ij}^l(t)$	the difference between the throughput achieved by resource allocation l and throughput requirement Q_{ij}^t
$d_{ij}^k(t)$	the difference between propagation delay tolerance $D_{ij}(t)$ and propagation delay from AP i to cloudlet k

Auction Design. As illustrated in Fig. 2, within the edge demand response event, the cloudlet operator, who acts as the auctioneer, solicits a bid from each service provider at the beginning of each time slot t. Besides the workload $\lambda_{ij}(t)$, the propagation delay tolerance $D_{ij}(t)$, and the throughput requirement $Q_{ij}(t)$, service provider j voluntarily submits a service quality reduction bid as $\{(\theta_{ij}, \varrho_{ij}, c_{ij})\}$, where θ_{ij} is the maximum propagation delay degradation, ϱ_{ij} is the maximum throughput degradation, and c_{ij} is the per-unit quality degradation penalty when the propagation delay exceeds D_{ij} or when the throughput does not meet Q_{ij}. Here, both propagation delay and throughput degradations are normalized and have the same magnitude, and c_{ij} is the price which translates the level of quality degradation to a monetary penalty. And no matter the service provider participates in the edge demand response or not, her thresholds of service quality will be strictly followed by the cloudlet operator.

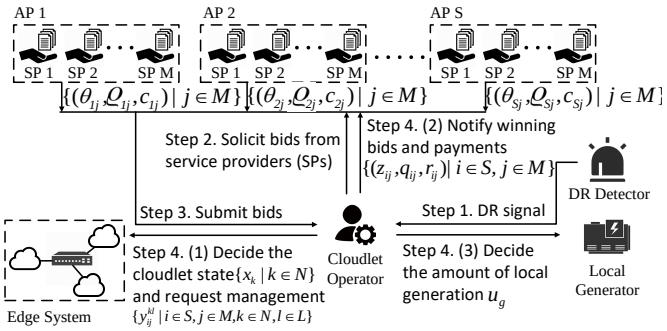


Fig. 2. An illustration of our auction mechanism in one round.

After receiving the bids, the operator determines the cloudlet activation status $\{x_k(t)|k \in \mathcal{N}\}$ and the workload management $\{y_{ij}^{kl}(t)|i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N}, l \in \mathcal{L}\}$, notifies the winning bids and payments $\{(z_{ij}(t), q_{ij}(t), r_{ij}(t))|i \in \mathcal{S}, j \in \mathcal{M}\}$, and sets the amount of local generation $u_g(t)$. As EDR event usually requires the cloudlet to reduce no more than 25% of power consumption, local generators equipped by each cloudlet are powerful enough for meeting the requirement [24], [25]. For simplicity, we regard these generators as a large virtual generator and we only consider the total amount of generation. The goal of the presented mechanism, EdgeDR, is to meet the EDR requirement, avoid running out of computing resources, and minimize the long-term social operational cost which consists of the operational costs of both the operator and the service provider. Note that EdgeDR works no matter whether the power EDR event and the computation-peak period occur simultaneously or separately.

Operational Cost of the Cloudlet Operator. The operator's total operational cost consists of the following components: 1) the local generation cost of $pu_g(t)$, where p is the fuel cost of the local generator for one-unit power; 2) the switching cost of $\sum_k \alpha_k[x_k(t) - x_k(t-1)]^+$, where $[\cdot]^+ = max\{0, \cdot\}$, and α_k is the start-up cost of activating the cloudlet k, representing the standard definition [15]; 3) the maintenance cost for active cloudlets of $\varsigma \sum_k x_k(t)$, where ς is the sunk cost [26] of maintaining one active cloudlet at one time slot, that is, if a cloudlet is active at one time slot, ς has to be paid for maintenance and cannot be recovered; 4) the payment of $\sum_i \sum_j r_{ij}(t)$ to service providers.

Operational Cost of Service Providers. The operational cost of service provider j is $\sum_i (f_{ij}(z_{ij}(t)) + h_{ij}(q_{ij}(t)) - r_{ij}(t) - \sum_k a_{ij}^k(t) y_{ij}^k(t))$. For the end user workload submitted from AP i, $r_{ij}(t)$ is the payment received from the operator; $a_{ij}^{kl}(t)$ indicates service provider j's evaluation of the value of distributing such workload to cloudlet k with resource allocation l, which could capture the varying levels of service reliability for different workload management, for example; $f_{ij}(z_{ij}(t))$ is the penalty function of the propagation delay violation $z_{ij}(t)$ and $h_{ij}(q_{ij}(t))$ is the penalty function of the throughput violation $q_{ij}(t)$:

$$f_{ij}(z_{ij}(t)) = \begin{cases} c_{ij}(t) z_{ij}(t), & z_{ij}(t) \leqslant \theta_{ij}(t) \\ +\infty, & z_{ij}(t) > \theta_{ij}(t) \end{cases},$$

$$h_{ij}(q_{ij}(t)) = \begin{cases} c_{ij}(t) q_{ij}(t), & q_{ij}(t) \leqslant \varrho_{ij}(t) \\ +\infty, & q_{ij}(t) > \varrho_{ij}(t) \end{cases}.$$

Problem Formulation. Having the operational costs of the cloudlet operator and the service providers, we formulate the social cost minimization (or the social welfare maximization, equivalently [27]) problem to determine the workload distribution and resource allocation, the activation status of each cloudlet, the delay degradation and the throughput degradation of each service, and the amount of local generation, where the payments of the operator and the service providers cancel one another:

$$\max \quad \sum_t \Big(\sum_{i,j,k,l} a_{ij}^{kl} y_{ij}^{kl}(t) - \sum_{i,j} f_{ij}(z_{ij}(t)) - \sum_{i,j} h_{ij}(q_{ij}(t)) - pu_g(t) - \varsigma \sum_k x_k(t) - \sum_k \alpha_k [x_k(t) - x_k(t-1)]^+ \Big) \tag{1}$$

$$\text{s.t.} \quad \sum_{i,j,l} \lambda_{ij}^l(t) y_{ij}^{kl}(t) \leqslant R_k x_k(t), \forall k \in \mathcal{N}, \forall t \in \mathcal{T} \tag{1a}$$

$$\sum_k e_k(t) \leqslant P_{EDR}(t) + u_g(t), \forall t \in \mathcal{T} \tag{1b}$$

$$\sum_k d_{ij}^k(t) \sum_l y_{ij}^{kl}(t) \leqslant z_{ij}(t), \forall i \in \mathcal{M}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \tag{1c}$$

$$\sum_l \omega_{ij}^l(t) \sum_k y_{ij}^{kl}(t) \leqslant q_{ij}(t), \forall i \in \mathcal{M}, \forall j \in \mathcal{M} \tag{1d}$$

$$\sum_{k,l} y_{ij}^{kl}(t) \leqslant 1, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \tag{1e}$$

$$x_k(t) \in \{0,1\}, \forall k \in \mathcal{N}, \forall t \in \mathcal{T} \tag{1f}$$

$$y_{ij}^{kl}(t) \in \{0,1\}, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall k \in \mathcal{N}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T} \tag{1g}$$

$$0 \leqslant z_{ij}(t) \leqslant \theta_{ij}(t), \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \tag{1h}$$

$$0 \leqslant q_{ij}(t) \leqslant \varrho_{ij}(t), \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \tag{1i}$$

$$u_g(t) \geqslant 0, \forall t \in \mathcal{T} \tag{1j}$$

For clarity, the important notations are listed in Table 1. Here, the power consumption of cloudlet k is represented by $e_k(t) = (\Pi_k P_{idle}^k x_k(t) + (P_{peak}^k - P_{idle}) \sum_{i,j,l} \lambda_{ij}^l(t) y_{ij}^{kl}(t)) \cdot \text{PUE}_k$, where PUE_k refers to the power usage effectiveness (PUE) of cloudlet k, Π_k is the number of servers in cloudlet k, and P_{idle}^k and P_{peak}^k denote the server's power in the idle and the peak utilization, respectively. For simplicity, we reformulate $e_k(t)$ as $e_k(t) = \beta_k \sum_{i,j} \lambda_{ij}(t) y_{ij}^k(t) + \gamma_k x_k(t)$, where $\beta_k = (P_{peak}^k - P_{idle})\text{PUE}_k$ and $\gamma_k = \Pi_k P_{idle}^k \text{PUE}_k$. In the constraint (1c), $d_{ij}^k(t) = \max\{0, \hat{d}_i^k - D_{ij}(t)\}$, where $\hat{d}_i^k$ is the propagation delay from AP i to cloudlet k. $d_{ij}^k(t)$ denotes the difference between the delay tolerance $D_{ij}(t)$ and the propagation delay from AP i to cloudlet k, and $d_{ij}^k(t) = 0$ indicates the propagation delay between i and k is within the delay tolerance $D_{ij}(t)$. Similarly, in constraint (1d), $\omega_{ij}^l(t)$ represents the difference between the throughput requirement $Q_{ij}(t)$ and the throughput achieved by resource allocation l, and $\omega_{ij}^l(t) = 0$ indicates the throughput meets the requirement $Q_{ij}(t)$. We only consider that different workload distribution schemes affect the propagation latency between the end user and the cloudlet. The service provider's requirement of computation time, which is affected by the resource allocation, is taken into account in the throughput requirement $Q_{ij}(t)$.

In problem (1), the constraint (1a) ensures that the resource utilization of active cloudlet k is under its capacity; if k is switched off, any workload cannot be distributed to it. The constraint (1b) guarantees that the total power demand

does not exceed the sum of the EDR power cap and the local generation. The constraints (1c) and (1h) guarantee that the delay degradation does not exceed the bound $\theta_{ij}(t)$. The constraints (1d) and (1i) guarantee that the throughput degradation does not exceed the bound $\varrho_{ij}(t)$. The constraint (1e) and constraint (1g) guarantee that workload is distributed to at most one cloudlet. Later in Sec. 3, we will show that the workload is indeed ensured to be distributed to one and only one cloudlet by the algorithm design.

In this work, we assume all cloudlets are heterogeneous that each cloudlet k has her own capacity R_k, start-up cost α_k, per-unit resource price μ_k, etc. Also, since the propagation delay between cloudlets depends on their geographical distance, there should be a difference in the propagation delay from one pair of cloudlets to another pair. We note that we assume the servers associated to different cloudlets are homogeneous, but our work can be extended to the case of heterogeneous servers. If considering the heterogeneity of servers, the power consumption and the throughput brought by different workload allocations are different, and the cloudlet operator also needs to decide which server the workload should be allocated to. This, however, will only bring an additional dimension of decision making, which can be easily added to our formulation and algorithms, and will not hurt the effectiveness of our approach.

Algorithmic Challenges. Solving the social welfare maximization problem in an online manner is highly challenging. Although the cloudlet operator can switch off cloudlets for energy saving, high switching cost may be incurred by frequent cloudlet activations. Without knowledge of the future inputs, it is nontrivial for the cloudlet operator to determine the cloudlet activation status at each time slot, because a decision at a time slot will influence the switching cost between that time slot and its next time slot; as the next time slot has not yet arrived, it is not easy to make a good decision for the current time slot. Note that the problem (1) subsumes a case of $z_{ij}(t) = 0$, $q_{ij}(t) = 0$, and $u_g(t) = 0$, $\forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T}$, where it reduces to an NP-hard multi-dimensional knapsack problem [28]. So even in the offline scenario where all the inputs are known in advance, the problem is still NP-hard in general. To guarantee individually rational and truthful bidding, we would need to leverage the Vickrey-Clark-Groves (VCG) auctions; however, such NP-hardness makes the direct utilization of VCG impossible, as it requires the social welfare maximization in one round to be solved in polynomial time [5].

To overcome such challenges and design a computationally efficient mechanism, we divide the social welfare into two parts: 1) the switching cost of $C_{SC}^t = \sum_k \alpha_k [x_k(t) - x_k(t-1)]^+$, the only term that is coupled over time, depending on the past cloudlet activation status $\boldsymbol{x}(t-1)$ and the current status $\boldsymbol{x}(t)$; 2) the non-switching welfare of $W_{-SC}^t = \sum_{i,j,k,l} a_{ij}^{kl}(t) y_{ij}^{kl}(t) - \sum_{i,j} f_{ij}(z_{ij}(t)) - \sum_{i,j} h_{ij}(q_{ij}(t)) - p u_g(t) - \varsigma \sum_k x_k(t)$ which can be obtained at each time slot t independently if the activation status $\boldsymbol{x}(t)$ is given. Leveraging the separation of social welfare, we first study the workload management (including workload distribution and resource allocation) and winner determination problem at each round with given cloudlet activation status $\boldsymbol{x}(t)$. A primal-dual-based approximation algorithm for the one-shot problem is proposed in Sec. 3. Based on the one-shot solution, in Sec. 4, we further present an online algorithm that decomposes the long-term social welfare maximization problem into a series of one-round problems. Both algorithms are in polynomial time.

3 Algorithm for One-Round Auction

3.1 Primal-Dual-Based Algorithm Design

In this section, we focus on the one-shot problem at a single time slot on the assumption that the cloudlet activation status $\boldsymbol{x}(t)$ is given. As optimizing the non-switching welfare W_{-SC}^t is NP-hard, instead of pursuing an optimal solution, we propose a primal-dual-based algorithm to obtain an efficient approximate solution within polynomial time.

For simplicity of the presentation, in the following, we omit the time index t of all the parameters and the variables. We reformulate the problem as follows:

$$\max \quad \sum_{i,j,k,l} a_{ij}^{kl} y_{ij}^{kl} - \sum_{i,j} f_{ij}(z_{ij}) - \sum_{i,j} h_{ij}(q_{ij}) - g(u) \tag{2}$$

$$\text{s.t.} \quad \sum_{i,j,l} \lambda_{ij}^l y_{ij}^{kl} \leqslant R_k x_k, \forall k \in \mathcal{N} \tag{2a}$$

$$\sum_{i,j,k,l} \beta_k \lambda_{ij}^l y_{ij}^{kl} \leqslant u \tag{2b}$$

$$\sum_k d_{ij}^k \sum_l y_{ij}^{kl} \leqslant z_{ij}, \forall i \in \mathcal{M}, \forall j \in \mathcal{M} \tag{2c}$$

$$\sum_l \omega_{ij}^l \sum_k y_{ij}^{kl} \leqslant q_{ij}, \forall i \in \mathcal{M}, \forall j \in \mathcal{M} \tag{2d}$$

$$\sum_{k,l} y_{ij}^{kl} \leqslant 1, \forall i \in \mathcal{S}, \forall j \in \mathcal{M} \tag{2e}$$

$$y_{ij}^{kl} \in \{0,1\}, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall k \in \mathcal{N}, \forall l \in \mathcal{L} \tag{2f}$$

$$u, z_{ij}, q_{ij} \geqslant 0, \forall i \in \mathcal{S}, \forall j \in \mathcal{M} \tag{2g}$$

Note that since the cloudlet activation status $\boldsymbol{x}$ is given, the maintenance cost $\varsigma \sum_k x_k$ is a constant and thus omitted. We also reformulate the local generation cost:

$$g(u) = \begin{cases} 0, & u \leqslant P'_{EDR} \\ p(u - P'_{EDR}), & u \geqslant P'_{EDR} \end{cases},$$

and replace (1b) by (2b), where $P'_{EDR} = P_{EDR} - \sum_k \gamma_k x_k$ and u is the active power caused by workload management. When $u \leqslant P'_{EDR}$, i.e., the active power demand is no larger than the EDR cap minus idle power consumption, the local generation cost is zero; otherwise, when $u \geqslant P'_{EDR}$, the extra power has to be met by the local generation.

By relaxing the binary variables y_{ij}^{kl} to the continuous non-negative variables and introducing the dual variables μ_k, φ, ξ_{ij}, ϖ_{ij}, and ρ_{ij} for the constraints (2a)-(2e), respectively, we obtain the dual problem [29] of the relaxed problem (2):

$$\min \quad \sum_k R_k x_k \mu_k + \sum_{i,j} \rho_{ij} + \sum_{i,j} f_{ij}^*(\xi_{ij}) + \sum_{i,j} h_{ij}^*(\varpi_{ij}) + g^*(\varphi) \tag{3}$$

$$\text{s.t.} \quad \rho_{ij} \geqslant a_{ij}^{kl} - \left(\lambda_{ij}^l \mu_k + d_{ij}^k \xi_{ij} + \omega_{ij}^l \varpi_{ij} + \beta_k \lambda_{ij}^l \varphi\right), \quad \forall i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N}, l \in \mathcal{L} \tag{3a}$$

$$\mu_k, \varphi, \xi_{ij}, \varpi_{ij}, \rho_{ij} \geqslant 0 \tag{3b}$$

where $f_{ij}^*(\xi_{ij})$, $h_{ij}^*(\varpi_{ij})$, and $g^*(\varphi)$ are the Fenchel conjugates [29], [30] of the functions $f_{ij}(z_{ij})$, $h_{ij}(q_{ij})$, and $g(u)$, respectively:

$$\begin{aligned} f_{ij}^*(\xi_{ij}) &= \sup_{z_{ij}\geqslant 0}\{z_{ij}\xi_{ij} - f_{ij}(z_{ij})\} \\ &= \begin{cases}\theta_{ij}(\xi_{ij}-c_{ij}), & \xi_{ij}^t \geqslant c_{ij} \\ 0, & \xi_{ij}^t \leqslant c_{ij}\end{cases}, \\ h_{ij}^*(\varpi_{ij}) &= \sup_{q_{ij}\geqslant 0}\{q_{ij}\varpi_{ij} - h_{ij}(q_{ij})\} \\ &= \begin{cases}\varrho_{ij}(\varpi_{ij}-c_{ij}), & \varpi_{ij}^t \geqslant c_{ij} \\ 0, & \varpi_{ij}^t \leqslant c_{ij}\end{cases}, \\ g^*(\varphi) &= \sup_{u\geqslant 0}\{u\varphi - g(u)\} \\ &= \begin{cases}\varphi P'_{EDR}, & u \leqslant P'_{EDR} \\ pP'_{EDR}, & u \geqslant P'_{EDR} \text{ and } \varphi \leqslant p \\ +\infty, & u \geqslant P'_{EDR} \text{ and } \varphi > p\end{cases}. \end{aligned}$$

Following the idea of primal-dual optimization, y_{jik} remains zero unless its corresponding dual constraint (3a) becomes tight. We let ρ_{ij} be the greater quantity between 0 and the right-hand side of constraint (3a):

$$\rho_{ij} = \max\left\{0, \max_{k,l}\{a_{ij}^{kl} - (\lambda_{ij}^l\mu_k + d_{ij}^k\xi_{ij} + \omega_{ij}^l\varpi_{ij} + \beta_k\lambda_{ij}^l\varphi)\}\right\}. \tag{4}$$

When $\rho_{ij} > 0$, the operator distributes the workload λ_{ij} to the cloudlet k with resource allocation l which maximizes the right hand side of (3a), i.e., $(k,l) = \arg\max\{a_{ij}^{kl} - (\lambda_{ij}^l\mu_k + d_{ij}^k\xi_{ij} + \omega_{ij}^l\varpi_{ij} + \beta_k\lambda_{ij}^l\varphi)\}$; otherwise, the workload λ_{ij} will not be managed, which, however, will not actually happen as described below.

The principle of the proposed approximation algorithm is as follows: If we interpret the dual variable μ_k as the per-unit resource price of cloudlet k, ξ_{ij} as the per-unit delay penalty, ϖ_{ij} as the per-unit throughput degradation penalty, and φ as the per-unit local generation cost, then the term $\lambda_{ij}^l\mu_k + d_{ij}^k\xi_{ij} + \omega_{ij}^l\varpi_{ij} + \beta_k\lambda_{ij}^l\varphi$ can be regarded as the total cost caused by distributing workload λ_{ij} to cloudlet k with resource allocation l, and the right hand side of (3a) can be regarded as the social utility of this management. As defined in Sec. 2, c_{ij} is the per-unit penalty when the delay exceeds the delay tolerance D_{ij} and/or throughput is lower than the requirement Q_{ij}. And p is the fuel cost of local generation. Then we let $\xi_{ij} = \varpi_{ij} = c_{ij}$ and $\varphi = p$. Given that as cloudlets are heterogeneous, the resource price μ may vary from one cloudlet to another. μ_k can be set following historical processing statistics, and it should be bounded by $\mu_k < \min_{i,j,l}\{(a_{ij}^{kl} - (d_{ij}^k + \omega_{ij}^l)c_{ij})/\lambda_{ij}^l - \beta_k p\}$ to guarantee that every unit of end user workload can be processed. Based on this idea, the algorithm of workload management and winner determination ensures that the workload λ_{ij} is processed with maximum social utility and hence guarantees the social welfare maximization.

Algorithm 1 illustrates the one-round auction framework. In Line 1, O is the set of winners, B is the set of unallocated workload, ζ_k presents the resource utilization of cloudlet k, $\mathcal{N}_{ij}$ indicates the candidate cloudlets and resource allocation level where workload λ_{ij} can be managed, and $\rho_{ij}^{kl} = a_{ij}^{kl} - (\lambda_{ij}^l\mu_k + d_{ij}^k\xi_{ij} + \omega_{ij}^l\varpi_{ij} + \beta_k\lambda_{ij}^l\varphi)$. Line 3 and Lines 5-7 determine the workload distribution and resource allocation y_{ij}^{kl}, winner's delay penalty $d_{ij}^k c_{ij}$ and throughput degradation penalty $\omega_{ij}^l c_{ij}$ based on the above rationale. After that, Lines 3-4 and Lines 8-9 calculate the payment to the winner, whose effectiveness is proved in the later Theorem 3. And then, Lines 11-14 update the cloudlet k^*'s resource utilization ζ_{k^*}, the active power consumption u, the workload management index (i^*, j^*, k^*, l^*), and the set of unallocated workload B, respectively.

Algorithm 1 The workload allocation, workload compression, and winner determination framework

1: Initialize: $z_{ij} = 0$, $q_{ij} = 0$, $y_{ij}^{kl} = 0$, $r_{ij} = 0$, and $u = 0$; $O = \emptyset$, $B = \{(i,j)|i \in \mathcal{S}, j \in \mathcal{M}\}$, $\mathcal{N}_{ij} = \{(k,l)|d_{ij}^k \leqslant \theta_{ij},\ \omega_{ij}^l \leqslant \varrho_{ij}\}$, $\zeta_k = 0$, $\xi_{ij} = c_{ij}$, $\varphi = p$, $\forall i \in \mathcal{S}$, $\forall j \in \mathcal{M}$, $\forall k \in \mathcal{N}$, $\forall l \in \mathcal{L}$;
2: **while** $B \neq \emptyset$ **do**
3: $(i^*, j^*, k^*, l^*) = \arg\max\{\rho_{ij}^{kl}|\zeta_k + \lambda_{ij}^l \leqslant R_k x_k, (i,j) \in B, (k,l) \in \mathcal{N}_{ij}\}$;
4: $(i^-, j^-, k^-, l^-) = \arg\max\{\rho_{ij}^{kl}|\zeta_k + \lambda_{ij}^l \leqslant R_k x_k, (i,j) \in B, (k,l) \in \mathcal{N}_{ij} \setminus \{(k^*, l^*) \in \mathcal{N}_{i^*j^*}\}\}$;
5: $y_{i^*j^*}^{k^*l^*} = 1$;
6: $z_{i^*j^*} = d_{i^*j^*}^{k^*}$;
7: $q_{i^*j^*} = \omega_{i^*j^*}^{l^*}$;
8: **if** $z_{i^*j^*} > 0$ **or** $q_{i^*j^*} > 0$ **do**
9: $r_{i^*j^*} = c_{i^*j^*}(z_{i^*j^*} + q_{i^*j^*}) + (\rho_{i^*j^*}^{k^*l^*} - \rho_{i^-j^-}^{k^-l^-})$;
10: **end if**
11: $\zeta_{k^*} = \zeta_{k^*} + \lambda_{i^*j^*}^{l^*}$;
12: $u = u + \beta_{k^*}\lambda_{i^*j^*}^{l^*}$;
13: $O = O \cup (i^*, j^*, k^*, l^*)$;
14: $B = B \setminus (i^*, j^*)$;
end while
15: $W_{-SC} = \sum_{i,j,k,l} a_{ij}^{kl} y_{ij}^{kl} - \sum_{i,j} f_{ij}(z_{ij}) - \sum_{i,j} h_{ij}(q_{ij}) - g(u) - \varsigma\sum_k x_k$;

3.2 Performance Analysis

Theorem 1. *Algorithm 1 generates a feasible solution to both the problem (2) and the problem (3) in polynomial time.*

Proof. For the primal problem (2), the resource limitation (2a) of each cloudlet is satisfied by both Line 3's condition and Line 11. The active power consumption of running workload (2b) is calculated by Line 12. The delay violation caused by workload distribution (2c) is given in Line 6, the throughput violation caused by resource allocation (2d) is given by Line 7, and the definition of set $\mathcal{N}_{ij}$ in Line 1 guarantees both the delay violation and throughput violation do not exceed the tolerance submitted by service provider j at AP i. It is easy to show that the constraint (2g) is satisfied. For any $i \in \mathcal{S}$, $j \in \mathcal{M}$, $k \in \mathcal{N}$, and $l \in \mathcal{L}$, the variable y_{ij}^{kl} is initialized to be 0 and set to 1 when request λ_{ij} is allocated to cloudlet k with resource allocation l, as in Line 5, hence the constraints (2e) and (2f) are satisfied. For the problem (3), the constraint (3a) is satisfied by Line 3.

For the time complexity, as the termination condition of the `while` loop in Line 2 is $B = \{(i,j)|i \in \mathcal{S}, j \in \mathcal{M}\} \neq \emptyset$, the loop runs SM times. And in the `while` loop, Lines 3-4 run at most $SMNL$ times. We conclude that the total time complexity of Algorithm 1 is $O(S^2M^2NL)$. □

Theorem 2. *Algorithm 1 is a $\frac{\sigma}{\sigma-1}$-approximation algorithm to the problem (2), i.e., the social welfare obtained by Algorithm 1 is at least $\frac{\sigma-1}{\sigma}$ times the optimal social welfare in the problem (2), where $\frac{1}{\sigma} = \frac{\sum_k R_k x_k \mu_k + pP'_{EDR}}{\sum_k R_k x_k \mu_k + pP'_{EDR} + \sum_i \sum_j \rho_{min}}$ and $\rho_{min} = \min_{i,j,k,l}\{a_{ij}^{kl} - (\lambda_{ij}^l \mu_k + d_{ij}^k \xi_{ij} + \omega_{ij}^l \varpi_{ij} + \beta_k \lambda_{ij}^l \varphi) | \lambda_{ij}^l \leqslant R_k x_k, d_{ij}^k \leqslant \theta_{ij}, \omega_{ij}^l \leqslant \varrho_{ij}, i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N}, l \in \mathcal{L}\}$.*

Proof. Let P and D be the overall objective value of the primal problem (2) and the dual problem (3) obtained by Algorithm 1, respectively. And P_0 and D_0 is the initial values of problem (2) and (3), respectively. According to Algorithm 1 and the definition (4), we have $P_0 = 0$, $D_0 = \sum_k R_k x_k \mu_k + pP'_{EDR}$, $D = \sum_k R_k x_k \mu_k + \sum_{i,j} \rho_{ij} + pP'_{EDR}$, and $P = \sum_{i,j} \rho_{ij} + \sum_{i,j} \lambda_{ij}^{l_{ij}} \mu_{k_{ij}} + \sum_{i,j} p\beta_{k_{ij}} \lambda_{ij} - g(u)$, where l_{ij} indicates the resource allocation level of workload λ_{ij} and k_{ij} indicates the distributed cloudlet of workload λ_{ij}. Note that since $u = \sum_{i,j} \beta_{k_{ij}} \lambda_{ij}^{l_{ij}}$, we have $\sum_{i,j} p\beta_{k_{ij}} \lambda_{ij}^{l_{ij}} - g(u) \geqslant 0$, and $P = P - P_0 \geqslant D - D_0$.

Let ρ_{min} be the minimum management utility over all workload management choices, i.e., $\rho_{min} = \min_{i,j,k,l}\{a_{ij}^{kl} - (\lambda_{ij}^l \mu_k + d_{ij}^k \xi_{ij} + \omega_{ij}^l \varpi_{ij} + \beta_k \lambda_{ij}^l \varphi) | \lambda_{ij}^l \leqslant R_k x_k, d_{ij}^k \leqslant \theta_{ij}, \omega_{ij}^l \leqslant \varrho_{ij}, i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N}, l \in \mathcal{L}\}$. It is easy to show that unless all cloudlets are homogeneous, bids and workload of all service providers are the same as well as the propagation delay between each cloudlet and AP, we will have $OPT \geqslant \sum_k R_k x_k \mu_k + pP'_{EDR} + \sum_{i,j} \rho_{min}$. And $D_0 \leqslant \frac{1}{\sigma} OPT$, where $\frac{1}{\sigma} = \frac{\sum_k R_k x_k \mu_k + pP'_{EDR}}{\sum_k R_k x_k \mu_k + pP'_{EDR} + \sum_{i,j} \rho_{min}}$. By the weak duality, we have

$$P \geqslant D - D_0 \geqslant \frac{\sigma - 1}{\sigma} OPT. \qquad \square$$

Here, the parameter σ in the approximation ratio can be re-expressed as $\sigma = 1 + SM\rho_{min}/(\sum_k R_k x_k \mu_k + pP'_{EDR})$, and it is bounded by $[1, 1 + SM\rho_{min}/(K \min_k\{R_k \mu_k\} + pP'_{EDR})]$. We can find that the value of σ in practice depends on the social utility of workload management ρ, the per-unit resource price μ, the per-unit local generation cost p, and the threshold of active power requirement P'_{EDR}. According to the simulation results in Sec. 6, the empirical approximation ratio on average for Algorithm 1 ranges from 1.08 to 2.71, and σ is from 0.58 to 12.5.

Theorem 3. *The proposed procurement auction mechanism achieves individual rationality and truthfulness.*

Proof. We prove the truthfulness of the proposed mechanism referring to the following lemma:

Lemma 1. *[27] According to the Myerson's theorem [31], [32], a reverse auction with bids $\{(c_{ij}, \theta_{ij}, \varrho_{ij}) | i \in \mathcal{S}, j \in \mathcal{M}\}$ and payment $\{r_{ij}\}$ is truthful iff[1]:*

i) The mechanism is bid-monotonic, i.e., $z_{ij} + q_{ij}$ is monotonically non-increasing in c_{ij}, $\forall i \in \mathcal{S}, \forall j \in \mathcal{M}$;

ii) The winners are paid with critical payment, i.e., assume service provider j in AP i wins and her workload is distributed to cloudlet k with resource allocation l, if she reports a new unit-delay penalty c'_{ij}, she will also win and her workload will be also distributed to k with resource allocation l if $c'_{ij}(z_{ij} + q_{ij}) \leqslant r_{ij}$, otherwise she will lose in this round.

1. As discussed in Sec. 2, θ_{ij} and ϱ_{ij} are normalized and have the same magnitude, the bid can be rewritten as $\{(c_{ij}, (\theta_{ij} + \varrho_{ij})) | i \in \mathcal{S}, j \in \mathcal{M}\}$ and analyzed by Myerson's theorem [31], [32] in the one-parameter environment.

i) The mechanism is bid-monotonic: We assume workload λ_{ij} is distributed to cloudlet k with resource allocation l, and the service provider j in AP i wins the auction with delay degradation $z_{ij} = d_{ij}^k$ and throughput degradation $q_{ij} = \omega_{ij}^l$. Consider a case that she reports a larger penalty $c'_{ij} \geqslant c_{ij}$, which makes the workload distribution changes from cloudlet k to k' and the resource allocation changes from l to l', and total penalty varies to $z'_{ij} + q'_{ij} = d_{ij}^{k'} + \omega_{ij}^{l'}$. When $\xi_{ij} = \varpi_{ij} = c_{ij}$, according to the definition of ρ_{ij} and the service provider's winning condition of Algorithm 1, we have $(d_{ij}^k + \omega_{ij}^l - (d_{ij}^{k'} + \omega_{ij}^{l'}))c_{ij} \leqslant (a_{ij}^{kl} - \lambda_{ij}^l \mu_k - p\beta_k \lambda_{ij}^l) - (a_{ij}^{k'l'} - \lambda_{ij}^{l'} \mu_{k'} - p\beta_{k'} \lambda_{ij}^{l'})$. And when $\xi_{ij} = \varpi_{ij} = c'_{ij}$, we have $(d_{ij}^k + \omega_{ij}^l - (d_{ij}^{k'} + \omega_{ij}^{l'}))c_{ij} \geqslant (a_{ij}^{kl} - \lambda_{ij}^l \mu_k - p\beta_k \lambda_{ij}^l) - (a_{ij}^{k'l'} - \lambda_{ij}^{l'} \mu_{k'} - p\beta_{k'} \lambda_{ij}^{l'})$. Hence $d_{ij}^k + \omega_{ij}^l - (d_{ij}^{k'} + \omega_{ij}^{l'}) \geqslant 0$, i.e., $z_{ij} + q_{ij} \geqslant z'_{ij} + q'_{ij}$. That is, $z_{ij} + q_{ij}$ is monotonically non-increasing with the increase of c_{ij}.

ii) The payment is critical: According to the definition of ρ_{ij}^{kl} and the winning condition of Algorithm 1, we have $\rho_{ij}^{kl} \geqslant \rho_{i^- j^-}^{k^- l^-}$ and $r_{ij} = c_{ij}(z_{ij} + q_{ij}) + \rho_{ij}^{kl} - \rho_{i^- j^-}^{k^- l^-} = a_{ij}^{kl} - (\lambda_{ij}^l \mu_k + p\beta_k \lambda_{ij}^l) - \rho_{i^- j^-}^{k^- l^-}$, where (i^-, j^-) is the winner and her workload is distributed to k^- with resource allocation l^- if (i, j, k, l) is excluded from the candidate set. When the winner (i, j) reports a new penalty c'_{ij} which makes $c'_{ij}(z_{ij} + q_{ij}) \leqslant r_{ij}$, we have $c'_{ij}(z_{ij} + q_{ij}) \leqslant a_{ij}^{kl} - (\lambda_{ij}^l \mu_k + p\beta_k \lambda_{ij}^l) - \rho_{i^- j^-}^{k^- l^-}$, i.e., $\rho'^{kl}_{ij} \geqslant \rho_{i^- j^-}^{k^- l^-}$. Therefore, (i, j) will also win and her request is also distributed to k with resource allocation l when the new reported penalty c'_{ij} ensuring $c'_{ij}(z_{ij} + q_{ij}) \leqslant r_{ij}$; otherwise, (i, j) will lose in this round. That is, the payment obtained by Algorithm 1 is critical.

Above all, we can conclude that Algorithm 1 achieves truthfulness. And the proposed mechanism also achieves individual rationality as $r_{ij} - c_{ij}(z_{ij} + q_{ij}) = \rho_{ij} - \rho_{i^- j^-} \geqslant 0$, i.e., the utility of winner is no less than zero. $\square$

4 Online Algorithm for Long-Term Problem

4.1 Online Algorithm Design

In this section, we present an online algorithm to determine the cloudlet activation state to achieve the long-term social welfare maximization, based on the one-shot solution from Algorithm 1 at each time slot.

Due to the lack of a priori knowledge, the operator should be careful about the control of cloudlets at each time slot. One solution is to pursue only the long-term optimum non-switching welfare by obtaining the maximum non-switching value among all possible cloudlet statuses at each time slot. However, this solution is hard to accomplish, and may result in aggressive switching of cloudlets and thus hurt the long-term social welfare. First, there are totally 2^N possible cloudlet status combinations in the edge system consisting of N heterogeneous cloudlets, which makes obtaining the optimum solution at one time slot computationally prohibitive. Moreover, even if one has the optimal solution, not considering the switching cost can lead to the result that, for example, the local optimal solution suggests

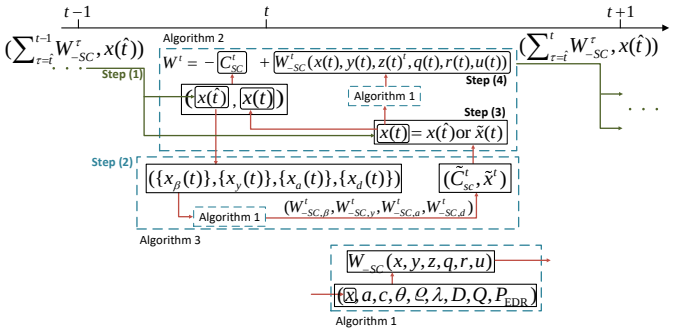


Fig. 3. An illustration of our online framework.

to shut down and then activate a cloudlet, while better long-term solutions may let this cloudlet remain activated to save the switching cost over time.

We propose an online algorithm for obtaining cost-efficient solutions while avoiding aggressive cloudlet switching, as shown in Algorithm 2 with a component shown in Algorithm 3. The principle is as follows. In Algorithm 3, we obtain a near-optimal cloudlet activation status. And then in Algorithm 2, we postpone the cloudlet switching required by the near-optimal solution, until the surplus (i.e., the cumulative non-switching welfare) exceeds the switching cost obviously or until there is no feasible solution with the unchanged activation state. Given that Algorithm 1, as a component of Algorithm 2 and Algorithm 3, is used to obtain the workload management and winner determination with corresponding non-switching welfare when the cloudlet status is given.

Fig. 3 shows the illustration of the proposed online framework. We first describe how to obtain a near-optimal solution in Algorithm 3. Assume the last switching time slot before the current t is $\hat{t}$. Based on the formulation (2), given the previous cloudlet activation state $\boldsymbol{x}(t-1) = \boldsymbol{x}(\hat{t})$ (step (1) in Fig. 3), there are four possible variations in cloudlet status $\boldsymbol{x}(t)$ by which we may achieve higher non-switching welfare W^t_{-SC}: 1) Closing cloudlets with lower power efficiency to reduce active power of processing workload and total static power of cloudlets; 2) Closing cloudlets with less distributed workload to cut down static power while ensuring limited utility decrease; 3) Opening cloudlets with higher "reliability" to increase the utility of workload management; 4) Opening cloudlets with less propagation delay to other connected cloudlets to reduce the possible delay penalty. As indicated by step (2) in Fig. 3 and Lines 1 to 5 in Algorithm 3, we compare the non-switching welfare obtained by the above variations in cloudlet status and choose the status with the highest non-switching welfare, denoted by $\tilde{\boldsymbol{x}}(t)$, as the near-optimal solution. And the switching cost caused by cloudlet switching is $\tilde{C}^t_{SC} \triangleq \tilde{C}^t_{SC}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(\hat{t}))$.

In Algorithm 2, at time slot t, we compare the cumulative non-switching welfare from $\hat{t}$ to $t-1$, $\sum_{\tau=\hat{t}}^{t-1} W^\tau_{-SC}$, with the switching cost $\tilde{C}^t_{SC}$ obtained by Algorithm 3: if $\sum_{\tau=\hat{t}}^{t-1} W^\tau_{-SC}$ exceeds κ times $\tilde{C}^t_{SC}$ or there is no feasible solution with the status $\boldsymbol{x}(\hat{t})$, we will vary the cloudlet activation status from $\boldsymbol{x}(\hat{t})$ to $\tilde{\boldsymbol{x}}(t)$; otherwise, the activation state is unchanged at time slot t. After deciding the cloudlet status $\boldsymbol{x}(t)$, the corresponding non-switching welfare W^t_{-SC} as well as workload management and winner determination can be obtained by Algorithm 1. The above process is indicated by step (3) and step (4) in Fig. 3 and Lines 6 to 15 in Algorithm 2.

Algorithm 2 The Online Algorithm Framework

1: Define: $t = 1$, $\hat{t} = 0$, $\Delta W = 0$;
2: **while** $t \leqslant T$ **do**
3: Service providers submit requests $\{(\lambda_{ij}(t), D_{ij}(t), Q_{ij}(t)) | i \in \mathcal{S},\ j \in \mathcal{M}\}$ and bids $\{(\theta_{ij}(t), \varrho_{ij}(t), c_{ij}(t)) | i \in \mathcal{S},\ j \in \mathcal{M}\}$;
4: Obtain $\hat{W}^t_{-SC} \triangleq W^t_{-SC}(\boldsymbol{x}(\hat{t}), \boldsymbol{y}(t), \boldsymbol{z}(t), \boldsymbol{q}(t), \boldsymbol{r}(t), u(t))$ by Algorithm 1;
5: Obtain $\tilde{C}^t_{SC}$ and $\tilde{W}^t_{-SC}$ by Algorithm 3;
6: **if** $\Delta W \geqslant \kappa \tilde{C}^t_{SC}$ **or** $W^t_{-SC} = -\infty$ **then**
7: $\boldsymbol{x}(t) = \tilde{\boldsymbol{x}}(t)$;
8: $W^t_{-SC} = \tilde{W}^t_{-SC}$;
9: $\Delta W = W^t_{-SC}$;
10: $\hat{t} = t$;
11: **else**
12: $\boldsymbol{x}(t) = \boldsymbol{x}(\hat{t})$;
13: $W^t_{-SC} = \hat{W}^t_{-SC}$;
14: $\Delta W = \Delta W + W^t_{-SC}$;
15: **end if**
16: The operator notifies the winning bids $\boldsymbol{z}(t)$ and payments $\boldsymbol{r}(t)$, distributes workload according to $\boldsymbol{y}(t)$ with resource allocation $\boldsymbol{q}(t)$, and sets the amount of local generation according to $u(t)$;
17: $t = t + 1$;
18: **end while**

Algorithm 3 Obtaining $\tilde{W}^t_{-SC}(\tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), \tilde{\boldsymbol{z}}(t), \tilde{\boldsymbol{q}}(t), \tilde{u}(t))$ and $\tilde{C}^t_{SC}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(\hat{t}))$

1: Close the active cloudlet in $\boldsymbol{x}(\hat{t})$ one by one according to the descending order of β_k, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W^t_{-SC,\beta}(\boldsymbol{x}_\beta(t), \boldsymbol{y}_\beta(t), \boldsymbol{z}_\beta(t), \boldsymbol{q}_\beta(t), \boldsymbol{r}_\beta(t), u_\beta(t))$;
2: Close the active cloudlet in $\boldsymbol{x}(\hat{t})$ one by one according to the ascending order of $\sum_{i,j} y^k_{ij}(t-1)$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W^t_{-SC,y}(\boldsymbol{x}_y(t), \boldsymbol{y}_y(t), \boldsymbol{z}_y(t), \boldsymbol{q}_y(t), \boldsymbol{r}_y(t), u_y(t))$;
3: Open the closed cloudlet in $\boldsymbol{x}(\hat{t})$ one by one according to the descending order of $\sum_{i,j} a^k_{ij}(t)$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W^t_{-SC,a}(\boldsymbol{x}_a(t), \boldsymbol{y}_a(t), \boldsymbol{z}_a(t), \boldsymbol{q}_a(t), \boldsymbol{r}_a(t), u_a(t))$;
4: Open the closed cloudlet in $\boldsymbol{x}(\hat{t})$ one by one according to the ascending order of $\sum_{i,j} d^k_{ij}$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W^t_{-SC,d}(\boldsymbol{x}_d(t), \boldsymbol{y}_d(t), \boldsymbol{z}_d(t), \boldsymbol{q}_d(t), \boldsymbol{r}_d(t), u_d(t))$;
5: Select the maximum from the above four non-switching welfare, which is indicated by $\tilde{W}^t_{-SC}(\tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), \tilde{\boldsymbol{z}}(t), \tilde{\boldsymbol{q}}(t), \tilde{\boldsymbol{r}}(t), \tilde{u}(t))$;
6: $\tilde{C}^t_{SC}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(\hat{t})) = \sum_k \alpha_k [\tilde{x}_k(t) - x_k(\hat{t})]$;

4.2 Performance Analysis

We first provide a sketched proof to the truthfulness of EdgeDR for the long-term problem as follows: As given by Theorem 3, the proposed one-round procurement auction

mechanism is truthful with a given cloudlet activation status. We can naturally see that, since the cloudlet activation status is determined by the operator and is transparent to the service provider, the auction mechanism for the long term, i.e., EdgeDR, is truthful at each time slot under any cloudlet activation status.

The time complexities of Algorithm 2 and Algorithm 3 are analyzed as follows. For Algorithm 3, each of the Lines 1 to 4 runs at most N times. Given that Algorithm 1's time complexity is $O(S^2M^2NL)$, the complexity of Algorithm 3 is $O(S^2M^2N^2L)$. For Algorithm 2, the `while` loop in Line 2 runs T times, and the complexities of Lines 4 and 5 are $O(S^2M^2NL)$ and $O(S^2M^2N^2L)$, respectively. In summary, Algorithm 2's time complexity is $O(TS^2M^2N^2L)$.

Theorem 4. *The online algorithm gives a $\frac{\kappa}{(\kappa-1)\epsilon}\cdot(\frac{\sigma}{\sigma-1})^2$-competitive solution to the social welfare maximization problem, i.e., the social welfare obtained by Algorithm 2 is at least $\frac{(\kappa-1)\epsilon}{\kappa}\cdot(\frac{\sigma-1}{\sigma})^2$ times the offline optimal social welfare, where $\epsilon=\min_{t\in\mathcal{T}}\frac{\min_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t)}W^t_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))}{\max_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t)}W^t_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))}$, and $\frac{\sigma}{\sigma-1}$ is the approximation ratio of Algorithm 1.*

Proof. In Algorithm 2, Line 6 guarantees that the switching cost at time slot t is at most $\frac{1}{\kappa}$ times the non-switching welfare incurred within time frame $[\hat{t},t-1]$, where $\hat{t}$ is the last time slot of cloudlet activation state switching before t. In the worst case, over the whole time frame $\mathcal{T}$, the cloudlet activation status $\boldsymbol{x}$ switches in each time frame, we have $\sum_{\tau=1}^{T}C^{\tau}_{SC}\leqslant\frac{1}{\kappa}\sum_{\tau=1}^{T}W^{\tau}_{-SC}$. Hence we have $\sum_{\tau=1}^{T}W^{\tau}=\sum_{\tau=1}^{T}W^{\tau}_{-SC}-\sum_{\tau=1}^{T}C^{\tau}_{SC}\geqslant(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}W^{\tau}_{-SC}$. Let $\bar{W}^t$ denote the offline optimal social welfare at time slot t, and ϵ be the minimum ratio of the smallest non-switching welfare to the largest non-switching welfare at each time slot, i.e., $\epsilon=\min_{t\in\mathcal{T}}\frac{\min_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t)}W^t_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))}{\max_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t)}W^t_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))}$. Note that W^t_{-SC} given by Algorithm 1 is at least $\frac{\sigma-1}{\sigma}$ times the optimal non-switching welfare W^{*t}_{-SC}. Hence $\epsilon^*=\min_{t\in\mathcal{T}}\frac{\min_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t)}W^{*t}_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))}{\max_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t)}W^{*t}_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))}\leqslant\frac{\sigma}{\sigma-1}\epsilon$. And we have

$$\sum_{\tau=1}^{T}W^{\tau}\geqslant(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}W^{\tau}_{-SC}\geqslant(\frac{\sigma-1}{\sigma})^2\epsilon(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}\bar{W}^{\tau}_{-SC}$$
$$\geqslant(\frac{\sigma-1}{\sigma})^2\epsilon(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}\bar{W}^{\tau}. \qquad \square$$

As obtaining the exact maximum and minimum non-switching welfare is computationally prohibitive, here we propose a rough estimate of the competitive ratio. According to the formulation of non-switching welfare, $W^t_{-SC}(\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{z}(t),\boldsymbol{q}(t),u(t))$ is bounded by $[\sum_{i,j}\min_{k,l}\{a^{kl}_{ij}(t)\}-p(P_{max}(t)-P_{EDR}(t))-\sum_{i,j}c_{ij}(t)(\theta_{ij}(t)+\varrho_{ij}(t)),\sum_{i,j}\max_{k,l}\{a^{kl}_{ij}(t)\}]$, where $P_{max}(t)$ is the power consumption of the case that all cloudlets are active and the workload is allocated to those most power-consuming cloudlets. And hence the competitive ratio is bounded by $[1,\frac{\kappa\sigma^2}{(\kappa-1)(\sigma-1)^2}\min_{t\in\mathcal{T}}\{(\sum_{i,j}\min_{k,l}\{a^{kl}_{ij}(t)\}-p(P_{max}(t)-P_{EDR}(t))-\sum_{i,j}c_{ij}(t)(\theta_{ij}(t)+\varrho_{ij}(t)))/\sum_{i,j}\max_{k,l}\{a^{kl}_{ij}(t)\}\}]$.

From the expression of the competitive ratio, we can find that the approximation ratio $\frac{\sigma}{\sigma-1}$ affects the competitive ratio most. That is, if we can obtain the non-switching welfare closer to the optimum at each time slot, EdgeDR will achieve better long-term performance as well. And the competitive ratio is also affected by the difference of the non-switching welfare that can be achieved at one time slot (i.e., ϵ). That is, if there is a large variation of the non-switching welfare from one cloudlet activation status to another, the online algorithm will face uncertainty in the long-term social welfare maximization problem, reducing its theoretical efficiency. And the competitive ratio is tight to our proposed algorithm since we only consider the worst and extreme case without ignoring any term in the proof. As it is still an open problem that what is the best theoretical competitive ratio for the online problem with switching cost which we focus on, we leave designing an online algorithm with the best-possible performance guarantee to future work. In the following section, we will show (cf. Fig. 13) that our proposed algorithm can achieve good empirical competitive ratios in realistic scenarios, and the efficiency of Algorithm 1 is the key to the competitive ratio improvement.

5 Dynamic Payment Mechanism

In Sec. 3, we propose the one-round EdgeDR auction mechanism ensuring both social welfare maximization and truthful bid. However, as the social welfare considers the whole benefit of all EdgeDR participants, i.e., both the cloudlet operator and the service provider, there is no guarantee of the utility of each participant. As discussed in the following Theorem 5, the cloudlet operator, i.e., the auctioneer of the EdgeDR auction mechanism, can adjust the payment from the winner's cost of reducing QoE to her "marginal contribution", ensuring individual rationality and truthfulness at the same time. In reality, for the operator, it is the most cost efficient to reduce the payment as much as possible and maximize her own profit. However, the service provider is unwilling to participate in EdgeDR without any benefit. Instead of the cost caused by attending EdgeDR, they may have their own private expectation on utility. When the payment does not meet the winner's expectation, she will drop out of the EdgeDR programs, which may hurt the long-term benefit of the operator.

In this section, we first analyze the range of payment in EdgeDR auction. Then we propose a dynamic payment mechanism, for which the operator dynamically adjusts the payment to winners and recruits the dropped bidders to improve her own benefit.

Theorem 5. *The payment $\{r_{ij}\}$ to winner ij can be tuned within range $\{c_{ij}(z_{ij}+q_{ij}),c_{ij}(z_{ij}+q_{ij})+(\rho_{ij}-\rho_{i^-j^-})\}$ while ensuring the individual rationality and the truthfulness at the same time.*

Proof. Let $r_{ij}=c_{ij}(z_{ij}+q_{ij})+r(c_{ij})$ be the payment to winner ij. According to Lemma 1, the winners should be paid with critical payment. Assume service provider j in AP i wins, her workload is distributed to cloudlet k with resource allocation l. If she reports a new unit-delay penalty c'_{ij}, she will also win and her workload will be also distributed to k with resource l if $c'_{ij}(z_{ij}+q_{ij})\leqslant r_{ij}$.

According to the definition of ρ_{ij}^{kl} and the winning condition of Algorithm 1, we have

$$\begin{aligned} {\rho'}_{ij}^{kl} &= a_{ij}^{kl} - (\lambda_{ij}^{l}\mu_k + c'_{ij}(d_{ijk} + \omega_{ij}^{l}) + \beta_k\lambda_{ij}^{l}p) \geqslant \rho_{i^-j^-}^{k^-l^-} \\ \Rightarrow \quad & c'_{ij}(z_{ij} + q_{ij}) \leqslant a_{ij}^{kl} - (\lambda_{ij}^{l}\mu_k + \beta_k\lambda_{ij}^{l}p) - \rho_{i^-j^-}^{k^-l^-}. \end{aligned} \tag{5}$$

To ensure the payment is critical, we have

$$c'_{ij}(z_{ij} + q_{ij}) \leqslant r_{ij} = c_{ij}(z_{ij} + q_{ij}) + r(c_{ij}). \tag{6}$$

Note that as long as the payment is critical, (5) must be satisfied. That is,

$$\begin{aligned} c'_{ij}(z_{ij} + q_{ij}) &\leqslant c_{ij}(z_{ij} + q_{ij}) + r(c_{ij}) \\ &\leqslant a_{ij}^{kl} - (\lambda_{ij}^{l}\mu_k + \beta_k\lambda_{ij}^{l}p) - \rho_{i^-j^-k^-l^-} \\ \Rightarrow \quad r(c_{ij}) &\leqslant a_{ij}^{kl} - (\lambda_{ij}^{l}\mu_k + c_{ij}(z_{ij} + q_{ij}) + \beta_k\lambda_{ij}^{l}p) \\ &\quad - \rho_{i^-j^-}^{k^-l^-} \\ &\leqslant \rho_{ij}^{kl} - \rho_{i^-j^-}^{k^-l^-}. \end{aligned} \tag{7}$$

Further, to ensure the individual rationality, the utility of winner ij is no less than zero, i.e., $r_{ij} \geqslant c_{ij}(z_{ij} + q_{ij})$. Above all, we can conclude that the range of payment $\{r_{ij}\}$ is $\{c_{ij}(z_{ij} + q_{ij}), c_{ij}(z_{ij} + q_{ij}) + (\rho_{ij} - \rho_{i^-j^-})\}$. □

Because of the restriction of both truthfulness and individual rationality, the operator has to set the payment $r_{ij} = c_{ij}(z_{ij} + q_{ij}) + \Delta(\rho_{ij} - \rho_{i^-j^-})$, where $\Delta \in [0, 1]$. To make the highest own profit, the operator would like to pay the minimum value to winners. However, in reality, the satisfaction of volunteer service providers may be decreased for receiving zero profit and they may drop out of the EdgeDR auction when the benefit cannot meet their expectation. In this case, although the operator achieves high short-term profit, the bidders' dropout will reduce the potential of workload management in edge demand response events, which further hurts the long-term benefit of the operator.

To improve the profit of the operator meanwhile ensure the performance of EdgeDR auction, we propose a dynamic payment mechanism for the operator to adjust the payment to winners according to the number of volunteers and the urgency of edge demand response events. Fig. 4 illustrates the flowchart of the dynamic payment mechanism.

To reduce the cost of paying winners and prevent a large amount of bidders from dropping out from EdgeDR auction, the operator stepwise adjusts the payment at each time slot. After receiving the workload and soliciting bids, in general, the operator reduces Δ by Δ_{step}. However, when the number of dropped bidders at the last time slot exceeds the threshold, the operator needs to multiply Δ by $\hat{\Delta} > 1$ to incentivize dropped bidders to rejoin EdgeDR again. And when the requirement of computing resources or the number of dropped bidders bursts, the operator needs to increase Δ to protect the edge system in EdgeDR. Specifically, if the resource requirement exceeds the threshold of computing demand response, the operator has to increase Δ to the maximum value to avoid remaining volunteers dropping out and recruit the dropped bidders. Otherwise, it will run out of computing resources. Similarly, if the number of remaining volunteers is less than the threshold, the potential of workload management in edge demand response events will be limited and hurt the long-term benefit of EdgeDR

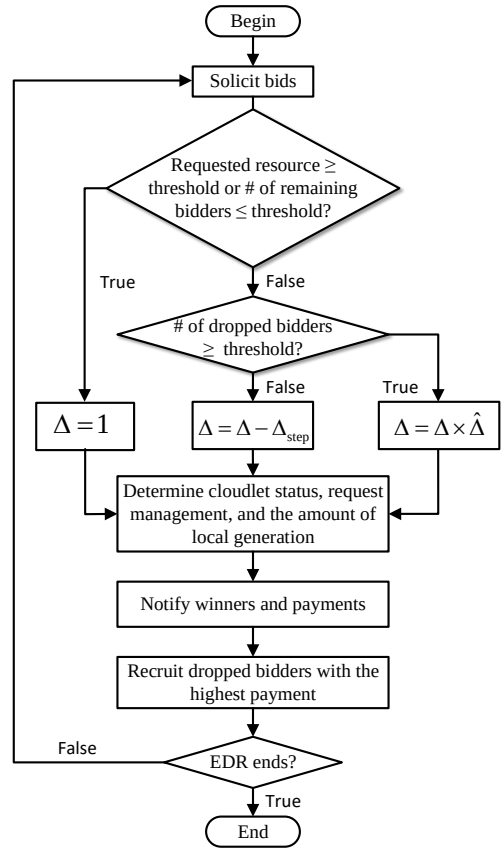


Fig. 4. The flowchart of payment adaptation.

in peak periods. After notifying winners and payments, the operator will reveal the highest payment in this round to the dropped bidders. And the dropped bidder may choose to rejoin EDR auction if it exceeds her private expectation.

6 Experimental Evaluation

6.1 Experiment Setup

We simulate a cloudlet operator owning 20 distributed cloudlets and serving 40 service providers. We use London's 20 underground stations with heavy passenger traffic to simulate the locations of distributed cloudlets, and the propagation delay between cloudlets is estimated by the geographical distance between two stations [15]. Servers in all cloudlets are homogeneous and each server can process 25 requests at one time slot. To evaluate the performance of EdgeDR in computing demand response, we divide the sum of the entire edge system's peak workload by the number of cloudlets, and randomly scale it from $0.8\times$ to $1.2\times$ to generate cloudlets' capacity. The cloudlet's PUE ranges from 1.3 to 2 randomly, and the idle and peak power of a server is 100 W and 300 W, respectively. The diesel price for local generation is set to 0.8 \$/kWh [33], [34].

We utilize the dynamic passenger numbers at a station to represent the total amount of workload submitted from that station (i.e., AP), and the amount of request for each service is assigned randomly. The reliability of workload management depends on the amount of workload. And the resource price of each cloudlet is generated referring to

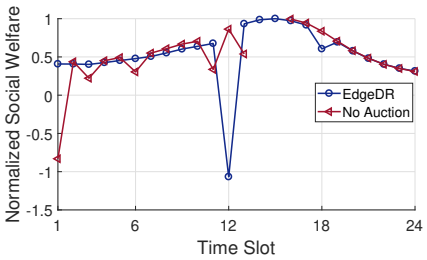


Fig. 5. Comparison of the normalized social welfare with EdgeDR and no-auction mechanism.

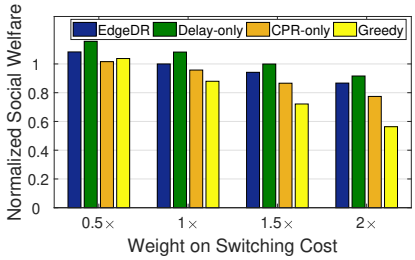


Fig. 6. Comparison of the normalized social welfare with different mechanisms as the weight on switching cost increases.

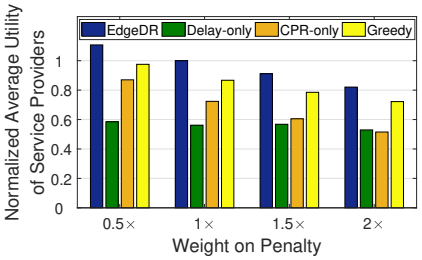


Fig. 7. Comparison of the service provider's normalized average utility as the weight on per-unit penalty increases.

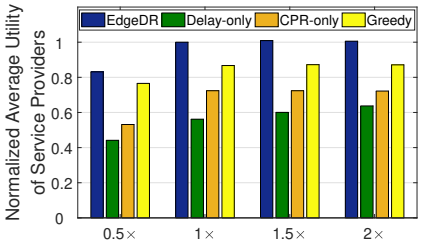


Fig. 8. Comparison of the service provider's normalized average utility as delay degradation and workload compression tolerance increases.

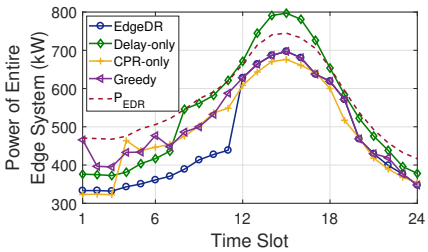


Fig. 9. Comparison of the entire edge system power within the edge demand response event with different mechanisms.

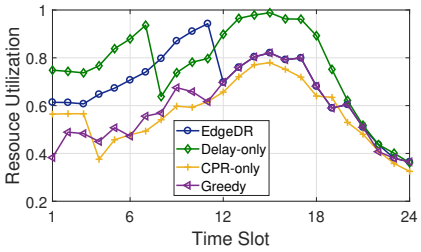


Fig. 10. Comparison of the entire edge system resource utilization within the edge demand response event with different EDR mechanisms.

Sec. 3.1. The delay tolerance of request is set according to the mean propagation delay between cloudlets. And the delay degradation tolerance submitted by the service provider follows a uniform distribution between 60% delay tolerance to 100% delay tolerance. We assume the allocated resource can be compressed to at most 20%, and the submitted throughput degradation tolerance follows a discrete uniform distribution from 0% (i.e., no resource compression), 20%, 40%, 60%, to 80% (i.e., maximum resource compression ratio). We vary the weight of the switching cost in the online framework, the threshold, and the per-unit delay penalty to obtain a spectrum of results, hence we do not give the concrete metric here.

We set the length of one time slot to 15 minutes and the total length of the EDR event is 24 time slots, according to a real 6-hour EDR event from the service region of NYISO on Aug. 28, 2018 [35], same with the setup used in the recent EDR paper [25]. The EDR signal requires the edge system to reduce 25% of the edge system's peak IT power consumption, which is reported as a reasonable setting in EDR events without significant impact on the participant's operation [7], [36].

Because there is no existing work on performing both power emergency demand response and computing demand response at the edge, in the evaluations, we design reasonable alternative approaches to represent the common settings and better show the performance of our proposed mechanism: 1) No auction, for which the operator just utilizes local generation and workload management within tolerance to meet the power demand response and computing demand response, and does not motivate service providers to further reduce QoS (i.e., propagation delay and throughput). 2) The online greedy and auction, for which the operator directly varies the cloudlet status when the near-optimal social welfare is larger than that in the previous status. 3) Delay-only auction, for which the operator only motivates service providers to increase delay degradation tolerance. 4) Compression-only (CPR-only) auction, for which the operator only motivates service providers to increase the throughput degradation tolerance (i.e., compress required computing resources). We also compare EdgeDR with the optimum algorithm where the one-shot problem is solved exactly by CVX [37] with Gurobi [38]: 1) The online optimum, for which the one-shot problem is optimum. 2) The offline optimum, for which we search all of possible cloudlet activation statuses at each time slot to obtain the optimum solution. As obtaining the optimum is time-consuming and not scalable, we only conduct the optimum framework simulation in small-scale scenarios. For simplicity, the metrics of social welfare, utility, and cost are normalized with respect to it achieved by EdgeDR.

6.2 Evaluation Results

6.2.1 EdgeDR

Fig. 5 shows the normalized social welfare obtained by EdgeDR and no-auction framework at each time slot. We observe that with the increasing demand of edge services, at time slots 14 and 15, there is no social welfare achieved by no-auction approach because it fails to provide reliable IaaS service, i.e., some requests cannot be processed. Specifically, some cloudlets run out of computing resources, since requests with tight delay tolerance and high throughput requirement burst at peak hours but cannot be distributed to remote cloudlets or be compressed to save scarce resources without the help of EdgeDR programs. And because of the ineffectiveness, in the following evaluation, we do not use

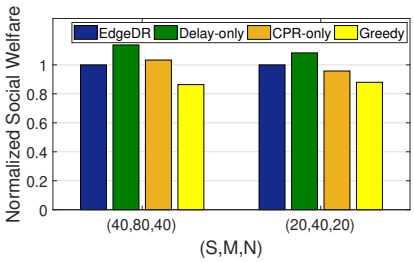


Fig. 11. Comparison of the normalized social welfare with different mechanisms in different scales.

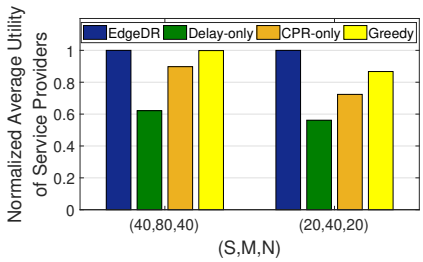


Fig. 12. Comparison of the service provider's normalized average utility with different mechanisms in different scales.

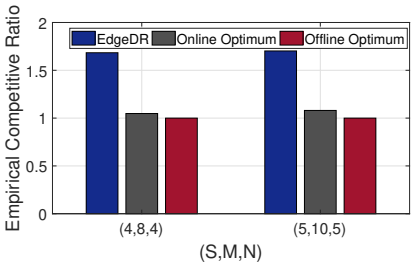


Fig. 13. Comparison of the normalized social welfare with online optimum and offline optimum in small-scale simulation.

the no-auction approach. For EdgeDR, there is a sharp drop of social welfare at time slot 12 since the cloudlet operator decides to activate multiple cloudlets for providing higher processing capacity, which incurs noticeable switching cost and hence the reduction of social welfare.

Fig. 6 compares the normalized social welfare obtained by EdgeDR and that obtained by three approaches, as the weight on the switching cost increases. Since the number of candidate cloudlets for workload distribution is higher than the number of resource allocation level, delay-only approach is more flexible than the CPR-only approach and it achieves higher social welfare. EdgeDR, which utilizes both delay degradation and throughput degradation to achieve edge demand response, has an intermediate performance in social welfare compared to Delay-only and CPR-only approaches. Moreover, as EdgeDR avoids aggressive cloudlet statuses switching effectively, compared to the greedy framework, it achieves great effectiveness in improving long-term social welfare, especially when the switching cost increases.

Fig. 7 and Fig. 8 present the normalized average utility of service providers, i.e., the winner's payment minus the penalty caused by quality reduction, as the per-unit penalty cost $c_{ij}(t)$ increases and as the tolerance threshold increases, respectively. The results in Fig. 7 show that the increasing per-unit penalty leads to a decline in service provider's utility since increasing penalty hurts the social utility of workload management ρ_{ij}. And compared to delay-only framework and CPR-only framework, the service provider in EdgeDR has the highest utility. This is because EdgeDR achieves edge demand response by increasing delay degradation tolerance and tolerance degradation tolerance together, which makes EdgeDR more efficient in workload scheduling than delay-only and CPR-only frameworks and hence increases the utility ρ_{ij}.

Fig. 9 and Fig. 10 show the power and resource utilization of the entire edge system within the edge demand response event. We find that in the computation low time, EdgeDR and other approaches all perform well in power shedding and resource preserving. While in peak periods, leveraging both dynamic workload distribution and resource allocation, EdgeDR has the best power efficiency and better resource preserving. It is worth noting that at time slot 12, there are a noticeable increase in power consumption and a sharp drop of resource utilization by EdgeDR, as shown in Fig. 9 and Fig. 10, respectively. The reason corresponds to the discussion for Fig. 5 that the operator activates multiple cloudlets for higher processing capacity at time slot 12, providing more available resources for the edge service while increasing the (idle) power consumption. Compared to delay-only framework, EdgeDR achieves local generation power free, indicating that the participation of resource allocation in the edge demand response program ensures the "greenness". EdgeDR also guarantees the reasonable resource utilization of the whole edge system to avoid the under-provisioning issue in peak periods.

Above all, although delay-only framework achieves the highest social welfare, it has limited potential for further improving the utility of service providers and the power efficiency. And it is also hard to ensure reliable resource utilization for computing demand response.

To evaluate the effect of the cloudlet network scale on the performance of EdgeDR, Fig. 11 and Fig. 12 compare the normalized social welfare and the normalized average utility of service providers obtained by EdgeDR to three alternative approaches in different scales, respectively. We can find that EdgeDR achieves similar performance compared to these approaches where it efficiently balances the tradeoff between the social welfare and utility of service providers.

We also compare the social welfare obtained by EdgeDR to the online optimum and offline optimum, in a small-scale scenario. The results in Fig. 13 show the great empirical competitive ratio of EdgeDR which is no more than 1.71. We also find that compared with the online algorithm, the efficiency of the workload allocation, workload compression, and winner determination framework (i.e., the approximation ratio of Algorithm 1, and the empirical average value in the scale $(S, M, N) = (5, 10, 5)$ is about 1.72) is the key to the competitive ratio improvement.

We finally measure the practical computation efficiency of EdgeDR. According to our evaluations conducted on a commercial laptop with an Intel® Core® i7 CPU and 64GB RAM, the running time of our algorithm for the one-shot problem in the scale $(S, M, N) = (20, 40, 20)$ at a single time slot is only about 3 seconds. For the problem in a larger cloudlet network's scale, the computation time can be reduced by leveraging parallel computing. In contrast, note that in a small-scale scenario as discussed in Fig. 13, it takes more than 18 minutes and 1 hour for the online optimum and offline optimum to make a decision at one time slot, respectively, which is unpractical for the EDR event with a 15-minute time slot.

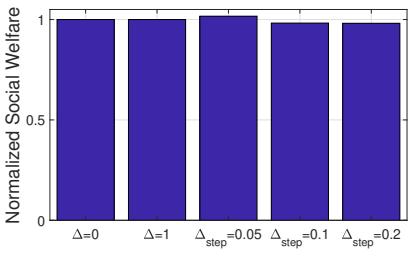


Fig. 14. Comparison of the normalized social welfare with different payment settings.

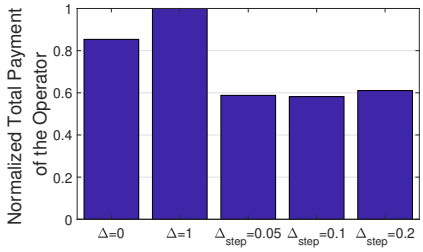


Fig. 15. Comparison of the normalized payment of the operator with different payment settings.

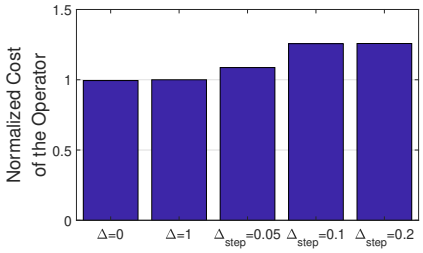


Fig. 16. Comparison of the normalized cost of the operator with different payment settings.

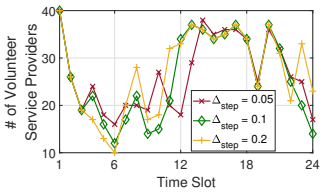


Fig. 17. Comparison of the number of volunteer service providers in each time slot with different payment settings.

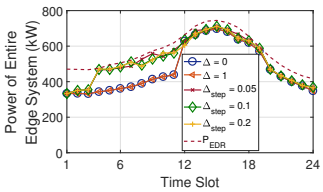


Fig. 18. Comparison of the entire edge system power within edge demand response event with different payment settings.

6.2.2 *Dynamic Payment Mechanism*

In this section, we evaluate the effect of different payment settings on the performance of the payment mechanism: 1) $\Delta = 0$, for which the service provider has no extra expectation on the received payment, and the payment always equals to her cost, i.e., $r_{ij} = c_{ij}(z_{ij}+q_{ij})$. 2) $\Delta = 1$, for which the service provider has no extra expectation on the received payment, and the operator always pay the winner with the highest value, i.e., $r_{ij} = c_{ij}(z_{ij}+q_{ij})+\rho_{ij}-\rho_{i^-j^-}$. 3) $\Delta_{step} = 0.05$, $\Delta_{step} = 0.1$, and $\Delta_{step} = 0.2$, for which the service provider has a private expectation on the received payment, and the operator adjusts the payment with $\Delta_{step} = 0.05$, $\Delta_{step} = 0.1$, and $\Delta_{step} = 0.2$, respectively.

Fig. 14, Fig. 15, and Fig. 16 compare the normalized social welfare, the normalized cost of the operator, and the normalized payment obtained by EdgeDR with different payment settings, respectively. Note that the payments of the operator and service providers cancel one another in social welfare, the payment value does not affect the total social welfare. Thus whatever which payment setting the operator uses, the social welfare are almost the same. Fig. 15 shows that compared to $\Delta = 1$, the operator saves about 15% of cost in $\Delta = 0$ as she does not give any extra benefit to the winner.

Further, compared to the ideal scenario where all service providers stay in EdgeDR all the time, the operator can save more cost of payment when the service provider has extra expectation on utility, and drops out of and returns to the auction freely. This is because the low number of bidders reduces the number of winners and the cost of paying winners decreases accordingly. However, the dropout of bidders hurts the potential of workload management in EdgeDR programs, and the operator needs to activate more cloudlets to ensure the reliability. As illustrated in Fig. 16, the total cost of the operator in this practical scenario is higher because of the increase of switching cost and maintenance cost. Fig. 16 also shows that the lower Δ_{step} reduces the total cost of the operator significantly. This is because aggressively reducing Δ may let large amount of volunteers drop out of EdgeDR, as illustrated in Fig. 17, and hurt the long-term benefit of the operator.

Fig. 18 shows the power load of the entire edge system with different payment setting. We find that because of the bidder dropout, the flexibility of workload management decreases. And the workload without any throughput degradation increases the power consumption and the operator may have to activate more cloudlets to provide sufficient resources. The case of bidder dropout consumes more 9.74%-10.54% of power load compared to the case where the service provider has no extra expectation on utility. However, EdgeDR still ensures the "greenness" and avoids the usage of local generation by dynamic payment setting and bidder recruiting.

7 Related Work

There exist many studies on improving power efficiency and achieving power cost saving in datacenters, such as workload management [39], [40], [41], [42], datacenter heat harvesting [43], and datacenter green computing [44], [45], [46]. In this section, we summarize very closely-related existing work only. Ren et al. [8] is among the first to study the colocation data center (referred to as "colo" henceforth) demand response scenario, where they propose a simple reverse auction to meet the energy reduction requirement. Zhang et al. [5] also design a reverse auction, introducing local generators to help meet the energy reduction target and VCG-based mechanisms to guarantee truthful bidding. Chen et al. [11] design a pricing mechanism based on supply function biddings to extract load reductions from tenants in demand response periods. Sun et al. [6], [7] investigate two cases of demand response in geo-distributed colos with deferrable batch jobs, and propose online multi-round auctions while determining when to execute each job. Tran et al. [10] and Islam et al. [9] may be among the few that do not use auctions but reward-based mechanisms to issue rewards in exchange for tenants' energy reductions. Song et al. [25] recently present an edge scheduling algorithm for power demand response, considering edge tasks which can be interrupted and executed across time slots.

Our research in this paper differs from all the above existing work in multiple aspects. First, we do "online"

reverse auctions for multiple demand response frames. All existing research, except [6], [7], [9], [25], uses a one-round, static auction which does not explore the temporal connection between different demand response frames. Second, we distribute tenants' workload explicitly and adjust service quality based on the operator's needs. Previous mechanisms, except [6], [7], [25], do not manage tenants' workload and are limited by tenants' fixed offers. Third, we are the first to explore the lever of switching on/off entire cloudlets, while considering the switching cost over time, to compensate for the energy reduction requirement. This is particularly a feature that may be possible for cloudlets only. Existing research mainly targets large-scale clouds and data centers [47] that cannot be usually switched off entirely.

We find some immediately related studies from the game theory community as follows. Deng et al. [48] design online combinatorial auction mechanisms which simultaneously ensure the universal truthfulness for submodular combinatorial valuations and maximize the long-term social welfare. Ban et al. [49] focus on the second-price sequential auction where items are sold sequentially and bidder valuations are time-varying, and propose an option-value bidding strategy to achieve the subgame-perfect equilibrium. Corazzini et al. [50] study the sequential reverse auction where each bidder's capacity is unknown to the auctioneer. Also, some studies [51], [52], [53] focus on a particular case where the sequential auction is with only two items or two bidders. Drutsa [54] presents an online learning algorithm for the repeated second-price auction with reserve where strategic bidders repeatedly participate in auctions and the seller needs to determine reserve prices in each round based on the previous information for maximizing her long-term revenue. Drutsa also proposes a learning-based algorithm for revenue maximization in the repeated contextual posted-price auction [55]. For the real-time bidding scenario of online display advertising, Lu et al. [56] train a real-time bidding strategy by using the emerging reinforcement learning technique integrating with the sequential information extraction and state aggregation schemes. The above online mechanisms, however, are not applicable to the problem considering the switching cost over time as what we study in our work.

Some existing studies on bidders retaining and/or recruiting in participatory sensing. The service provider here needs to incentivize enough number of active sensors and collects samples sufficient for ensuring the quality of service. Lee et al. [57] design a reverse auction based dynamic price (RADP) mechanism to motivate mobile users to provide their sensing data for participatory sensing. As the users with lower bidding price always win and make other bidders drop out, the price competition will be weakened, which causes an incentive cost explosion and hurts the profit obtained by the service provider. RADP introduces a virtual participation credit for those bidders with high bidding price to increase the winning probability and hence retain enough number of mobile users. And RADP also reveals the highest payment to those dropped users to recruit bidders when the incentive cost explosion happens. But although RADP maintains sufficient bidders and improves the profit of the service providers, it fails to provide any performance guarantee such as the truthfulness of reverse auction. Gao et al. [58] propose a Lyapunov-based long-term participatory incentive mechanism. They ensure the winning probability of each mobile user to retain all bidders in their participatory sensing framework, and utilize the VCG auction mechanism to guarantee the truthfulness of participators.

This work significantly extends the preliminary work [59]. By extending the scenario from power EDR programs to compound edge demand response programs, this work presents a more general online market mechanism for cloudlets to achieve reliability and stability of the power grid and the edge clouds. Instead of only considering the overall social welfare in peak periods, in this work, we further investigate a more practical scenario and propose a dynamic payment mechanism to balance the short-term profit and long-term benefit of the operator.

8 Conclusion

We study the power emergency demand response and computing demand response for distributed cloudlets. In our mechanism, we design a series of procurement auctions to address the operator-tenant split incentives. And we focus on enabling the operator to directly distribute tenants' workload across cloudlets and schedule computing resources in order to gain more flexibility in the procurement of tenants' bids and to adapt to the changing market conditions. We are also the first to propose to let the operator switch on/off entire cloudlets to improve the stability of the grid and/or ensure the reliability of the edge while dynamically striking the balance between the demand response benefit and the incurred switching cost. We propose an online algorithm, which adopts a polynomial-time approximation algorithm in each auction, with provable performance guarantees and validated practical superiority compared to existing methods. We also investigate a more practical case where the bidder may drop out of EdgeDR if the utility is lower than her expectation. We present a dynamic payment mechanism for the operator to reduce her cost while ensuring individual rationality, truthfulness, as well as the long-term benefit of EdgeDR.

Acknowledgments

This work was supported in part by NSFC under Grant 61722206 and 61761136014 (and 392046569 for DFG) and 61520106005, in part by the National Key Research & Development (R&D) Plan under grant 2017YFB1001703, in part by the Fundamental Research Funds for the Central Universities under Grant 2017KFKJXX009 and 3004210116, in part by the National Program for Support of Top-notch Young Professionals in National Program for Special Support of Eminent Professionals. The work of L. Jiao is supported in part by the Ripple Faculty Fellowship.

References

[1] M. Satyanarayanan, "The emergence of edge computing," *IEEE Computer*, vol. 50, no. 1, pp. 30–39, 2017.
[2] W. Shi and S. Dustdar, "The promise of edge computing," *IEEE Computer*, vol. 49, no. 5, pp. 78–81, 2016.
[3] M. Trifiro and J. Smith. State of the Edge Report 2020. [Online]. Available: https://stateoftheedge.com/reports/state-of-the-edge-2020/

[4] I. Morris. China Unicom to Build 'Thousands' of Edge DCs. [Online]. Available: https://www.lightreading.com/data-center/data-center-infrastructure/china-unicom-to-build-thousands-of-edge-dcs/d/d-id/737151

[5] L. Zhang, S. Ren, C. Wu, and Z. Li, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *IEEE INFOCOM*, 2015.

[6] Q. Sun, S. Ren, C. Wu, and Z. Li, "An online incentive mechanism for emergency demand response in geo-distributed colocation data centers," in *ACM e-Energy*, 2016.

[7] Q. Sun, C. Wu, Z. Li, and S. Ren, "Colocation demand response: Joint online mechanisms for individual utility and social welfare maximization," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3978–3992, 2016.

[8] S. Ren and M. A. Islam, "Colocation demand response: Why do I turn off my servers?" in *USENIX ICAC*, 2014.

[9] M. A. Islam, H. Mahmud, S. Ren, and X. Wang, "Paying to save: Reducing cost of colocation data center via rewards," in *IEEE HPCA*, 2015.

[10] N. H. Tran, T. Z. Oo, S. Ren, Z. Han, E.-N. Huh, and C. S. Hong, "Reward-to-reduce: An incentive mechanism for economic demand response of colocation datacenters," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3941–3953, 2016.

[11] N. Chen, X. Ren, S. Ren, and A. Wierman, "Greening multi-tenant data center demand response," *Performance Evaluation*, vol. 91, pp. 229–254, 2015.

[12] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet enhanced fiber-wireless access networks for mobile-edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3601–3618, 2017.

[13] G. Iyengar and A. Kumar, "Optimal procurement mechanisms for divisible goods with capacitated suppliers," *Review of Economic Design*, vol. 12, no. 2, pp. 129–154, 2008.

[14] N. Li, L. Chen, and M. A. Dahleh, "Demand response using linear supply function bidding," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1827–1838, 2015.

[15] L. Jiao, L. Pu, L. Wang, X. Lin, and J. Li, "Multiple granularity online control of cloudlet networks for edge computing," in *IEEE SECON*, 2018.

[16] M. Ganeshalingam, A. Shehabi, and L.-B. Desroches, "Shining a Light on Small Data Centers in the U.S." *Lawrence Berkeley National Laboratory Report*, 2017.

[17] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in *IEEE INFOCOM*, 2018.

[18] L. Jiao, A. M. Tulino, J. Llorca, Y. Jin, and A. Sala, "Smoothed online resource allocation in multi-tier distributed cloud networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2556–2570, 2017.

[19] P. Jin, X. Fei, Q. Zhang, F. Liu, and B. Li, "Latency-aware vnf chain deployment with efficient resource reuse at network edge," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 267–276.

[20] Q. Zhang, F. Liu, and C. Zeng, "Online adaptive interference-aware vnf deployment and migration for 5g network slice," *IEEE/ACM Transactions on Networking*, 2021.

[21] Q. Chen, Z. Zheng, C. Hu, D. Wang, and F. Liu, "On-edge multi-task transfer learning: Model and practice with data-driven task allocation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1357–1371, 2020.

[22] B. Gao, Z. Zhou, F. Liu, F. Xu, and B. Li, "An online framework for joint network selection and service placement in mobile edge computing," *IEEE Transactions on Mobile Computing*, 2021.

[23] PJM. Demand Response and Why It's Important. [Online]. Available: https://www.pjm.com/~/media/markets-ops/dsr/end-use-customer-fact-sheet.ashx

[24] ——. Current Demand Response opportunities. [Online]. Available: https://www.pjm.com/-/media/committees-groups/task-forces/sodrstf/20171213/20171213-dr-summer-opportunities.ashx

[25] Z. Song, R. Zhou, S. Zhao, S. Qin, J. C. Lui, and Z. Li, "Edge emergency demand response control via scheduling in cloudlet cluster," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 394–399.

[26] N. G. Mankiw, *Principles of economics*. Cengage Learning, 2020.

[27] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1737–1749, 2017.

[28] I. Alaya, C. Solnon, and K. Ghédira, "Ant algorithm for the multi-dimensional knapsack problem," in *International Conference on Bioinspired Optimization Methods and their Applications*, 2004.

[29] R. Cole, N. Devanur, V. Gkatzelis, K. Jain, T. Mai, V. V. Vazirani, and S. Yazdanbod, "Convex program duality, fisher markets, and nash social welfare," in *ACM EC*, 2017.

[30] F. Nielsen, "Legendre transformation and information geometry," 2010.

[31] R. B. Myerson, "Optimal auction design," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 58–73, 1981.

[32] A. Archer and É. Tardos, "Truthful mechanisms for one-parameter agents," in *IEEE FOCS*, 2001.

[33] Calculator of Diesel engine power. [Online]. Available: https://power-calculation.com/generator-diesel-energy-calculator-genset.php#consumptionvalues

[34] Price of diesel fuel in London. [Online]. Available: https://www.statista.com/statistics/299547/price-of-diesel-in-london/

[35] NYISO Summer 2018 Hot weather operations. [Online]. Available: http://www.nysrc.org

[36] G. Ghatikar, "Demand response opportunities and enabling technologies for data centers: Findings from field studies," 2012.

[37] CVX. [Online]. Available: http://cvxr.com/cvx/

[38] Gurobi. [Online]. Available: http://www.gurobi.com

[39] Z. Zhou, F. Liu, B. Li, B. Li, H. Jin, R. Zou, and Z. Liu, "Fuel cell generation in geo-distributed cloud services: A quantitative study," in *IEEE ICDCS*, 2014, pp. 52–61.

[40] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2648–2658, 2014.

[41] S.-K. Jo, L. Wang, J. Kangasharju, and M. Mühlhäuser, "Green named data networking using renewable energy," in *ACM e-Energy*, 2018, pp. 414–416.

[42] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, and H. Jin, "Carbon-aware online control of geo-distributed cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2506–2519, 2015.

[43] S. Chen, Z. Zhou, F. Liu, Z. Li, and S. Ren, "Cloudheat: An efficient online market mechanism for datacenter heat harvesting," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 3, no. 3, p. 11, 2018.

[44] W. Deng, F. Liu, H. Jin, B. Li, and D. Li, "Harnessing renewable energy in cloud datacenters: opportunities and challenges," *IEEE Network*, vol. 28, no. 1, pp. 48–55, 2014.

[45] Z. Zhou, F. Liu, and Z. Li, "Bilateral electricity trade between smart grids and green datacenters: Pricing models and performance evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3993–4007, 2016.

[46] Z. Zhou, F. Liu, Z. Li, and H. Jin, "When smart grid meets geo-distributed cloud: An auction approach to datacenter demand response," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2650–2658.

[47] Z. Zhou, F. Liu, S. Chen, and Z. Li, "A truthful and efficient incentive mechanism for demand response in green datacenters," *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[48] Y. Deng, D. Panigrahi, and H. Zhang, "Online combinatorial auctions," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2021, pp. 1131–1149.

[49] A. Ban and R. Lavi, "Option values in sequential auctions with time-varying valuations," *International Journal of Game Theory*, pp. 1–30, 2020.

[50] L. Corazzini, S. Galavotti, and P. Valbonesi, "An experimental study on sequential auctions with privately known capacities," *Games and Economic Behavior*, vol. 117, pp. 289–315, 2019.

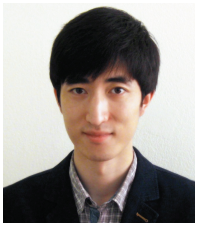
[51] H. Āzacis, "Information disclosure by a seller in sequential first-price auctions," *International Journal of Game Theory*, vol. 49, no. 2, pp. 411–444, 2020.

[52] M. Ş. Ahunbay and A. Vetta, "The price of anarchy of two-buyer sequential multiunit auctions," in *International Conference on Web and Internet Economics*. Springer, 2020, pp. 147–161.

[53] G. Ghosh and H. Liu, "Sequential second-price auctions with private budgets," *Games and Economic Behavior*, vol. 113, pp. 611–632, 2019.

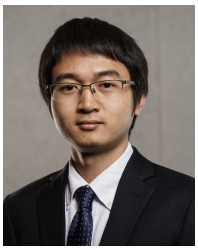
[54] A. Drutsa, "Reserve pricing in repeated second-price auctions with strategic bidders," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2678–2689.
[55] ——, "Optimal non-parametric learning in repeated contextual auctions with strategic buyer," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2668–2677.
[56] J. Lu, C. Yang, X. Gao, L. Wang, C. Li, and G. Chen, "Reinforcement learning with sequential information clustering in real-time bidding," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1633–1641.
[57] J.-S. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2010, pp. 60–68.
[58] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2803–2811.
[59] S. Chen, L. Jiao, L. Wang, and F. Liu, "An online market mechanism for edge emergency demand response via cloudlet control," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2566–2574.

Shutong Chen received her B.Sc. degree in the College of Mathematics and Econometrics, Hunan University, China. She is currently a Ph.D. student in the School of Computer Science and Technology, Huazhong University of Science and Technology, China. Her research interests include edge computing, green computing, and datacenter energy.



Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently an assistant professor at the Department of Computer and Information Science, University of Oregon, USA. Previously he worked as a member of technical staff at Alcatel-Lucent/Nokia Bell Labs in Dublin, Ireland and also as a researcher at IBM Research in Beijing, China. He is interested in the mathematics of optimization, control, learning, and mechanism design, applied to computer and telecommunication systems, networks, and services. He publishes papers in journals such as IEEE/ACM Transactions on Networking, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, and IEEE Journal on Selected Areas in Communications, and in conferences such as INFOCOM, MOBIHOC, ICNP, and ICDCS. He is a recipient of the NSF CAREER Award. He also received the Best Paper Awards of IEEE LANMAN 2013 and IEEE CNS 2019, and the 2016 Alcatel-Lucent Bell Labs UK and Ireland Recognition Award. He served as a guest editor for IEEE JSAC. He was on the program committees of many conferences including INFOCOM, MOBIHOC, ICDCS, and IWQoS, and was also the program chair of multiple workshops with INFOCOM and ICDCS.

Fangming Liu (S'08, M'11, SM'16) received the B.Eng. degree from the Tsinghua University, Beijing, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong. He is currently a Full Professor with the Huazhong University of Science and Technology, Wuhan, China. His research interests include cloud computing and edge computing, datacenter and green computing, SDN/NFV/5G and applied ML/AI. He received the National Natural Science Fund (NSFC) for Excellent Young Scholars, and the National Program Special Support for Top-Notch Young Professionals. He is a recipient of the Best Paper Award of IEEE/ACM IWQoS 2019, ACM e-Energy 2018 and IEEE GLOBECOM 2011, the First Class Prize of Natural Science of Ministry of Education in China, as well as the Second Class Prize of National Natural Science Award in China.



Lin Wang received the Ph.D. degree in computer science with distinction from the Institute of Computing Technology, Chinese Academy of Sciences. He is now an assistant professor at the department of computer science at VU Amsterdam. He has been a visiting researcher at IMDEA Networks Institute, Spain, from 2012-2014, a research associate at SnT Luxembourg from 2015-2016, and a group leader at TU Darmstadt, Germany from 2016-2018. He is interested in distributed systems and networking and his current research is focused on programming models and resource management for edge AI inference, in-network computing, and intermittent computing. He received the Athene Young Investigator award from TU Darmstadt in 2018 and has been a PI of the German Research Foundation (DFG) Collaborative Research Center for Future Internet in Germany.