A market-oriented incentive mechanism for emergency demand response in colocation data centers

Youshi Wang\textsuperscript{a,b}, Fa Zhang\textsuperscript{c}, Ce Chi\textsuperscript{a,b}, Shaolei Ren\textsuperscript{d}, Fangming Liu\textsuperscript{d}, Rui Wang\textsuperscript{e}, Zhiyong Liu\textsuperscript{a,*}

\textsuperscript{a} High Performance Computer Research Center, Institute of Computing Technology, Chinese Academy of Sciences, China
\textsuperscript{b} University of Chinese Academy of Sciences, China
\textsuperscript{c} University of California, Riverside, USA
\textsuperscript{d} Huazhong University of Science and Technology, China
\textsuperscript{e} Microsoft, Beijing, China

**Abstract**

Rapidly developing colocation data centers (or colocations, for short) have become important participants in emergency demand response (EDR) programs. Different from traditional data centers, in colocations, tenants control their own servers; thus, they may not coordinate to reduce their power consumption. In this paper, to encourage tenants to join EDR programs, we propose a market-oriented incentive mechanism, MicDR, which can effectively reduce energy costs. MicDR includes a local incentive mechanism (LiMec), a global incentive mechanism (GiMec) and a server-sharing incentive mechanism (SiMec). LiMec motivates tenants to improve their energy efficiency locally. GiMec encourages tenants to improve their energy efficiency by requesting public server resources. To support the requests sparked by GiMec, SiMec ensures tenants to share idle server resources. A \((1+\epsilon)\)-approximation algorithm is proposed to achieve an asymptotic optimal energy-saving scheme. The performance of the proposed algorithm is evaluated, and trace-driven simulations verify the effectiveness and feasibility of MicDR.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Large-scale data centers are power-hungry, but their power demands are flexible [2]. Thus, data centers can participate in demand response (DR) programs, especially in emergency demand response (EDR) programs [3]. EDR is a widely adopted approach to improve the fragile power infrastructure. When emergency events (e.g., extreme weather) occur, EDR providers inform all participants, providing them with a fixed energy-saving target [4]. Then, the participants should reduce their energy consumption to achieve the energy-saving target.

In recent years, one important type of data center, called a colocation data center (or colocation, for short), has developed rapidly. Colocations currently account for approximately 37.3\% of all data centers [5]. Colocations help tenants build private data centers by providing professional infrastructure and services, and they are often located in metropolitan areas [6]. Due to the high population densities in metropolitan areas, colocations incur high energy demands, and the available energy is frequently insufficient. Thus, it is necessary for colocations to participate in EDR programs to avoid energy shortages and improve power grid stability [7].

To achieve EDR in these colocations, we focus on ways to improve the energy efficiency of colocations. Energy efficiency technologies have been widely investigated for traditional data centers (e.g., server resource virtualization [8], traffic engineering [9] and energy-efficient data center networks (DCNs) [10]). However, in these works, all the facilities, e.g., servers and infrastructure, were fully controlled by the data center operators. Fig. 1 shows the structure of the colocation data center. The colocation consists of two parts (i.e., the infrastructure and the IT equipment). The infrastructure is managed by the colocation operator. The operator is also responsible for the routine maintenance of the colocation. In contrast, the IT equipment, such as servers in colocations, are fully
controlled by tenants. Thus, due to the lack of control of the IT equipment, approaches that focus on improving the energy efficiency of traditional data centers are not feasible for colocations.

The special management pattern of colocations was considered in [11],[12] and termed the “uncoordinated relationship” issue, which includes two aspects. First, tenants lack coordination with the colocation operator for energy saving purposes. Some retail tenants pay for energy in advance based on their peak demand. Thus, saving energy is of no benefit to the tenants, and they have no incentive to reduce their energy consumption. Others are wholesale tenants who are charged for energy based on their consumption. This approach ensures that tenants prefer to minimize the energy they use to save costs. However, such tenants may not reduce their energy consumption when an EDR situation occurs. Thus, designing an incentive mechanism to encourage tenants to coordinate with the colocation operator to save energy is a critical problem. Second, tenants lack coordination with each other. By incentivizing the coordination among tenants, the colocation operator can make global optimization decisions rather than relying on the local optimization of each tenant. Thus, designing an incentive mechanism to encourage tenants to coordinate with each other is also a crucial problem.

Many efforts have been made to address the “uncoordinated relationship” between colocation operators and tenants [4,6,12–18]; however, these studies have ignored the problem of coordination among tenants and thus cannot yield good energy efficiency. In this work, we jointly consider coordination between tenants and operators and coordination among tenants.

1.1. Related work

As shown in Fig. 2, there are two main approaches to achieve an EDR program in colocations. One is to use a backup energy storage (BES) system, and another is to design incentive mechanisms. In the mechanism design, two situations of “uncoordinated relationship” issues are considered. One is the “uncoordinated relationship” between the colocation operator and tenants, and the other is the “uncoordinated relationship” among tenants.

Due to the isolation of the colocation operator and tenants, the most popular solution of the EDR program for colocations is to replace the power grid with their own BES system. As shown in Fig. 1, a BES system includes uninterrupted power supply (UPS) and diesel generators. Several prior studies have focused on how to take advantage of BES systems to optimize the total cost of data centers [19–23]. In [19], for saving electricity costs, battery management technology was jointly considered with center-level load balancing as well as the server-level configuration. An optimization framework was proposed in [20] that leverages BES systems in data centers to jointly optimize both peaking shaving and regulation market participation for reducing electricity costs. In [21], a BES system was used to reduce data centers’ expenses, considering leakage losses, conversion losses and charging/discharging rates. By leveraging BES systems, the colocation operators can reduce the power demand from the power grid, thus achieving the EDR target. Although this solution bypasses the “uncoordinated relationship” issue, it is far from a good solution due to the high cost and/or high pollution of BES systems.

Some works have considered the energy efficiency issue of colocations. The “split incentive” issue in colocations was first considered in [9], and an incentive mechanism, iCODE, was proposed to incentivize tenants to join the EDR programs. Some subsequent works focused on how to improve the incentive mechanism (e.g., Truth-DR [12]). A joint DR that included economic DR and emergency DR was discussed in [4]. A novel thermal-aware and cost-efficient mechanism called TECH was proposed in [13]. A contract-based mechanism, Contract-DR, was proposed in [14], and different types of tenant costs that applied complete tenants’ information and incomplete tenants’ information were discussed. In [15], RECO, which provided financial rewards to improve tenants’ power management, considered three challenges: the time-varying operation environment, a peak power demand charge and the tenants’ unknown responses to the offered rewards. In [16], the demand response provider (DRP) was considered and an incentive mechanism called R2R was proposed to show the interaction between compensation from the DRP and rewards paid to tenants. Some other works exist that consider the incentivization issue in colocations [17,18]. A common feature of the above works is that they all focused on how to incentivize tenants to coordinate with the colocation operator. However, in these studies, the tenants work independently without coordination, which is not conducive to good resource utilization and energy efficiency.

To solve the coordination issue among tenants, a novel framework that incentivizes tenants to reduce their energy consumption via public resources was proposed in [11]. However, three key problems remained unsolved in this framework. First, it did not consider how to ensure the authenticity of tenants’ declared cost2 when they requested public server resources to migrate their workloads. However, guaranteeing truthfulness is a key feature in the design of such a mechanism. Second, the study assumed that the available public server resources could satisfy all the requests. Third, it considered using only some cloud resources and some standby servers to build the public resources. However, it is more cost-efficient to additionally consider tenants’ idle servers.

---

1 A “split incentive” denotes that while the colocation operator is requested to respond to the energy reduction target from the power grid, the tenants may have no incentive to comply.

2 “Declared cost” denotes the evaluated energy-saving loss by tenants based on their energy reduction target.
Table 1
Notation description.

<table>
<thead>
<tr>
<th>( \mathcal{N} )</th>
<th>Set of tenants</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta )</td>
<td>PUE of the colocation</td>
</tr>
<tr>
<td>( p_{ij} )</td>
<td>Average utilization of tenant ( i ) at the time slot ( j )</td>
</tr>
<tr>
<td>( E )</td>
<td>( \tau )</td>
</tr>
<tr>
<td>( \lambda_i, \mu_i )</td>
<td>Parameters of tenant ( i )'s utility function</td>
</tr>
<tr>
<td>( \psi^i )</td>
<td>Optimal solution of ( \psi^i ) for tenant ( i )</td>
</tr>
<tr>
<td>( y^{inv}_{i} )</td>
<td>Unit payment for a shared server in the colocation</td>
</tr>
<tr>
<td>( d_i )</td>
<td>Declared cost of tenant ( i ) in LiMec</td>
</tr>
<tr>
<td>( l_i )</td>
<td>Tenant ( i )'s bid in LiMec</td>
</tr>
<tr>
<td>( c_i )</td>
<td>Declared cost of tenant ( i ) in GiMec</td>
</tr>
<tr>
<td>( B^* )</td>
<td>Bid set in GiMec</td>
</tr>
<tr>
<td>( D^* )</td>
<td>Number of cloud VM instances from cloud providers</td>
</tr>
<tr>
<td>( G )</td>
<td>Number of VM instances in the shared cloud</td>
</tr>
<tr>
<td>( T_{opt} )</td>
<td>The theoretical optimal cost of problem (P_{1})</td>
</tr>
<tr>
<td>( T_s )</td>
<td>The upper bound of ( T_{opt} )</td>
</tr>
<tr>
<td>( R )</td>
<td>The cost function of building shared cloud</td>
</tr>
<tr>
<td>( M_i )</td>
<td>Number of tenant ( i )'s servers</td>
</tr>
<tr>
<td>( h^m )</td>
<td>Number of VM instances held by tenant ( i )'s server</td>
</tr>
<tr>
<td>( C_{ij} )</td>
<td>Upper bound of tenant ( i )'s available servers at the time slot ( j )</td>
</tr>
<tr>
<td>( g_e )</td>
<td>Power supply from the BES</td>
</tr>
<tr>
<td>( \psi^p )</td>
<td>Number of needed servers in tenant ( i )'s available servers</td>
</tr>
<tr>
<td>( \psi^{i}_{s} )</td>
<td>Number of shared servers in tenant ( i )'s available servers</td>
</tr>
<tr>
<td>( y^{inv}_{s} )</td>
<td>Unit cost of shared servers for tenant ( i )</td>
</tr>
<tr>
<td>( e_i )</td>
<td>Energy reduction target of tenant ( i ) in LiMec</td>
</tr>
<tr>
<td>( b^* )</td>
<td>Tenant ( i )'s bid in GiMec</td>
</tr>
<tr>
<td>( \delta )</td>
<td>Unit price of a VM instance for cloud providers</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>The approximation ratio parameter for Algorithm 3</td>
</tr>
<tr>
<td>( T_l )</td>
<td>The lower bound of ( T_{opt} )</td>
</tr>
<tr>
<td>( K )</td>
<td>A normalized parameter</td>
</tr>
</tbody>
</table>

1.2. Contributions

In this paper, we design a market-oriented incentive mechanism (MicDR) that not only encourages coordination between tenants and the colocation operator but also encourages coordination among tenants to further improve resource utilization and reduce energy consumption. To achieve the coordination among tenants, MicDR allows tenants to request server resources from a shared cloud to further integrate tasks. In the shared cloud, the colocation operator can improve resource utilization by integrating heterogeneous tasks and resources based on centralized control. Then, to support tenants’ requests and reduce the cost for public server resources, tenants are motivated to share their idle server resources. The main contributions of this paper are summarized as follows.

- We propose a novel mechanism, MicDR, which not only encourages tenants to optimize their local servers, but also incentivizes tenants to improve their energy efficiency based on a shared cloud. Meanwhile, based on a Stackelberg game, MicDR is also designed to incentivize tenants to share their idle servers to supplement the shared cloud.
- MicDR is formulated as a mixed-integer nonlinear programming (MINLP) problem, and we develop a \((1+\epsilon)\)-approximation algorithm to solve it. For the developed algorithm, we discuss its time complexity and prove that it can satisfy the truthfulness requirement of MicDR.
- Based on the developed algorithm, we provide detailed proofs that MicDR is a truthful and feasible mechanism. We also demonstrate that MicDR can achieve the Nash equilibrium when tenants share their idle servers to supplement the shared cloud.
- We validate the efficiency of the proposed mechanism and algorithm through simulations based on real workload traces and show that MicDR can achieve significant energy efficiency improvements in colocations.

1.3. Organization

The rest of this paper is organized as follows. In Section 2, we introduce the colocation model, the EDR model and the tenant server price model. In Section 3, a novel mechanism is proposed to solve the “uncoordinated relationship” issue, and we formulate the cost minimization problem for the proposed mechanism. We solve the problem though an effective algorithm developed in Section 4. Then, the truthfulness and feasibility of the proposed mechanism are proved in Section 5. In Section 6, we show the simulation results and verify the practical performance of our mechanism. Finally, we conclude this paper in Section 7.

2. System model

In this section, we introduce the colocation model and the server price model. The parameters used in this paper are listed and described in Table 1.

2.1. Colocation and EDR model

We consider a colocation data center with \( n \) tenants denoted as \( \mathcal{N} = \{1, 2, \ldots, n\} \). For tenant \( i \in \mathcal{N} \), we use \( M_i \) to denote the number of its servers. The Power Usage Effectiveness (PUE) is used to describe the ratio of a colocation’s total energy consumption to its IT energy consumption, denoted as \( \beta \). The computing capacity of a server is considered to be that of an m4.large virtual machine (VM) instance, which is the latest generation of general purpose instances in the Amazon Elastic Compute Cloud (EC2). We assume that all the tenants’ servers are homogeneous and that each server can hold \( \lambda^m \) VM instances. Then, we divide one day into twenty-four time slots and measure the average utilization of tenant \( i \) during time slot \( j \), denoted as \( \rho_{ij} \). We use \( C_{ij} \) to denote the upper bound of the available servers for tenant \( i \) at time slot \( j \), where \( C_{ij} = \left[ 1 - \rho_{ij} \right] \cdot M_i \).

When the colocation is notified of an EDR program, the colocation operator is requested to reduce the energy demand on the power grid, and the reduction target is denoted as \( E \), which is a fixed value for the colocation operator. A traditional method adopted by the colocation operator is to run its own BES system. The BES’s power supply is denoted as \( g_e \), where \( \tau \) is the unit cost of power generation.

2.2. Tenant server price model

In this subsection, a model is introduced to measure the utility of tenants’ available servers. For tenant \( i \in \mathcal{N} \), we use \( u_i(\psi^p, C_{ij}) \) to denote its utility when the tenant is using \( \psi^p \) servers for its own needs. Based on the law of diminishing returns, a logarithmic function is used to evaluate the utility \( u_i(\psi^p, C_{ij}) \) [24][25], given as

\[
u_i(\psi^p, C_{ij}) = \psi^p \ln(\frac{\mu_i \psi^p}{C_{ij}}),\]

where \( \lambda_i \) and \( \mu_i \) are coefficients of the utility function. We assume that tenants can share some servers for public use. For tenant \( i \), the number of shared servers is denoted as \( \psi^s_i \) with the unit cost \( \gamma^{cost} \).
thus, the total cost for sharing servers can be expressed as $\psi_i^r \cdot \gamma_{\text{cost}}$. In addition, tenant $i$ can also earn some income from the colocation operator. Let $\gamma_{\text{pay}}$ denote the unit payment for a shared server. By sharing servers, tenant $i$ can obtain as the following: $\psi_i^r \cdot \gamma_{\text{pay}}$.

3. Incentive mechanism design

For solving the “uncoordinated relationship” issue, we propose a joint incentive mechanism called MicDR to solve the “uncoordinated relationship” issue and satisfy the energy reduction target of an EDR program. In the following, we first introduce the implementation details and mathematical formulations for these three sub-mechanisms of MicDR. Then, considering the framework of MicDR, the cost minimization problem of colocations is formulated.

3.1. Designs of the sub-mechanisms

3.1.1. LiMec

The “uncoordinated relationship” issue between tenants and the colocation operator stems from a lack of incentives: tenants have no incentive to optimize their own servers if they do not profit by doing so. An intuitive approach is to provide some economic benefits to incentivize tenants to improve their energy efficiency. Thus, we adopt LiMec, a local incentive mechanism, to incentivize tenants to cooperate with the colocation operator to save energy during an EDR program.

In LiMec, tenants can be regarded as the sellers and the operator as the buyer. Thus, it forms a typical auction pattern called a “reverse auction” [6]. For tenant $i \in \mathcal{N}$, let $e_i$ and $d_i$ denote the energy-saving target and the declared cost, respectively. In LiMec, tenant $i$’s bid can be expressed as $b_i = (e_i, d_i)$. Then, we use $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$ to denote the set of bids.

3.1.2. GiMec

Beyond the lack of incentives, another important issue is that each tenant is relatively independent in the colocation. Thus, the “uncoordinated relationship” issue also exists among tenants. We discuss two situations in which tenants gain by cooperating with each other. First, tenant cooperation can help reduce energy waste by maintaining high server usage rates. For example, for each tenant, when a task executes on and requires only one VM instance, keeping a server running or turning it on may waste energy. However, if the tenants were to cooperate by merging tasks, a higher level of energy efficiency could be maintained. Second, when tenants cooperate, tasks can be integrated more efficiently. Tasks can be divided into several types (e.g., CPU-bound and I/O-bound). Each tenant’s tasks may belong to a single type. Assuming that most tasks are the I/O-bound type, optimally integrating these tasks will free many CPU resources and cause idle servers. In contrast, when tenants cooperate with each other, all the resources can be better allocated by integrating different types of tasks to achieve higher utilization and thus better energy efficiency. Therefore, in addition to LiMec, we also design GiMec, a global incentive mechanism, which helps tenants cooperate in further optimizing colocation utilization and energy efficiency.

Unlike in LiMec, it is infeasible to pay economic benefits directly to tenants to incentivize them to cooperate with each other. Thus, we propose to provide a shared cloud so that tenants can integrate different types of tasks. The minimum unit in the shared cloud is a VM instance. In GiMec, we collect tenants’ information in the form of bids. For tenant $i \in \mathcal{N}$, $g_i$ denotes the number of required VM instances, and $s_i$ and $c_i$ denote tenant $i$’s energy-saving target and declared cost, respectively. Therefore, tenant $i$’s bid can be expressed as $b_i = (s_i, c_i, g_i)$. Then, we use $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$ to denote the set of bids.

3.1.3. SiMec

In GiMec, an important issue is how to build the shared cloud. First, because colocations are typically built in downtown areas close to customers, cloud providers such as Amazon and Google have increasingly deployed part of their servers in these colocations. Thus, to satisfy tenants’ VM instance demands, the colocation operator can rent cloud VM instances. We assume that the colocation operator rents $D^*$ VM instances at a unit price of $\delta$ from the cloud providers.

Considering that tenants in the colocation usually have some idle servers, it is feasible to incentivize tenants to share their idle servers to build the shared cloud. Thus, we design SiMec, which is a server-sharing incentive mechanism, to build a supply-and-demand relationship between the colocation operator and the tenants based on a Stackelberg game which includes a leader and multiple followers. Specifically, the leader moves first, and the followers then react sequentially [26]. In SiMec, the colocation operator (who is responsible for building the shared cloud for GiMec) acts as the leader, and the tenants (who share some servers to maximize their profits) are regarded as the followers. The colocation operator first issues the unit price for shared servers; then the tenants react independently and selfishly to the unit price to decide how many servers they will share [27]. The shared cloud is composed of both the cloud resources and the servers shared by tenants.

For tenant $i \in \mathcal{N}$, the total profit $U_i$ is composed of three parts: the utility of the servers it needs, the server-sharing cost and the payment amount from the colocation operator. Then, we have

$$U_i(\psi_i) = U_i(C_{ij} - \psi_i^r, C_{ij}) - \gamma_{\text{cost}}^r \psi_i^r + \gamma_{\text{pay}} \psi_i^r.$$  

To maximize its own profit, tenant $i$ can play a subgame, given as

$$\psi_i^* = \arg \max_{\psi_i \in [0, C_{ij}]} U_i(\psi_i),$$

where $\psi_i^*$ is the optimal solution of $\psi_i^r$ for tenant $i$ to achieve the maximum revenue. By calculating the stationary point, the optimal number of shared servers for tenant $i$ can be expressed as follows:

$$\psi_i^* = \begin{cases} \frac{C_{ij} - \lambda_i}{\gamma_{\text{pay}} - \gamma_{\text{cost}}} & \text{if } \gamma_{\text{pay}} > \gamma_{\text{cost}}^r \\ 0 & \text{if } \gamma_{\text{pay}} \leq \gamma_{\text{cost}}^r \end{cases}$$

We assume that the parameters $C_{ij}$, $\lambda_i$ and $\gamma_{\text{cost}}^r$ can be estimated by a machine-learning algorithm. Thus, the colocation operator can predict the reaction of tenant $i \in \mathcal{N}$ for any unit payment $\gamma_{\text{pay}}$.

3.2. A market-oriented incentive mechanism: MicDR

The framework of MicDR is shown in Fig. 3, and the detailed steps are as follows: (1) MicDR begins when an EDR program notification reaches the colocation operator and specifies the energy reduction target. (2) The colocation operator announces the beginning of MicDR, and tenants can decide whether to submit bids to improve their own energy efficiency. This is a reverse auction process. (3) Tenants make their bids based on their own traffic loads and task types. Two sub-mechanisms can be chosen independently. Accordingly, tenant $i$ can determine the bids $b_i^L$ and $b_i^G$ in the sub-mechanisms LiMec and GiMec, respectively. Then, all bids are collected and submitted to the colocation operator. (4) Based on all collected bids, the colocation operator makes the optimal decision to achieve the overall energy-saving target with the minimal cost. However, to satisfy the capacity of the shared cloud, sub-mechanism SiMec builds a server sharing market between the colocation operator and tenants based on the Stackelberg game theory. Thus, in the fourth step, the colocation operator sets the market price $\gamma_{\text{pay}}$ of shared servers and notifies all tenants. (5) Tenants
receive the market price $\gamma_{\text{pay}}$, and the optimal number of shared servers is computed based on Eq. (3). Then, to obtain the benefits, tenants share $\psi_{i}^\ast (i \in \{1, \mathcal{N}\})$ under sub-mechanism SiMec.

(6) The colocation operator pays for all selected bids in LiMec and GiMec. Based on $\gamma_{\text{pay}}$ and $\psi_{i}^\ast (i \in \{1, \mathcal{N}\})$, the colocation operator also pays for tenants’ earnings of sharing idle servers. (7) The shared cloud provides server resources based on the bids $b_i \in \{1, \mathcal{N}\}$, and the energy efficiency optimization officially launches in the colocation.

In MicDR, we assume that two bids in LiMec and GiMec from the same tenant can be selected independently. The VM instances in the shared cloud involve two factors. First, the colocation operator can rent $D^\ast$ VM instances from the cloud providers. Second, based on SiMec, tenant $i \in \mathcal{N}$ can share $\psi_{i}^\ast$ servers. Then, the cost to the colocation operator includes the payments for the selected bids in LiMec and GiMec, the cost of using its own BES system and the payment for the VM instances in the shared cloud.

Thus, the cost minimization problem for the colocation operator can be formulated as problem (P$_1$), as follows:

$$ \begin{align*}
\text{(P$_1$)} \quad & \min \delta D^\ast + \tau_{\text{ge}} + \sum_{i \in \mathcal{N}} (\gamma_{\text{pay}} \cdot h_{\text{vm}} \cdot \psi_{i}^\ast + d_i x_i + c_i y_i) , \\
\text{s.t.} \quad & g_e + \beta \sum_{i \in \mathcal{N}} (e_i x_i + s_i y_i) \geq E, \quad (4a) \\
& \sum_{i \in \mathcal{N}} g_{yi} \leq \delta D^\ast + h_{\text{vm}} + \sum_{i \in \mathcal{N}} \psi_{i}^\ast, \quad (4b) \\
& \gamma_{\text{pay}}, \psi_{i}^\ast \in \{0, 1\}, \quad (4c) \\
& D^\ast, g_e \in \mathcal{N}, \quad (4d) \\
& x_i, y_i \in \{0, 1\}, \quad i \in \mathcal{N}, \quad (4e)
\end{align*}$$

where $x_i$ denotes whether tenant $i$’s bid is selected in LiMec and $y_i$ denotes whether tenant $i$’s bid is selected in GiMec. Constraint (4a) guarantees the energy reduction target, and Constraint (4b) implies that the required VM instance in GiMec is less or equal to that in the shared cloud.

### 4. Algorithm design and analysis

In this section, we first discuss how to obtain a feasible approximate solution to problem (P$_1$) and then analyze the developed algorithms’ features and theoretical performances.

(P$_1$) is a mixed-integer nonlinear programming (MINLP) problem, and it is NP-hard in general [28]. Compared with the problem formulated in Truth-DR [12], an extra constraint, Constraint (4b), is added in our problem, which is about the capacity constraint of shared cloud. To our knowledge, considering the new constraint, no feasible algorithm exists to solve the problem. Moreover, there are two requirements for the algorithm design. First, the developed algorithms should solve problem (P$_1$) with a reasonable time complexity. Second, the truthfulness of MicDR must be guaranteed by the design of the algorithm. Thus, we first divided and reformulated problem (P$_1$) and developed algorithms for the resulting problems. Then, by combining all the developed algorithms, we can obtain a $(1 + \epsilon)$-approximation solution for (P$_1$).

We use $T_{\text{opt}}$ to denote the theoretical optimal cost of (P$_1$) and assume that $T_L$ and $T_u$ can satisfy $T_L \leq T_{\text{opt}} \leq T_U$. Let $(x_i', y_i') (i \in \{1, \mathcal{N}\})$ denote a feasible solution vector for (P$_1$); thus, $T_u$ can be expressed as $T_u = \sum_{i \in \mathcal{N}} (d_i x_i' + c_i y_i')$. Note that if it is difficult to find a feasible solution vector, we let $T_u = \min_{i \in \mathcal{N}} (x_E, \sum_{i \in \mathcal{N}} (d_i + c_i) + \tau \cdot \max((E - \sum_{i \in \mathcal{N}} (d_i + s_i))}, 0))$. Then, $T_L$ is expressed as $T_L = \min_{i \in \mathcal{N}} (x_E, \eta_1, \eta_2, \eta_3)$, where $\eta$ is a constant and denotes the lower bound of the smallest energy-saving unit price. We define

$$K = \frac{e \cdot T_L}{2n},$$

where $e$ is a parameter related to the approximate ratio of Algorithm 3. $K$ is a scaling parameter that is used to solve the dual problem of (P$_1$). It is also important to guarantee the approximation ratio of Algorithm 3. We will show how to obtain the value of $K$ in Lemma 2. Then, let $d_i = \frac{d_i}{D^\ast}$ and $c_i = \frac{c_i}{D^\ast}$. Then, we use $F(G)$ to denote the cost of building a shared cloud with at least $G$ VM instances. Based on SiMec, the colocation operator can obtain $(D^\ast + h_{\text{vm}} \cdot \sum_{i \in \mathcal{N}} \psi_{i}^\ast)$ VM instances. Thus, we can formulate the cost minimization problem for $F(G)$:

$$\begin{align*}
F(G) = \min \delta D^\ast + \gamma_{\text{pay}} \cdot h_{\text{vm}} \cdot \sum_{i \in \mathcal{N}} \psi_{i}^\ast, \\
\text{s.t.} \quad D^\ast + h_{\text{vm}} \cdot \sum_{i \in \mathcal{N}} \psi_{i}^\ast \geq G. \quad (5) \\
\end{align*}$$

Accordingly, we can transform (P$_1$) to (P$_2$):

$$\begin{align*}
\text{(P$_2$)} \quad & \min F(G) + \tau_{\text{ge}} + K \sum_{i \in \mathcal{N}} (d_i x_i + c_i y_i) , \\
\text{s.t.} \quad g_e + \beta \sum_{i \in \mathcal{N}} (e_i x_i + s_i y_i) \geq E, \quad (5a) \\
& \sum_{i \in \mathcal{N}} g_{yi} \leq G, \quad (5b) \\
& x_i, y_i \in \{0, 1\}, \quad i \in \mathcal{N}. \quad (5c)
\end{align*}$$

To simplify (P$_2$), we first consider the case with fixed $G$ and $g_e$, denoted as $G_f$ and $g_{e,f}$ respectively. The minimum cost of building a shared cloud with at least $G_f$ VM instances can be denoted as $F(G_f)$. Then, problem (P$_2$) can be simplified as follows:

$$\begin{align*}
\text{(P$_3$)} \quad & \min F(G_f) + \tau_{\text{ge},f} + K \sum_{i \in \mathcal{N}} (d_i x_i + c_i y_i) , \\
\text{s.t.} \quad \beta \sum_{i \in \mathcal{N}} (e_i x_i + s_i y_i) \geq E, \quad (6a) \\
& \sum_{i \in \mathcal{N}} g_{yi} \leq G_f, \quad (6b) \\
& x_i, y_i \in \{0, 1\}, \quad i \in \mathcal{N}. \quad (6c)
\end{align*}$$

where $E = E - g_{e,f}$.
Because $F(G_f)$ and $r_{G_f}$ are both constants in problem $(P_3)$, we can ignore them when calculating the optimal solution vector of $(P_3)$. Then, based on $(P_3)$, we can formulate the dual problem $(P_3^\ast)$:

$$(P_3^\ast) \quad \max \sum_{i \in N} e_i x_i + s_j y_j, \quad \text{s.t.} \quad \sum_{i \in \text{N}} d_i x_i + c_j y_j \leq t,$$

$$\sum_{i \in \text{N}} g_i y_j \leq G_f,$$

$$x_i, y_j = 0, 1, \quad i \in \mathcal{N}, \quad j \in \mathcal{J},$$

where $T' \in \mathcal{T}$ (i.e., $T = \{ [ \frac{n}{2}, \frac{n}{2} ] + 1, \ldots, 2n \}$).

To solve the dual problem $(P_3^\ast)$, we adopt the dynamic programming (DP) approach. Let $(k, t, G_f)$ denote a state of DP where $k \in \{0, 1, \ldots, 2n\}$, $t \in \{0, 1, \ldots, T'\}$, and $G_f' \in \{0, 1, \ldots, G_f\}$. Thus, for the state $(k, t, G_f')$, a sub-problem of $(P_3^\ast)$ is to maximize the energy saving for the first $k$ bids, where the payment for the selected bids is no greater than $t$ and the required VM instance is less or equal to $G_f'$, given as

$$(P_3^\ast) \quad \max \sum_{1 \leq i \leq \min(n, k)} e_i x_i + \sum_{1 \leq i \leq k-n} s_j y_j, \quad \text{s.t.} \quad \sum_{1 \leq i \leq \min(n, k)} d_i x_i + \sum_{1 \leq i \leq k-n} c_j y_j \leq t,$$

$$\sum_{1 \leq i \leq k-n} g_i y_j \leq G_f',$$

$$x_i, y_j = 0, 1, \quad i \in \mathcal{N}.$$

The optimal solution of the state $(k, t, G_f')$ is denoted as $OPT(k, t, G_f')$. The corresponding optimal decision variables are denoted as $x_i(k, t, G_f')$ and $y_j(k, t, G_f')$, where $x_i(k, t, G_f') = 1$ or $0$ and $y_j(k, t, G_f') = 1$ or $0$. For the DP process, the initial state is set to

$$OPT(k = 0, t \geq 0, G_f' \geq 0) = 0$$

and

$$OPT(t < 0, G_f' < 0) = -INF.$$
the energy saving target. Similarly, we use $C^F_b$ to denote $\min \, \mathcal{G}_i$.

Let $\mathcal{D}_i(b_i)$ [i ∈ [1, 2n]] denote that bid $b_i$ is deleted from set $\mathcal{D}$, which can be written as $\mathcal{D}_i(b_i) = \{b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{2n}\}$. In set $\mathcal{D}_i(b_i)$ [i ∈ [1, 2n]], the approximate optimal solution of (P1) can be denoted as $\psi^E_{\mathcal{D}_i(b_i)}$. Let $p^i_1 = V^E_{\mathcal{D}_i(b_i)}$ and $p^i_2 = V^E_{\mathcal{D}_i(b_i)}$. Then, the market price $p_i$ of bid $b_i \in \mathcal{D}$ can be written as follows:

$$p_i = p^i_1 - p^i_2 = V^E_{\mathcal{D}_i(b_i)} - V^E_{\mathcal{D}_i(b_i)}.$$  

(9)

Let $h_{i}^{true}$ denote the authentic cost of the bid $b_i \in \mathcal{D}$, and let $u_i$ denote bid $b_i$’s utility. To guarantee the authenticity of bids, Lemma 3 shows that it is impossible for any tenant to obtain higher utility by declaring a false cost.

**Lemma 3. (Truthfulness)** If a bid $b_i \in \mathcal{D}$ declares a false cost, $h_{i}^{false}$, its utility does not increase.

To guarantee the feasibility of our pricing strategy, Lemma 4 shows that when a bid is selected, its utility is nonnegative.

**Lemma 4. (Feasibility)** If bid $b_i \in \mathcal{D}$ is selected, its utility is nonnegative.

Finally, we explain how Nash equilibrium can be achieved based on SiMec. SiMec is designed based on a Stackelberg game. We use the utility function in Eq. (1) to describe the total profits when tenant $i$ shares the servers. The utility function in Eq. (1) builds a supply-and-demand relationship between the colocation operator and the tenants, but assumes that the tenants operate independently, which means that each tenant also makes decisions independently. Then, tenant $i \in \mathcal{N}$ can calculate the optimal number of shared servers $\psi^*$ to maximize its total profits, as shown in Eq. (3). Moreover, because tenants are independent, whether tenant $i$ changes the number of shared servers $\psi^*$ has no influence on the decisions of the other tenants. Thus, for tenant $i \in \mathcal{N}$, $\psi^*$ can be regarded as its Nash equilibrium point. Accordingly, we find that SiMec can achieve Nash equilibrium.

Most importantly, we have shown that MicDR can guarantee the authenticity of tenants’ bids based on the VCG theory and have also proved that tenants always obtain positive utility when their bids are selected. In addition, we explain how SiMec can build a supply-and-demand relationship between the colocation operator and the tenants and achieve the Nash equilibrium. Thus, we can conclude that MicDR is a truthful and feasible mechanism.

### 6. Performance evaluation

In this section, we present simulations conducted to evaluate the performance of the proposed MicDR mechanism. We first introduce the simulation settings, which are based on both widely used parameters [6,12,24,32,33] and real traces [2,34]. Then, we validate the performance of the proposed approximation algorithm. Finally, the simulation results verify the effectiveness and feasibility of MicDR.

#### 6.1. Settings

**Colocation data center:** Assume that there are six tenants (denoted as Tenant #1, Tenant #2, . . . , Tenant #6) in the colocation, and each tenant has 10,000 servers. We assume that all the servers in the colocation are homogeneous Dell PowerEdge R730s. We also adopt the m4.large VM instance from the Amazon EC2 to measure the capacity of each server. Under these conditions, each server can hold five m4.large VM instances, i.e., $h_{nm} = 5$. Moreover, we can obtain the price of an m4.large VM instance from an Amazon EC2 spot instance [35]. Specifically, we set $\delta$ to 1.55 cents/h based on the price of a m4.large VM instance on October 25, 2017 in Ohio, USA. We assume that the static power $P_s$ and the dynamic power $P_d$ of a server are 0.15 kW and 0.1 kW, respectively [6][33]. A reasonable PUE for the colocation is set to 1.6 [12]. Accordingly, we can obtain the peak power of the colocation, which is 24 MW.

**Energy reduction target and workload:** The energy reduction targets were sourced from PJM’s EDR on April 22, 2015 [34], and the data are scaled down to 15% of the colocation’s peak power to avoid affecting normal operations [32]. The energy reduction targets are shown in Fig. 4(a). Eight events occurred from 6 am to 13 pm, and each event lasted for one hour. The workload traces were obtained from “MSR” and “Florida International University” [2], as shown in Fig. 4(b).

**Tenants’ energy reduction and costs:** In MicDR, we consider that tenants optimize energy efficiency by turning off idle servers [38]. In SiMec, these idle servers may be shared to build the shared cloud, which is managed by the colocation operator in a unified manner for achieving higher energy efficiency. From a global view, sharing servers is helpful to further optimize the energy efficiency. In LiMec, let $n_{ij}$ denote the number of turned-off servers for tenant $i$. Then, we
Table 2
Theoretical performance comparisons among different mechanisms.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Approximate ratio</th>
<th>Time complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicDR</td>
<td>MINLP</td>
<td>$1 + \varepsilon$</td>
</tr>
<tr>
<td>Truth-DR [12]</td>
<td>MILP</td>
<td>2</td>
</tr>
<tr>
<td>LG-Mec</td>
<td>ILP</td>
<td>$1 + \varepsilon$</td>
</tr>
<tr>
<td>Branch-bound</td>
<td>INLP</td>
<td>1</td>
</tr>
</tbody>
</table>

find that $e_i = n_j \cdot P^* \cdot T$, where $T = 1$ hour is one EDR period. Next, let $n_{i,j}$ denote the number of servers joining the GiMec mechanism for tenant $i$. The utilization of each server is denoted as $\rho_k$, where $k \in [1, n_{i,j}]$. Thus, we obtain $s_i = T \cdot \sum_{k=1}^{n_{i,j}} (1 - \rho_k)P^*$. We assume that each tenant has its own bid parameter, $\theta_i$, which is used to distinguish the tenants’ different expected costs. From [12], we know that the tenants’ costs obey a uniform distribution between 6.7 and 13.3 cents/kWh. Thus, based on the tenants’ energy reductions, tenant $i$’s costs $b_i$ and $c_i$ can be obtained.

Server price model: The available servers for tenant $i$ in time slot $j$ is denoted by $C_{i,j} = \sum_{k=1}^{n_{i,j}} M_k$, where $M_k = 10,000$ and $\rho_k$ can be obtained from Fig. 4(b). Then, the parameters $\gamma^{\text{cost}}$ and $\mu_j$, $i \in N$, are set to 0 and 100, respectively [24]. Thus, the parameter $\alpha_i$ can be given as $\alpha_i = \frac{\delta_{i,j}}{m_{i,j}}$.

6.2. Results and analysis

MicDR is a market-oriented incentive mechanism composed of three sub-mechanisms: LiMec, GiMec and SiMec. In Section 1, we mentioned that most existing works focused on incentivizing tenants to reduce energy consumption as does LiMec (e.g., iCODE [6] and Truth-DR [12]). Thus, we choose one typical mechanism, Truth-DR, as a control group. To evaluate the effects of SiMec, we also consider a mechanism that includes LiMec and GiMec but utilizes only cloud VM instances in GiMec, termed LG-Mec in the following results.

6.2.1. Theoretical performance

We compare the theoretical performance of different mechanisms in Table 2 using the branch-and-bound strategy as a baseline. Compared with LG-Mec, MicDR involves a more complicated mathematical problem. Moreover, MicDR maintains both the same approximate ratio and a similar time complexity as LG-Mec. Compared with Truth-DR, MicDR can solve more complex problems with a better approximate ratio. The response time of an EDR program is always within several hours. Thus, considering the scale of colocations, the time complexity of MicDR is acceptable and can satisfy the time limitation of an EDR program.

6.2.2. Simulation results

In this paper, we first need to validate that MicDR always achieves the energy reduction target of EDR program at all time slots, which is shown in Fig. 4(a). As shown in Fig. 5, MicDR always satisfies the energy reduction target; consequently, we can conclude that MicDR is effective for meeting the EDR’s goals. We can also find that MicDR maintains a small difference with the EDR target at each time slot. Consider that MicDR is designed to satisfy the requirement of an EDR program with minimum costs, this result shows that MicDR can always find a cost-efficient solution to avoid higher costs. Moreover, considering the robustness of MicDR, which will be discussed in Section 6.3, the power supply $g_e$ from the BES system is retained. Thus, to verify whether MicDR can incentivize tenants to save energy, we show the sources of energy reduction in Fig. 6. For each time slot, MicDR results in a slight energy reduction from the BES system, which means that the overall energy reduction from tenants alone nearly satisfies the EDR target. Thus, MicDR is a green mechanism, which incentivizes tenants to reduce energy consumption by improving energy efficiency rather than replacing the power grid with a BES system.

We use social costs to denote the total payments of the colocation operator to achieve the energy reduction target of an EDR program, and it is the main index to measure the mechanism performance in this work. Fig. 7 shows the social costs of different mechanisms. For all tenants, server utilization is calculated based on a normal distribution, where the expectation is the tenant workload. Thus, we obtain the social cost from the average of 150 experiments. We first compare the social cost of different incentive mechanisms in each time slot in Fig. 7(a); then, we show the overall social cost savings between MicDR and the other two mechanisms during the EDR program period in Fig. 7(b). It was shown in [12] that Truth-DR achieves a close-to-optimal performance when tenants are incentivized to reduce energy consumption by improving the energy efficiency of their local servers. By introducing the global incentive mechanism GiMec, LG-Mec obtained an even lower social cost because global resource management and the integration of multiple task types helps further optimize the energy efficiency of colocations. Compared with LG-Mec, MicDR reduces the cost of building a shared cloud using the server-sharing incentive mechanism SiMec. Thus, MicDR achieves a lower social cost than LG-Mec. Furthermore, in addition to the comparison among different mechanisms, we also compare the social cost between MicDr and a BES system in Fig. 7(c). Although using a BES system to achieve an EDR program is not influenced by the “uncoordinated relationship” issue, this approach uses extra power to replace the power from the grid rather than reducing energy consumption. Thus, the energy cost still accounts for a large proportion of the social cost for
the BES system. We use the lower bound of the unit cost to measure the social cost of BES systems, which is 150 $/MWh. However, MicDR can effectively reduce much of the energy consumption to satisfy the energy reduction target of EDR programs. As shown in Fig. 7(c), MicDR has a lower cost than BES systems, where the social cost of a BES system is more than twenty times that of MicDR. The results also show that although MicDR needs to pay for incentivizing tenants, it is a more cost-effective approach than replacing energy sources with a BES system.

To verify the features of MicDR proposed in Lemma 3 and Lemma 4, we show the average payments of the colocation operator for different tenants and resources in Fig. 8(a). The optimal cost is also added for comparison. By analyzing the social cost components, we find that only a small payment is paid for the power supply from a BES system as well as building the shared cloud. To echo the energy reduction shown in Fig. 6, MicDR can achieve a large energy reduction by incentivizing, and it only needs a small amount of power from a BES system. Moreover, by introducing the mechanism SiMec to incentivize tenants to share their idle servers to the shared cloud, the colocation operator can use idle servers for the shared cloud with lower costs. Accordingly, MicDR successfully achieves a lower social cost by reducing expenses related to extra energy reduction. Based on the average payments to tenants, we calculate the tenants’ average utilities and show them in Fig. 8(b). For 150 experiments, the utility of each tenant is always nonnegative. Thus, based on the payments calculated in MicDR, tenants do not lose anything when their bids are selected; thus, Lemma 4 is verified. To show the results more intuitively, we select an individual experiment from the 150 experiments randomly in Fig. 9, which shows that not all tenants’ bids are selected in one EDR program. By comparing Fig. 9(a) and (b), we find that tenants can obtain utilities only when tenants’ bids are selected in MicDR. Furthermore, Lemma 4 is
verified again that tenants never receive negative utility when they join the EDR program based on MicDR.

Finally, we analyze the influence of three important parameters for the performance of MicDR, including the unit cost of a BES system \( \tau \), the unit price of a VM instance \( \delta \) and the static server power \( P^s \). For each parameter, we calculate the average social cost of each mechanism during the EDR program.

- **Fig. 9** shows the payment and utility for tenants based on a randomly selected experiment.

- **Fig. 10** shows the average social cost comparison among different mechanisms and approaches when the unit cost \( \tau \) of the BES system changes from 150 $/MWh to 350 $/MWh. We first compare MicDR and LG-Mec. Because they only need a small power supply from a BES system, the change in \( \tau \) has a negligible effect on their average social cost. Thus, the social cost difference between MicDR and LG-Mec remains stable for different \( \tau \). However, because Truth-DR uses more BES power compared with MicDR, when \( \tau \) increases, the difference between MicDR and Truth-DR also increases. However, the rising tendency of the difference decreases, which indicates that Truth-DR can also reduce the power demand from a BES system for increasing \( \tau \). We also show the social cost ratio between a BES system and MicDR. Because the average social cost of MicDR is approximately constant as \( \tau \) changes, the ratio has an approximately linear correlation with \( \tau \). This finding means that MicDR can reduce costs even more for increasing \( \tau \). Therefore, we can conclude that (1) both MicDR and LG-Mec have more stable performance than Truth-DR as well as BES systems when \( \tau \) changes, and (2) although both have good performance, MicDR is better than LG-Mec.

- **Fig. 11** shows a comparison of the average social cost of the three mechanisms when the unit price of a VM instance \( \delta \) changes from 1.55 cents/h to 6.55 cents/h. The average social cost of MicDR increases almost linearly as \( \delta \) increases. For LG-Mec, as \( \delta \) changes from 1.55 to 4.55, the speed of increase in its average social cost diminishes, and when \( \delta > 4.55 \), its average social cost does not change. For Truth-DR, because it does not consider using public resources to improve the energy efficiency, when \( \delta \) changes, its average social cost remains constant. We can reach two conclusions from **Fig. 11**. First, compared with Truth-DR, when \( \delta \) decreases, MicDR and LG-Mec achieve better performances. In addition, no matter how much \( \delta \) increases, MicDR and LG-Mec always perform better than Truth-DR. Second, compared with LG-Mec, the average social cost savings of MicDR initially increases and then decreases, and it reaches its maximum when \( \delta = 3.55 \). When \( \delta < 3.55 \), as \( \delta \) increases, the influence of the total VM instance cost on the social cost becomes greater. Thus, the average social cost savings of MicDR increases as \( \delta \) rises when \( \delta < 3.55 \). However, when \( \delta > 3.55 \), an increase in \( \delta \) (and the consequent higher VM instance cost) causes fewer global bids to be selected. Thus, the average social cost savings of MicDR decreases as \( \delta \) goes up after \( \delta > 3.55 \).

- **Fig. 12** shows the average social cost comparison of three mechanisms as the static server power \( P^s \) changes from 0.15 kW to 0.4 kW. When \( P^s \) increases, the average social cost of the three mechanisms increases almost linearly; however, the average
of three sub-mechanisms (LiMec, GiMec and SiMec) that improve energy efficiency and resource utilization in colocations. LiMec and GiMec incentivize tenants to reduce their energy consumption, while SiMec is designed to create a server-sharing market to provide idle resources to GiMec based on a Stackelberg game. Then, for MiCDR, we formulated a MINLP cost minimization problem and developed a (1 + ε)-approximation algorithm to solve it. In addition, we also proved that MiCDR is both a truthful and feasible mechanism. We analyzed the theoretical performance of MiCDR and compared it with existing works. Our simulations in this study were based on widely used settings and real traces. By comparing MiCDR with different mechanisms, we showed that it incurs lower social costs for the same energy reduction target, thus validating its effectiveness. By analyzing the relationship between the energy reduction and the utility for each tenant, we also verified the lemmas proposed in this paper.

Acknowledgements

This work is supported partially by the National Natural Science Foundation of China under the grants 61520106005, 61761136014, and the National Key Research and Development Program of China under the grant 2017YFB1001001.

Appendix A. Proof of Lemma 1

For \( \beta \cdot \text{OPT}^f(2n, t, G') \geq E(t = 1, \ldots, 2n) = x_{e,f} + c(x_{e,f}^\prime) \&\& G' \in \{0, 1, \ldots, G_{\text{Max}}\} \), we use \( x(2n, t, G')_{\geq E}(x_{e,f}^\prime) \) for short) and \( y(2n, t, G')_{\geq E}(y_{e,f}^\prime) \) for short) to denote the corresponding solution vectors. Based on \( x_{e,f}^\prime \) and \( y_{e,f}^\prime \), for problem \( (P_1') \), it is easy to obtain \( g_{e,f} = \beta(e(x_{e,f}^\prime + y_{e,f}^\prime)) \geq E \) and \( g_{e,f}^\prime \leq G, \) where all the constraints of \( (P_2) \) are satisfied. Then, \( x_{e,f}^\prime \) and \( y_{e,f}^\prime \) are the feasible solution vectors for problem \( (P_2) \). When \( \beta \cdot \text{OPT}^f(2n, t, G') < E - g_{e,f} \), constraint \( (5) \) cannot be satisfied. Then, \( x_{e,f}^\prime \) and \( y_{e,f}^\prime \) are not feasible solution vectors for problem \( (P_2) \). Thus, Lemma 1 has been proved.

Appendix B. Proof of Lemma 2

Let \( \{(x_{e,f}^\text{opt}, y_{e,f}^\text{opt}) \mid i \in \mathcal{N}\} \) denote the optimal solution vector for \( (P_1) \). The corresponding \( g_{e,f} ^\text{opt} \) and \( G_{e,f} ^\text{opt} \) are denoted as \( g_{e,f} ^\text{opt} \) and \( G_{e,f} ^\text{opt} \), respectively. Because the constraints for \( (P_1) \) and \( (P_2) \) are the same, \( \{(x_{e,f}^\text{opt}, y_{e,f}^\text{opt}) \mid i \in \mathcal{N}\} \) is a feasible solution vector for \( (P_2) \), and \( \{(x_{i}(\text{min } G_{e,f} ^\text{opt}), \text{min } g_{e,f} ^\text{opt}), (y_{i}(\text{min } G_{e,f} ^\text{opt}), \text{min } g_{e,f} ^\text{opt}) \mid i \in \mathcal{N}\} \) is a feasible solution for \( (P_1) \). Thus, we can conclude the following:

\[
T_{\text{opt}} = F(G_{e,f} ^\text{opt}) + y_{e,f}^\text{opt} + \sum_{i \in \mathcal{N}} (d_{e,f} + c_{i}y_{e,f}^\text{opt}) + K(\text{min } G_{e,f} ^\text{opt}) \geq F(G_{e,f} ^\text{opt}) + y_{e,f}^\text{opt} + \sum_{i \in \mathcal{N}} (d_{i}(\text{min } G_{e,f} ^\text{opt}) + c_{i}y_{e,f}^\text{opt}) + K(\text{min } G_{e,f} ^\text{opt}) \geq F(\text{min } G_{e,f} ^\text{opt}) + t \cdot \text{min } g_{e,f} + \sum_{i \in \mathcal{N}} (d_{i}(\text{min } G_{e,f} ^\text{opt}) + c_{i}y_{e,f}^\text{opt}) = F(\text{min } G_{e,f} ^\text{opt}) + t \cdot \text{min } g_{e,f} \geq F(\text{min } G_{e,f} ^\text{opt}) + t \cdot \text{min } g_{e,f} + c_{i}y_{e,f}^\text{opt} \geq 2Kn, \]

7. Conclusions

Due to their high energy consumption, colocations play an important role in EDR programs. By analyzing the special management pattern of colocations, we showed that solving the “uncoordinated relationship” issue is the key to improving energy efficiency at colocations. In this paper, we proposed a market-oriented incentive mechanism called MiCDR, which is composed

Fig. 12. Comparison of the average social costs at different static server power values.
Because $K = \frac{e}{2T_2}$, we can obtain
t_{\min}^D \leq T_{\text{opt}} + \epsilon \cdot T_1 \leq (1 + \epsilon)T_{\text{opt}}.

Thus, we have proved that $t_{\min}^D$ is the $(1 + \epsilon)$-approximation solution of $(P_1)$.

Appendix C. Proof of Lemma 3

The utility of bid $b_i \in \mathcal{D}$ can be expressed as the difference between its market price and its cost. We use $u_{\text{true}}^i$ to denote the utility when $b_i$ declares the authentic cost, $h_{\text{true}}^i$, and $u_{\text{false}}^i$ to denote the utility when $b_i$ declares a false cost, $h_{\text{false}}^i$. Let $\Delta u_i = u_{\text{false}}^i - u_{\text{true}}^i$ denote the difference between $u_{\text{false}}^i$ and $u_{\text{true}}^i$. Based on the self-interest principle, the false cost must be greater than the authentic cost, i.e., $h_{\text{true}}^i < h_{\text{false}}^i$.

We consider two cases. First, $b_i$ is not selected when it declares a false cost. Thus, $u_{\text{false}}^i = 0$. Because $u_{\text{true}}^i \geq 0$, $\Delta u_i = u_{\text{false}}^i - u_{\text{true}}^i \leq 0$. Accordingly, in this case, Lemma 3 is true. Second, consider the situation that $b_i$ is selected when it declares a false cost. Thus, the cost when $b_i$ is selected is no more than the cost when $b_i$ is not selected, i.e., $V_{\mathcal{D} \setminus \{b_i\}}^E - F(G_{\mathcal{D} \setminus \{b_i\}}) - h_{\text{true}}^i \leq V_{\mathcal{D} \setminus \{b_i\}}^E - F(G_{\mathcal{D} \setminus \{b_i\}}) - h_{\text{false}}^i$. Then, when $b_i$ declares a cost $c_{\text{false}}^i$, the cost when selecting $b_i$ is less than the cost when $b_i$ is not selected, given as

$$V_{\mathcal{D} \setminus \{b_i\}}^E - F(G_{\mathcal{D} \setminus \{b_i\}}) - h_{\text{false}}^i < V_{\mathcal{D} \setminus \{b_i\}}^E - F(G_{\mathcal{D} \setminus \{b_i\}}) - h_{\text{true}}^i$$

Thus, $b_i$ is selected when it declares a cost $c_{\text{true}}^i$. Based on Eq. (9), we can get $u_{\text{false}}^i = V_{\mathcal{D} \setminus \{b_i\}}^E - V_{\mathcal{D} \setminus \{b_i\}}^E - h_{\text{true}}^i$ and $u_{\text{false}}^i = V_{\mathcal{D} \setminus \{b_i\}}^E - V_{\mathcal{D} \setminus \{b_i\}}^E - h_{\text{true}}^i$. Accordingly, $\Delta u_i$ is

$$\Delta u_i = u_{\text{false}}^i - u_{\text{true}}^i = 0. \tag{10}$$

Based on Eq. (10), $\Delta u_i$ is zero. Thus, for bid $b_i \in \mathcal{D}$, a tenant cannot obtain higher utility by declaring a false cost.

Most importantly, Lemma 3 is proved.

Appendix D. Proof of Lemma 4

Based on Lemma 3, we know that to obtain higher utility, all tenants will declare their costs authentically (i.e., $h_i = h_{\text{true}}^i$). The utility of bid $b_i \in \mathcal{D}$ can be expressed as $u_i(b_i)$, given by

$$u_i(b_i) = V_{\mathcal{D} \setminus \{b_i\}}^E - V_{\mathcal{D} \setminus \{b_i\}}^E - h_i.$$ 

When $b_i$ is selected, based on Algorithm 3, we can obtain

$$V_{\mathcal{D} \setminus \{b_i\}}^E \leq V_{\mathcal{D} \setminus \{b_i\}}^E,$$

and we can also obtain

$$V_{\mathcal{D} \setminus \{b_i\}}^E - F(G_{\mathcal{D} \setminus \{b_i\}}) - h_i.$$ 

Because $F(G_{\mathcal{D} \setminus \{b_i\}}) \geq F(G_{\mathcal{D} \setminus \{b_i\}} - h_i)$, we find that

$$V_{\mathcal{D} \setminus \{b_i\}}^E - h_i \geq V_{\mathcal{D} \setminus \{b_i\}}^E.$$

Thus, we obtain

$$V_{\mathcal{D} \setminus \{b_i\}}^E \geq V_{\mathcal{D} \setminus \{b_i\}}^E - h_i,$$

which means that

$$u_i(b_i) = V_{\mathcal{D} \setminus \{b_i\}}^E - V_{\mathcal{D} \setminus \{b_i\}}^E - h_i \geq 0.$$

Therefore, Lemma 4 is proved.

Appendix E. Algorithm 1

Algorithm 1. State initialization for $(P_3^*)$.

1. Initialize $T_{\text{true}} = \min_{d \in \mathcal{D}} \{\sum_{i \in N} \xi_i (d, c_i + c_{\text{true}}^i) + \tau \max \{E - \sum_{i \in N} \xi_i e_i + s_i, 0\}\}$
2. $T_{\text{false}} = \min_{d \in \mathcal{D}} \{\xi(d, c_{\text{false}}^i) + k = \frac{e}{2}\}$
3. Let $T = [\frac{1}{2} T_{\text{true}} + \frac{1}{2} T_{\text{false}}] + 2n + \sum_{i \in N} \xi_i (d_{\text{true}}^i + c_{\text{true}}^i)$. Then, for $i = 1, \ldots, n$, $k = 1, \ldots, N$,
4. Initialize $\text{OPT}^p(k, t, G^i)$, $\text{OPT}^p(k = 0, t = 0, G^i = 0) = 0$, $\text{OPT}^p(t \geq 0 | G^i < 0) = \infty$, $\text{OPT}^p(k = 0, t = 0, G^i = 0) = 0$.

Appendix F. Algorithm 2

Algorithm 2. Optimal solution set for $(P_3^*)$.

1. for $t = 0$ to $2n + \sum_{i \in N} \sum_{i \in N} (d_{\text{true}}^i + c_{\text{true}}^i)$ do
2. for $k = 1$ to $2n$ do
3. for $G^i = 0$ to $G^i$ do
4. if $k = n$ then
5. if $\text{OPT}^p(k - 1, t - d_{\text{true}}^i, G^i + c_{\text{true}}^i)$ then
6. $\text{OPT}^p(k, t, G^i) = \text{OPT}^p(k - 1, t - d_{\text{true}}^i, G^i + c_{\text{true}}^i)$
7. $\text{OPT}^p(k, t, G^i) = \text{OPT}^p(k - 1, t, G^i)$
8. $\text{OPT}^p(k, t, G^i)$
9. $\text{OPT}^p(k, t, G^i) = \text{OPT}^p(k - 1, t, G^i)$
10. $\text{OPT}^p(k, t, G^i) = \text{OPT}^p(k - 1, t, G^i)$
11. $\text{OPT}^p(k, t, G^i)$
12. $\text{OPT}^p(k, t, G^i)$
13. $\text{OPT}^p(k, t, G^i)$
14. $\text{OPT}^p(k, t, G^i)$
15. $\text{OPT}^p(k, t, G^i)$
16. $\text{OPT}^p(k, t, G^i)$
17. $\text{OPT}^p(k, t, G^i)$
18. $\text{OPT}^p(k, t, G^i)$
19. $\text{OPT}^p(k, t, G^i)$
20. $\text{OPT}^p(k, t, G^i)$
21. $\text{OPT}^p(k, t, G^i)$
22. $\text{OPT}^p(k, t, G^i)$
23. $\text{OPT}^p(k, t, G^i)$
24. $\text{OPT}^p(k, t, G^i)$
25. $\text{OPT}^p(k, t, G^i)$
26. $\text{OPT}^p(k, t, G^i)$
27. Return $\text{OPT}^p(2n, t, G^i) \in \{0, 1, \ldots, 2n + \sum_{i \in N} (d_{\text{true}}^i + c_{\text{true}}^i)\}$ & $G^i \in \{0, 1, \ldots, G^i\}.$

Appendix G. Algorithm 3

Algorithm 3. Optimal solution for $(P_2)$

Input: $\text{OPT}^p(2n, t, G^i)$

1. for $t = 0, 1, \ldots, 2n + \sum_{i \in N} (d_{\text{true}}^i + c_{\text{true}}^i)$ do
2. $G^i = E$
3. for $t = [\frac{e}{2}]$ to $2n + \sum_{i \in N} (d_{\text{true}}^i + c_{\text{true}}^i)$ do
4. while $G^i < 0$ & $\text{OPT}^p(2n, t, G^i) \geq E - g_{\text{true}}$ do


\[ x_t(G_t', g_t) = x'_t(2n, t, G_t') \]
\[ y_r(G_r', g_r) = y'_r(2n, t, G_r') \]
\[ g_r = g_t = 1 \]
\[ \text{end while} \]
\[ \text{if } g_r < \text{then} \]
\[ \text{Break} \]
\[ \text{end if} \]
\[ \text{end for} \]
\[ \min g_r', \min g_r = \arg\min_{g_r'} \{ F(G_r') + x_r + \sum_{i=1}^{n} (d(x_i(G_r', g_r) + c_i(y_i(G_r', g_r))) \}
\[ \min g_r' \}
\[ \text{end for} \]

References


