Energy Efficiency in Colocation Data Centers: A Joint Incentive Mechanism Approach

Youshi Wang∗†, Fa Zhang‡, Shaolei Ren§, Fangming Liu¶, Rui Wang∥, and Zhiyong Liu†∗∗

∗University of Chinese Academy of Sciences, China
†Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), China
‡Key Laboratory of Intelligent Information Processing, ICT, CAS, China
§University of California, Riverside, USA
¶Huazhong University of Science and Technology, China
∥Hong Kong University of Science and Technology, China
∗∗State Key Laboratory for Computer Architecture, ICT, CAS, China

Email: {wangyoushi, zhangfa}@ict.ac.cn, renshaolei@gmail.com, fmliu@hust.edu.cn, rwangae@connect.ust.hk, and zyliu@ict.ac.cn

Abstract—Colocation data centers (or colocations, for short) are important participants in emergency demand response (DR) programs. One key challenge in colocations is that tenants control their own servers, thus, may not coordinate to reduce their power consumption. In this paper, we propose a joint truthful incentive mechanism Co-Colo to encourage tenants joining EDR programs, which includes a local optimization mechanism (LocalOpt) and a global optimization mechanism (GlobalOpt). In LocalOpt, tenants are motivated to improve the energy efficiency locally. In GlobalOpt, tenants can request some public server resources to improve the energy efficiency. By jointly considering the two mechanisms, Co-Colo effectively reduces the energy-saving cost. A $(1 + \epsilon)$-approximation algorithm is proposed to obtain the asymptotic optimal energy-saving scheme. We also consider a special case when the public resources are sufficient, and design a 2-approximation algorithm. Furthermore, the robustness of the proposed algorithms are proved. Trace-driven simulations verify the effectiveness and feasibility of Co-Colo.

I. INTRODUCTION

Large-scale data centers are power-hungry and the power demand of them is flexible [1]. Thus, data centers can participate in demand response (DR) programs, especially in emergency demand response (EDR) programs [2]. EDR is widely adopted to improve the fragile power infrastructure. When some emergency events (e.g., extreme weathers) happen, EDR providers inform all participants about a fixed energy-saving target [3], and then the participants need to reduce the energy consumption and achieve the energy-saving target.

One important kind of data centers, called colocation data centers (or colocations, for short), develops rapidly in recent years which accounts for 37.3% of total data centers [4]. Colocations help tenants build their private data centers by providing professional infrastructure and service, and are often located in metropolitan areas [5]. Due to the dense population, high energy demand is incurred and the energy is frequently insufficient in the metropolitan areas. Thus, it is necessary for colocations to participate in EDR programs to avoid energy shortage and improve the stability of the power grid [6].

For achieving colocations’ EDR, we focus on how to improve the energy efficiency of colocations. For traditional data centers, energy efficiency technologies have been widely investigated, which include server resource virtualization [7], traffic engineering [8] and energy-efficient data center network (DCN) [9]. In these works, all the facilities, e.g., servers and infrastructure, are fully controlled by the data center operators. However, in colocations, the servers are managed by the tenants, which cannot be fully controlled by the colocation operators. Therefore, these works, which focus on the energy efficiency techniques of traditional data centers, are not feasible in colocations.

The special management pattern is considered in [10][11], and called “uncoordinated relationship” issue, which includes two aspects. Firstly, tenants lack of coordination with the colocation operator to save energy. For some retail tenants, the energy is paid up in advance based on their peak demand. Thus, if there is no benefits to the tenants, they have no incentive to reduce the energy consumption. For some wholesale tenants, they are charged based on their actual energy consumption, which ensures that tenants prefer to use energy as needed for saving cost. However, the tenants may not reduce the energy consumption when the EDR happens. Thus, it is important to incentivize tenants to coordinate with the colocation operator to save energy. Secondly, tenants lack of coordination with each other. By incentivizing the coordination among tenants, the colocation operator can make the optimization decision from a global view rather than relying on the local optimization of each tenant. Thus, how to design an incentive mechanism to encourage tenants to coordinate with each other is also a crucial problem.

There have been some works considering the energy efficiency issue of colocations. In [5], the “split incentive” issue in colocations was first considered, and an incentive mechanism iCODE was proposed to incentivize tenants to join the EDR programs. Some following works focused on how to improve the incentive mechanism, e.g. Truth-DR [11] and...
A joint DR was discussed including economic DR and emergency DR in [3]. A novel thermal-aware and cost efficient mechanism TECH was proposed in [13]. 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 works, the tenants work independently without coordination with each other, which cannot achieve good resource utilization and energy efficiency.

To solve the coordination issue among tenants, a novel framework was proposed in [10]. The framework incentivizes tenants to improve the utilization using some public resources. However, two key problems were not solved in the framework. One is that how to ensure the truthfulness of tenants’ information including the energy-saving targets and cost when they request some public server resources to migrate their workload. Moreover, it was assumed that the public server resources can satisfy all the requests. Considering the limited public resources, the assumption may not be practical.

A. Contributions

Different from existing techniques, we design a mechanism that not only encourages the coordination between the tenants and the colocation operator, but also encourages the coordination among different tenants in order to further improve the resource utilization as well as reduce the energy consumption. We summarize the major contributions as follows:

- To solve the “uncoordinated relationship” issue in colocations, we propose a joint incentive mechanism, called Co-Colo (Coordinated Colocation). Meanwhile, we introduce the working principle of Co-Colo, and also show its advantages in terms of the energy efficiency and the resource utilization.
- By discussing whether the public resources are sufficient to satisfy tenants’ total resource demands, Co-Colo is formulated as two different mathematical problems. For these two problems, we develop a \((1 + \epsilon)\)-approximation algorithm and a 2-approximation greedy algorithm, respectively. Moreover, we explain the robustness of the developed algorithms.
- For ensuring the truthfulness of the mechanism, we introduce the Vickrey-Clarke-Groves (VCG) theory into Co-Colo. We explain the feasibility of Co-Colo, and prove the truthfulness of Co-Colo.
- We validate the efficiency of the proposed mechanism and algorithms by simulations based on real workload traces. It is shown that significant energy efficiency improvement in colocations can be achieved by Co-Colo.

II. System Model

The “uncoordinated relationship” issue causes that the traditional energy efficiency optimization methods are infeasible in colocations. To improve the energy efficiency, it is necessary to design effective incentive mechanisms. Thus, we propose a joint incentive mechanism Co-Colo. When the EDR happens, we first encourage tenants to optimize the energy efficiency of their private servers independently. Because tenants can only operate their own servers, the sub-mechanism is called the local optimization mechanism (LocalOpt). Secondly, tenants are allowed to request some public server resources to migrate some workload to further improve the energy efficiency. The migrated workload can be globally optimized by the colocation operator, thus, the sub-mechanism is called the global optimization mechanism (GlobalOpt). In colocations, it is feasible to provide some public resources to tenants. These public resources include some colocations’ standby servers and cloud resources (i.e., some cloud providers have deployed servers in colocations such as Amazon and Google). The framework of Co-Colo is shown in Fig. 1, where we assume that there are \(n\) tenants. Then, by discussing whether the public resources are sufficient to satisfy the demand of tenants, Co-Colo is formulated as two mathematical problems.

A. Mechanism design

In this part, we will introduce LocalOpt and GlobalOpt in detail, and show the advantages of Co-Colo.

1) LocalOpt: This mechanism is used to incentivize tenants to reduce energy consumption by optimizing their own servers’ efficiency during the EDR period. In LocalOpt, tenants join the EDR program by submitting biddings which include their energy-saving targets and cost. In this mechanism, tenants can be regarded as sellers and the operator is the buyer. Thus, it is a typical auction pattern called the reverse auction [5]. For tenant \(i\), the energy-saving target and the declared cost are denoted as \(e_i\) and \(d_i\). Therefore, in LocalOpt, tenant \(i\)’s bidding can be expressed as \(b_i^p = (e_i, d_i)\). Then, we use \(\mathcal{B} = \{b_1^p, b_2^p, \ldots, b_n^p\}\) to donate the set of \(b_i^p (i \in [1, n])\).

2) GlobalOpt: In this mechanism, some public server resources are provided and tenants can request them based on demands. In GlobalOpt, tenants may further optimize their own energy efficiency. For example, during the EDR period when a task reaches and only needs one VM instance, keeping or turning on a server will cause low utilization. However, if the task is migrated to the public resources, the tenants can keep a high utilization. Thus, GlobalOpt can help tenants improve the utilization of the tenants. Moreover, when tenants migrate some tasks to the public resources, the colocation operator can globally optimize them to achieve higher utilization than the local optimization. Thus, GlobalOpt can also improve the global utilization of colocations.

In GlobalOpt, the capacity of the public resources is measured based on its total number of VM instances \(G\). For
tenant \( i \), \( g_i \) is used to denote the number of requested VM instances. Tenant \( i \)'s cost includes two parts. The first part is the energy-saving cost, which is denoted as \( c_i \). The second part is the cost of requesting VM instances \( f_i^{\text{cost}} \), which can be calculated by a cost function \( f_i(g_i) \). Meanwhile, \( s_i \) is used to denote tenant \( i \)'s energy-saving target. Therefore, tenant \( i \)'s bidding can be expressed as \( b_i^* = (s_i, c_i + f_i^{\text{cost}}, g_i) \). Then, we use \( B_i^* = \{b_1^*, b_2^*, ..., b_n^*\} \) to denote the set of \( b_i^*(i \in [1,n]) \).

3) Co-Colo: The incentive mechanisms LocalOpt and GlobalOpt solve the “uncoordinated relationship” issue from different views. By combining the two mechanisms, we propose a joint incentive mechanism Co-Colo. The objective of LocalOpt is to minimize the total cost while satisfying the energy-saving constraint. In GlobalOpt, an additional factor is considered which is the limited public resources. Accordingly, the colocation operator needs to select some of the tenants to satisfy their resource requests, which directly influences the total energy-saving and cost. Based on the function \( f_i(g_i) \), we can map the VM instance demands to a part of tenants’ cost \( f_i^{\text{cost}} \). Then, the public resource allocation problem can be translated to a cost optimization problem. Thus, the optimization objective of GlobalOpt and LocalOpt can be unified, which means that it is feasible and meaningful to combine these two mechanisms.

Co-Colo has two main advantages compared with two independent incentive mechanisms. Firstly, we show the cost-effectiveness of Co-Colo. In LocalOpt, tenants improve the utilization by integrating the local workload, and then turn off idle servers for energy reduction. However, there may still remain some low utilization servers. Thus, in GlobalOpt, we allow the tenants to request a few public VM instances to turn off them. Accordingly, we can avoid the energy waste which is caused by the low utilization of servers. By combining two mechanisms, tenants can make biddings based own their local information, and the operator can make a decision by taking some biddings based on its benefit and considering the local as well as public resources utilization. Secondly, Co-Colo can better improve the utilization. This is because most tenants in colocations are small and medium companies, and their workload types are relatively single. However, the same type of workloads usually has the same main resource (i.e., the main resource of a compute-intensive workload is the CPU resources). This causes the low utilization of other resources except the main resource. Based on Co-Colo, we can integrate different types of workloads from all tenants in the public resources, and improve the utilization of all resources. Accordingly, Co-Colo can effectively improve the utilization of colocations.

B. Mathematical models

In Co-Colo, our objective is to achieve the EDR energy-saving requirement with the minimal cost. Meanwhile, \( E \ (E \in \mathbb{Q}^+) \) is used to denote the constraint of the amount of the energy reduction. Besides, we also consider the limited number of VM instances \( G \ (G \in \mathbb{N}^+) \) in the public resources. As described in the previous section, tenants can bid for some public resource for their own benefits, while they can also bid for turning off their own servers and receive some rewards from the operator. In Co-Colo, we will combine them in a unified mathematical model. For each tenant, two biddings are submitted to LocalOpt and GlobalOpt, respectively. It may occur that it is better to accept only one bidding from one tenant. Thus, in our models, two biddings from one tenant can be selected independently. The mathematical problem of Co-Colo can be formulated as:

\[
(P_1) \quad \min \sum_{i=1}^{n} d_i x_i + (c_i + f_i^{\text{cost}}) y_i
\]

s.t.

\[
\sum_{i=1}^{n} e_i x_i + s_i y_i \geq E
\]

\[
\sum_{i=1}^{n} g_i y_i \leq G
\]

\[
x_i, y_i = \{0, 1\}
\]

where \( x_i \) and \( y_i \) are binary variables indicating whether tenant \( i \)'s bidding is selected. By solving (P1), we can obtain a set of \( (x_i, y_i)(i \in [1,n]) \), which can minimize the total cost when all constraints are satisfied. \( (P_1) \) is a min-knapsack problem. It is more complex than the classical min-knapsack problem because of an additional constraint \((P_1)-(3)\). Compared with the classical problem, obtaining a feasible solution of \( (P_1) \) with polynomial time cannot be achieved, because the process of obtaining a feasible solution is still an NP-hard problem.

Except \( (P_1) \), we also consider a simplified model \( (P_2) \). As stated earlier, not only the colocation operator’s standby servers are a part of the public resources, but also some cloud resources can be provided to the public resources. Thus, when the colocation operator has enough resources, the constraint \((P_1)-(3)\) can be ignored. \( (P_2) \) is a classical min-knapsack problem and is formulated as below:

\[
(P_2) \quad \min \sum_{i=1}^{n} d_i x_i + (c_i + f_i^{\text{cost}}) y_i
\]

s.t.

\[
\sum_{i=1}^{n} e_i x_i + s_i y_i \geq E
\]

\[
x_i, y_i = \{0, 1\}
\]

III. ALGORITHMS DESIGN

We design different algorithms to solve \( (P_1) \) and \( (P_2) \), respectively. Meanwhile, we also analyse their theoretical performance, and explain that the developed algorithms are robust, which means that the solution obtained by our algorithms is no worse than the solutions obtained when any bidding is deleted.

A. A \((1+\epsilon)\)-approximation algorithm for \( (P_1) \)

\( (P_1) \) is a min-knapsack problem. It is NP-hard, and has an additional constraint compared with the classical min-knapsack problem, which means that \( (P_1) \) is a more complex problem. Thus, an algorithm, Algorithm 1 is developed to solve \( (P_1) \). Meanwhile, for considering the robustness guarantee, we design Algorithm 2 based on Algorithm 1. We use \( T_{\text{opt}} \) to denote the theoretical optimal cost of \( (P_1) \), and assume that \( T_i \) and \( T_u \) can satisfy \( T_i \leq T_{\text{opt}} \leq T_u \). Assume that \( (x'_i, y'_i)(i \in [1,n]) \) is a feasible solution of \( (P_1) \), \( T_u \) can be expressed as \( T_u = \sum_{i=1}^{n} (d_i x'_i + (c_i + f_i^{\text{cost}}) y'_i) \). Note that if it is difficult to find a feasible solution, let \( T_u = \sum_{i=1}^{n} (d_i + (c_i + f_i^{\text{cost}})) \). Then, \( T_i \)
is expressed as \( T_i = \min \{ d_i, c_i + f_i^c \} \). We define
\[
K = \frac{\varepsilon T_i}{2n},
\]
where \( \varepsilon \) is a parameter which is related to the approximation ratio of Algorithm 1. Let \( d_i' = [d_i/K] \) and \( c_i' = [(c_i + f_i^c)/K] \). Correspondingly, \( b_i^p' = (e_i, d_i') \) and \( b_i^e' = (s_i, c_i', g_i) \). Then, we formulate the dual problem of \((P_1)\) as \((P_1')\):
\[
(P_1') \quad \begin{array}{ll}
\min & \sum_{i=1}^{n} c_i x_i + s_i y_i \\
st. & \sum_{i=1}^{n} d_i' x_i + c_i' y_i \leq T' \\
& \sum_{i=1}^{n} g_i y_i \leq G \\
& x_i, y_i = \{0, 1\} 
\end{array}
\]
where \( T' \in \{ \left\lceil \frac{T_i}{K} \right\rceil, \left\lceil \frac{T_i}{K} \right\rceil + 1, ..., 2n \} \).

To solve \((P_1')\), we adopt the dynamic programming (DP) approach. At each state \((k, t, G')\), where \( k \in \{0, 1, ..., n\} \), \( t \in \{0, 1, ..., T'\} \) and \( G' \in \{0, 1, ..., G\} \), we solve the following sub-problem of \((P_1')\):
\[
(P_1''') \quad \begin{array}{ll}
\max & \sum_{i=1}^{k} e_i x_i \\
st. & \sum_{i=1}^{n} d_i' x_i + c_i' y_i \leq t \\
& \sum_{i=1}^{n} g_i y_i \leq G' \\
& x_i, y_i = \{0, 1\} 
\end{array}
\]

The corresponding optimal value of state \((k, t, G')\) is denoted as \( OPT''(k, t, G')\). For the DP process, the initial state is shown as below:
\[
\begin{cases}
OPT'(k = 0, t \geq 0, G' \geq 0) = 0 \\
OPT'(k < 0) = -INF 
\end{cases}
\]
For the state transition equation, we should consider two different cases. Firstly, when \( k \leq n \), we can get:
\[
OPT'(k, t, G') = \max\{OPT''(k-1, t - d_k', G') + e_k, OPT''(k - 1, t, G')\}. 
\]
Secondly, when \( n < k \leq 2n \), we can get:
\[
OPT'(k, t, G') = \max\{OPT''(k-1, t - c_{k-n}', G' - g_{k-n}) + s_{k-n}, OPT''(k - 1, t, G')\}. 
\]
Then, a mark function is defined to record whether a state has been calculated in the DP process.
\[
Mark(k, t, G') = \begin{cases}
1 & \text{calculated} \\
0 & \text{otherwise}
\end{cases}
\]
For each \( t \in T' \), we can get an optimal solution vector \( \bar{A}_t \) of \((P_1')\). Based on \( \bar{A}_t \), we can get the corresponding optimal value \( OPT''(k, t, G') \). Considering the constraint \((P_1)-(2)\), if \( OPT''(k, t, G') \geq E \) is true if and only if \( t \geq t(I) \) \((t(I) \in T')\), we can get that \( \bar{A}_t \) is the optimal solution vector of \((P_1')\) on the bidding set \( I \).

Then, Lemma 1 shows that \( \bar{A} \) is also the feasible solution of \((P_1)\), where \( \bar{A} = \{(x_i', y_i') | i \in [1, n]\} \).

**Lemma 1.** If \((P_1)\) exists a feasible solution, it must be the feasible solution of \((P_1')\).

**Proof:** Assuming that \((P_1)\) has a feasible solution \((x_i', y_i') (i \in [1, n])\), we can get that \( \sum_{i=1}^{n} (c_i x_i + s_i y_i) \geq E \) and \( \sum_{i=1}^{n} g_i y_i \leq G \) are true. And because \( \sum_{i=1}^{n} (d_i' x_i + c_i' y_i) \in T' \), \((x_i', y_i')(i \in [1, n])\) is a feasible solution of \((P_1')\). \( \square \)

Next, we explain how to obtain \( K \) and prove that the approximation ratio of Algorithm 1 is \((1 + \varepsilon)\). Let \( T_{opt} = \sum_{i=1}^{n} (d_i x_i + c_i + f_i^c y_i) \). If \( T_{opt} < \sum_{i=1}^{n} (K(d_i x_i - 1) + K(c_i y_i - 1)) \), we can get that \( \sum_{i=1}^{n} (K(d_i x_i - 1) + K(c_i y_i - 1)) < \sum_{i=1}^{n} (K(d_i x_i + K(c_i y_i - 1)) \), which means that \((x_i', y_i')\) is better than \((x_i', y_i')\) for \((P_1')\). Because \((x_i', y_i')(n \in [1, n])\) is the optimal solution of \((P_1')\), \( T_{opt} < \sum_{i=1}^{n} (K(d_i x_i - 1) + K(c_i y_i - 1)) \) is false. Thus, we can get that
\[
T_{opt} \geq \sum_{i=1}^{n} (K(d_i x_i - 1) + K(c_i y_i - 1)) = K \sum_{i=1}^{n} (d_i x_i + c_i y_i) - 2Kn \geq \sum_{i=1}^{n} (d_i x_i + c_i + f_i^c y_i) - 2Kn. 
\]
Let \( 2Kn = \varepsilon T_i \), and then we can get that \( \sum_{i=1}^{n} (d_i x_i + c_i + f_i^c y_i) \leq T_{opt} + \varepsilon T_i \leq (1 + \varepsilon)T_{opt}. \) Thus, we get that \( K = \varepsilon T_i/(2n) \), and \((1 + \varepsilon)\) is the approximation ratio of Algorithm 1. Furthermore, because \( t_{opt} = \min\{t(I), t(I)\} \), \( t_{opt} \leq t(I) \) \((I \in [1, n])\), we can get that the approximation ratio of Algorithm 2 is also \((1 + \varepsilon)\).

Then, we explain the robustness of Algorithm 2. \( \{b_i^p\} \) denotes that \( b_i^p (1 \leq i \leq n) \) is deleted from \( I \), and \( \{b_i^e\} \) denotes that \( b_i^e (1 \leq i \leq n) \) is deleted from \( I \). According to Algorithm 2, we can get that \( t_{opt} \leq t(I) \), \( t_{opt} \leq t(I) \), and \( t_{opt} \leq t(I) \) are true. Thus, it can be obtained that the sub-optimal value, which is calculated by Algorithm 2, does not change when anyone bidding is deleted. Therefore, the robustness of Algorithm 2 is guaranteed.

Finally, by analysing Algorithm 2, the time complexity can be expressed as \( O(n \cdot (2n \cdot (2n + \sum_{i=1}^{n} (d_i x_i + c_i y_i))) \).

Because \( d_i' = [d_i/K] \) and \( c_i' = [(c_i + f_i^c)/K] \), we can get \( \sum_{i=1}^{n} (d_i x_i + c_i y_i) \leq [T_u/K] + 2n = [2nT_u/(\varepsilon T_i)] + 2n. \)
Replacing $\sum_{i=1}^{n} (d_i'x_i^* + c_i'y_i^*)$ with $[2nT_u/(\varepsilon T_1)] + 2n$, the time complexity can be simplified as $O(n^3 [T_u/(\varepsilon T_1)])$.

Algorithm 1 A $(1+\varepsilon)$-approximation algorithm for $(P_1)$

1. Initialize $T_u = \sum_{i=1}^{n} (d_i x_i^* + (c_i + f_{i^*}^\text{cost}))y_i^*$, $T_i = \min_{1 \leq i \leq n} \{d_i/c_i + f_{i^*}^\text{cost}\}$ and $K = \frac{c_i}{2\varepsilon}.
2. Let $d'_i = \lfloor d_i/K \rfloor$ and $c'_i = \lceil (c_i + f_{i^*}^\text{cost})/K \rceil.
3. Let $T' = \left\{ \left\lfloor \frac{T_u}{K} \right\rfloor \right\} + 1, \ldots, 2n + \sum_{i=1}^{n} (d'_i x_i^* + c'_i y_i^*)$.
4. Initialize $OPT'(k,t,G')$. $OPT'(k,0,t,0,G' \geq 0) = 0$ and $OPT'(t < 0||G' < 0) = -INF$.
5. for all $t \in T'$ do
6. for $k = 1 \to 2n$ do
7. if $k \leq n$ then
8. $OPT'(k,t,G') = \maxOPT'(k-1, t-d_k', G') + e_k, OPT'(k-1, t, G')$.
9. end if
10. if $k > n$ then
11. $OPT'(k,t,G') = \maxOPT'(k-1, t-e_k-n, G' - g_k-n) + s_k-n, OPT'(k-1, t, G')$.
12. end if
13. end for
14. for $t = t/2$ to $2n + \sum_{i=1}^{n} (d'_i x_i^* + c'_i y_i^*)$ do
15. when $t = t(I)$ ($t(I) \in T'$), $OPT'_1 \geq E$.
17. end for
18. Thus, $t(I)$ is the $(1+\varepsilon)$-approximation value of $(P_1)$ and the corresponding solution vector is denoted as $\mathcal{A}_{t(I)}$.

Algorithm 2 The robustness guarantee of Algorithm 1

1. Based on Algorithm 1, we can calculate $t(I \{b_i^\text{opt}\})$ and $t(I \{\{b_i^\text{opt}\}\}) (i \in [1,n])$.
2. Let $t_{opt} = \min \{t(I), t(I \{b_i^\text{opt}\}), t(I \{\{b_i^\text{opt}\}\})\}$, $t_{opt}(i \in [1,n])$.
3. Thus, we can get that $t_{opt}$ is the sub-optimal value of $(P_1)$.

The corresponding solution vector is denoted as $\mathcal{A}_{t_{opt}}$.

B. A 2-approximation greedy algorithm for $(P_2)$

When ignoring the constraint $G$, $(P_1)$ can be simplified as $(P_2)$, which is a classical min-knapkack problem. However, in our system model, an additional constraint is to guarantee the robustness of designed algorithms. Based on an existing algorithm $GR$ [14], which is described in Algorithm 3, a 2-approximation ratio can be obtained. However, Algorithm 3 cannot be used here for the robustness of the solution and the truthfulness of the mechanism. The reason is that in the process deciding a bid’s truthfulness, it cannot guarantee a correct decision by using Algorithm 3 for computing the minimum cost because of its approximation nature. Thus an algorithm, Algorithm 4, is developed to guarantee the robustness of the algorithm and the truthfulness of the mechanism.

Firstly, because the constraint $G$ is ignored in $(P_2)$, we do not consider the parameter $g_i$ in $b_i^\text{opt}$. Thus, $b_i^\text{opt}$ can be simplified as $b_i^\text{opt} = (s_i, c_i + f_{i^*}^\text{cost})(i \in [1,n])$. We use $(a_i,b_i)(i \in [1,n])$ to denote $(e_i, d_i)(i \in [1,n])$, and $(a_i, b_i)(i \in [n+1,2n])$ to denote $(s_i, c_i + f_{i^*}^\text{cost})(i \in [1,n])$. The detail steps of the algorithm is shown in Algorithm 4. Based on Algorithm 3, we can get a 2-approximation value $T_{\text{min}}^2(L,E)$ of $(P_2)$. We use $T^2$ to denote the sub-optimal value of $(P_2)$, which can be calculated by Algorithm 4. According to Algorithm 4, we can get that $T^2 \leq T_{\text{min}}^2(L,E)$ and $T^2 \leq T_{\text{min}}^2(L,E)(i \in [1,2n])$ are true. Thus, we can get two results: (1) The approximation ratio of Algorithm 4 is also 2, (2) The robustness of Algorithm 4 can be guaranteed.

Algorithm 3 A 2-approximation greedy algorithm for $(P_2)$

1. $T' = \{ (a_i,b_i) \mid 1 \leq i \leq 2n \}$
2. Sort all elements in $T'$ according to $b_i/a_i \leq b_2/a_2 \leq \ldots \leq b_n/a_n$.
3. Let $L = (a_1, a_2, \ldots, a_n)$, $L_i = L \backslash \{a_i\}$.
4. Let $k_1$ be the index for which $\sum_{i=1}^{k_1} a_i < E \leq \sum_{i=1}^{k_1+1} a_i$, then getting a solution $(a_1, a_2, \ldots, a_{k_1+1})$.
5. Let $S_1 = \{a_1, a_2, \ldots, a_{k_1}\}$, then $S_1 \cup \{a_{k_1+1}\}$ is also a solution.
6. If $\sum_{i=1}^{k_1} a_i + a_j \geq E$ when $j \in \{k_1+2, \ldots, k_2-1\}$, all $S_1 \cup \{a_j\}$ is solutions. Let $B_1 = (a_{k_1+1}, a_{k_1+2}, \ldots, a_{k_2-1})$.
7. Assume that $k_2$ is the first next index for which $\sum_{i=1}^{a_k} a_i + a_k < E$ and $k_3 (k_3 \geq k_2)$ is the index for which $\sum_{i=1}^{a_k} a_i + a_k < E \leq \sum_{i=1}^{a_k+1} a_i$, and set $S_2 = \{a_{k_2}, a_{k_2+1}, \ldots, a_{k_3}\}$, $B_2 = (a_{k_3+1}, a_{k_3+2}, \ldots, a_{k_4-1})$, then $S_1 \cup S_2 \cup \{a_{k_4+1}\}$ is also a solution.
8. Iterate 7 until all elements in $T'$ are traversed, and obtain the 2-approximation value $T_{\text{min}}^2(L,E)$ of $(P_2)$.

Algorithm 4 The robustness guarantee of Algorithm 3

1. Let $T^2 = \min \{T_{\text{min}}^2(L,E), T_{\text{min}}^2(L,E)\}(i \in [1,2n])$.
2. $T^2$ is the sub-optimal value of $(P_2)$.

Then, we discuss the time complexity of Algorithm 4. In Algorithm 3, the 2ed step is a quicksort, so its time complexity is $O(n \log n)$. From the 4th step to the 8th step, the time complexity of each step is $O(n)$. Thus, the time complexity of Algorithm 3 is $O(n \log n)$. In Algorithm 4, we need 2n times of iterative calculation for $T_{\text{min}}^2(L,E)(i \in [1,2n])$. Thus, the time complexity of Algorithm 4 can be expressed as $O(n \log n + 2n \cdot n)$, which can be simplified as $O(n^2)$.

IV. TRUTHFUL AUCTION MECHANISM

In this section, we analyse the feasibility and truthfulness of Co-Color based on the VCG theory. The VCG theory is a sealed-bid and truthful auction mechanism which can achieve a socially-optimal solution [15]. When adopting the VCG theory, tenants participate in the auction without knowing others’ biddings, and tenants’ actual benefits are decided by others’ biddings. Besides, it also can incentivize tenants to bid their true valuations for obtaining more benefits. Before analysing, three hypotheses are given as the precondition:
Tenants are rational people who know own preference and have clear understanding of their goals. Furthermore, they can make choice independently which means they are not influenced by others in their bidding process.

- Tenants always make rational choices, which means that random or experiential decisions do not exist when tenants make decisions in the auction.
- Self-interest principle, which means that tenants always participate in the auction for obtaining the maximal profit and do not pay attention to others.

Besides, in Section III, we have explained that the developed algorithms are robust. This feature guarantees that these algorithms can satisfy the policy of the VCG theory.

Then, for Co-Colo, we discuss its pricing strategy, and prove its feasibility and truthfulness. We use \( D \) to denote the bidding set of Co-Colo, which can be expressed as \( D = \{b_1, ..., b_{2m}\} \), where \( b_i = (m_i, h_i, \bar{g}_i) \). When \( i \in [1, n] \), let \( m_i = e_i, h_i = d_i \) and \( \bar{g}_i = 0 \). When \( i \in [n + 1, 2n] \), let \( m_i = s_i, h_i = c_i + \frac{m_i}{t_{\text{cost}}} \) and \( \bar{g}_i = g_i \).

Based on the general problem \((P_1)\), we define that \( V_{D}^{E,G} \) is the \((1 + \epsilon)\)-approximation solution obtained by Algorithm \( 2 \). \( D \setminus \{b_i\} \) denotes that a bidding \( b_i \) is deleted from \( D \), just as \( D \setminus \{b_i\} = \{b_1, ..., b_{i-1}, b_{i+1}, ..., b_n\} \). Based on \( D \setminus \{b_i\} \), \((P_1)\)'s \((1 + \epsilon)\)-approximation solution is denoted as \( V_{D \setminus \{b_i\}}^{E,G} \), which is obtained from Algorithm 1. Let \( p_1 = V_{D \setminus \{b_i\}}^{E,G} \) and \( p_2 = V_{D \setminus \{b_i\}}^{E-m_i,G-\bar{g}_i} \), then tenant \( i \)'s market-clearing price \( p_i \) can be derived as:

\[
p_i = p_1 - p_2 = V_{D \setminus \{b_i\}}^{E,G} - V_{D \setminus \{b_i\}}^{E-m_i,G-\bar{g}_i}.
\]

\(h_i^{\text{true}}\) is defined as the truthful cost of bidding \( b_i \), and \( u_i\) denotes the utility of tenant \( i \). To guarantee the feasibility of Co-Colo, all tenants' utility must be non-negative. Meanwhile, to guarantee the truthfulness, we need to ensure that it is impossible for any tenant to obtain higher utility by declaring a false cost. Then, we provide two lemmas to prove the feasibility and truthfulness of Co-Colo.

**Lemma 2.** In Co-Colo, if bidding \( b_i \) is selected, \( u_i \geq 0 \).

**Proof:** For tenant \( i \), the utility can be expressed as the difference between the market-clearing price and the truthful cost of bidding \( b_i \).

\[
u_i = p_i - h_i^{\text{true}} = V_{D \setminus \{b_i\}}^{E,G} - (V_{D \setminus \{b_i\}}^{E-m_i,G-\bar{g}_i} + h_i^{\text{true}}).
\]

When \( b_i \) is selected, by the algorithms, we can get

\[
V_{D}^{E,G} \leq V_{D \setminus \{b_i\}}^{E,G}.
\]

Meanwhile, because \( h_i^{\text{true}} \) is the truthful cost of tenant \( i \), it means that \( h_i^{\text{true}} \leq h_i \) is true. Thus, we can get

\[
V_{D}^{E,G} \geq (V_{D \setminus \{b_i\}}^{E-m_i,G-\bar{g}_i} + h_i^{\text{true}}).
\]

Thus, based on (23) and (24), we can get that \( u_i = V_{D}^{E,G} - (V_{D \setminus \{b_i\}}^{E-m_i,G-\bar{g}_i} + h_i^{\text{true}}) \geq 0 \). Thus, Lemma 2 is true.

---

**TABLE I. SIMULATION PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of servers for each tenant</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of tenants</td>
<td>6</td>
</tr>
<tr>
<td>Static power consumption of each server ( P_s )</td>
<td>0.15 kW</td>
</tr>
<tr>
<td>Dynamic power consumption of each server ( P_d )</td>
<td>0.1 kW</td>
</tr>
<tr>
<td>Electricity price ( p_e )</td>
<td>0.1 $/kWh</td>
</tr>
<tr>
<td>Energy-saving cost ( p_s )</td>
<td>5 \times 10^{-3} $/kWh</td>
</tr>
<tr>
<td>The average price of VM instance ( p_{vm} )</td>
<td>0.007 $</td>
</tr>
<tr>
<td>Physical machine</td>
<td>Dell PowerEdge R730</td>
</tr>
</tbody>
</table>

---

Fig. 2. 2(a) EDR energy reduction. 2(b) Workload traces.

**Lemma 3.** In Co-Colo, when tenant \( i \) declares a false cost \( h_i^{\text{false}} \), \( u_i \) does not increase.

**Proof:** Firstly, if \( b_i \) is not selected, tenant \( i \)'s utility is always zero whether the declared cost is true or false. Thus, in this case, Lemma 3 is true.

Secondly, if \( b_i \) is selected, we assume that \( u_i' \) is the utility when tenant \( i \) declares a false cost, so we can get

\[
\begin{align*}
 u_i' &= p_i - h_i^{\text{false}} \\
 &= V_{D \setminus \{b_i\}}^{E,G} - (V_{D \setminus \{b_i\}}^{E-m_i,G-\bar{g}_i} + h_i^{\text{false}}).
\end{align*}
\]

Then, \( \Delta u_i \) is defined as the difference between \( u_i \) and \( u_i' \), given as

\[
\Delta u_i = u_i' - u_i = h_i^{\text{true}} - h_i^{\text{false}}.
\]

According to the self-interest principle, the false cost cannot be less than the truthful cost, just as \( h_i^{\text{true}} \leq h_i^{\text{false}} \). Based on (26), \( \Delta u_i \) is a negative value, i.e., \( \Delta u_i < 0 \), which means that declaring a false cost cannot increase tenant \( i \)'s utility. Thus, Lemma 3 is true.

---

**V. EXPERIMENTS ANALYSIS AND PERFORMANCE EVALUATION**

In this section, we first describe the corresponding settings about the simulations, and explain the data sources used in the simulations. Based on the widely accepted settings [1][5][16] and the real traces, our simulations are more convincing. Secondly, combined with the theoretical analysis and the simulation results, we evaluate the performance of the proposed algorithms. Then, we evaluate the effectiveness and feasibility of Co-Colo based on the simulations.

**A. Settings**

Assume that a colocation has six tenants (denoted as Tenant #1, Tenant #2, ..., and Tenant #6). For each tenant, it has 10,000 homogeneous servers, and the static power \( P_s \) and the dynamic power \( P_d \) of one server are 0.1 kW and 0.15 kW.
Because we focus on tenants’ energy efficiency and cost, the PUE of the colocation would not affect our results. Thus, the PUE would be ignored in the simulations. Above all, the peak power of the colocation is 15 MW. Besides, we set the electricity price \( p_e \) as 0.1 $/kWh, which means that tenants pay 0.1 $ for 1 kWh. Meanwhile, we measure the energy-saving cost according to the server switching cost and the performance loss cost, which is denoted as \( p_{vm} \).

**Energy reduction targets:** The data of energy-saving targets comes from PJMs EDR on April 22, 2015 [17], and the data is scaled down to 15% of the colocation’s maximum power for avoiding affecting normal operation [18]. The energy-saving targets are shown in Fig. 2(a). There are eight events from 6 am to 13 pm, and each event lasted one hour.

**Workload:** The workload traces, which come from “MSR” and “Florida International University”, are from [1], and are shown in Fig. 2(b). The workload traces are divided into six sub-traces which are regraded as the workload of six tenants.

**Tenants’ energy-saving and cost:** In Co-Colo, we consider that tenants optimize the energy efficiency by turning off idle servers [16]. Meanwhile, we consider that different tenants have their own bidding parameters, which are used to distinguish tenants’ different expected cost. Tenant \( i \)'s bidding parameter is denoted as \( \alpha_i \). For LocalOpt, tenants need to optimize their local servers’ utilization by integrating workload and turning off idle servers. We use \( n_{i,s} \) to denote the number of turned off servers. Thus, the energy reduction of tenant \( i \) is \( e_i = n_{i,s} \cdot P^s \cdot T \), where \( T = 1 \) hour is one EDR period. For tenant \( i \), \( d_i \) includes two parts. Firstly, the energy charge, which is prepaid before the EDR, should be returned. Then, the switching cost also a part of \( d_i \). Thus, tenant \( i \)'s cost can be expressed as \( d_i = e_i \cdot p_e + n_{i,s} \cdot p_s \cdot \alpha_i \). For GlobalOpt, we assume that tenants request \( n_{i,vm} \) VM instances to migrate the workload of \( n_{i,C} \) servers, and the utilization of each server is denoted as \( u_{i,j} \). Then, the energy reduction of tenant \( i \) is \( s_i = \sum_{j=1}^{n_{i,C}} (u_{i,j} \cdot P^d + P^s) \). Compared with LocalOpt, the cost of tenant \( i \) in GlobalOpt has an additional cost of requesting VM instances. By analysing statistically seven different T2 instances from the Amazon EC2, we can obtain the average price \( p_{vm} \) of a VM instance. Thus, tenant \( i \)'s cost can be expressed as \( c_i = f_i^{cost} = s_i \cdot p_e + (n_{i,C} \cdot p_s + n_{i,vm} \cdot p_{vm}) \cdot \alpha_i \).

**B. Analysis and evaluation**

<table>
<thead>
<tr>
<th>TABLE II. Comparisons of overall performance among different algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximation ratio</td>
</tr>
<tr>
<td>Algorithm 2</td>
</tr>
<tr>
<td>Algorithm 4</td>
</tr>
<tr>
<td>Branch-Bound</td>
</tr>
</tbody>
</table>

1) Algorithm analysis: The theoretical performance of algorithms is summarized in Table II. By comparison, we find that the approximation ratio of 2 is better than Algorithm 4. However, when considering the time complexity, Algorithm 4 is better than Algorithm 2. Besides, Algorithm 2 is a more general algorithm than Algorithm 4. Thus, for the relatively simple problem \( (P_2) \), we use Algorithm 4 instead of Algorithm 2. Furthermore, except the above three features, we also show that Algorithm 2 and Algorithm 4 both are robust, which means that they can both satisfy the policy of the VCG theory.

The practical performance of algorithms is shown in Fig. 3 and Fig. 4. One control group is the branch and bound (b-b) algorithm, which can get the optimum value with the exponential time complexity. Because Algorithm 4 has the same approximation ratio as GR, Algorithm 4 can be regard as another control group. Firstly, we compare the optimal cost values of three algorithms. From Fig. 3(a), we can find that the performance of Algorithm 2 is close to the optimum value, and better than Algorithm 4. Comparing Fig. 3(a) with Fig. 3(b), we find that the optimal cost values of three algorithms have no obvious fluctuations. However, Algorithm 2 shows more obvious advantage than Algorithm 4 in Fig. 4. Comparing the social cost values, Algorithm 2 always keeps the same performance as the b-b algorithm in two different mechanisms. On the contrary, the performance of Algorithm 4 fluctuates obviously by comparing Fig. 4(a) with Fig. 4(b). It shows that Algorithm 2 has higher stability than Algorithm 4.

2) Effectiveness of Co-Colo: We verify the effectiveness of Co-Colo by comparing the optimal and social cost between two different mechanisms, because the cost is the main optimization objective for the colocation operator. There are two reasons about selecting LocalOpt as the control group rather...
than GlobalOpt. Firstly, considering the existing researches in colocations, LocalOpt is adopted more widely to incentivize tenants to optimize their own energy efficiency, which has been shown in related works. Secondly, we analyse the energy-saving cost of two mechanisms according to the system model shown in Section II-A. Compared with GlobalOpt, when $e_j = \delta_i$, we can get that $\delta_{i,p} < \delta_{i,C}$. Besides, GlobalOpt needs to consider an extra public resource cost $f_{i}^{op}$. Thus, the energy-saving cost of GlobalOpt is higher than LocalOpt, which means that LocalOpt is a better control group when considering the cost.

In Fig. 5, we compare the optimal cost and the social cost between Co-Colo and LocalOpt based on Algorithm 2. Fig. 5(a) shows that two mechanisms can get the similar optimal cost based on Algorithm 2. However, in Fig. 5(b), the difference of two mechanisms becomes obvious. The results explain that Co-Colo can reduce effectively the actual energy-saving expense, and are also a proof for the performance of Algorithm 2. Furthermore, Fig. 5(b) shows that Co-Colo can obtain better effect with the high energy-saving targets. For instance, Co-Colo has about 20% social cost improvement at 6 am, 10 am, 11 am and 12 am, and the value is about 5% at 8 am. At other time, two mechanisms have similar social cost. It is because LocalOpt has achieved the optimal social cost by synthesizing multiple factors, GlobalOpt is provisionally ignored. The results accord with our mechanism design.

3) Feasibility of Co-Colo: Fig. 6 shows tenants’ benefits when their biddings are selected in the EDR project. All tenants’ benefits are non-negative based on two algorithms. The results verify Lemma 2 proposed in Section IV. Furthermore, Fig. 6 explains that two proposed algorithms satisfy the policy of the VCG theory. Comparing Fig. 6(a) with Fig. 6(b), we also find that Algorithm 2 can reduce tenants’ benefits more effectively on the premise of non-negative benefits. Fig. 6 explains the feasibility of Co-Colo, and also shows that Algorithm 2 helps the colocation operator control effectively the additional cost, which is used to incentivize tenants for saving energy and guarantee the truthfulness of Co-Colo.

VI. Conclusion

Due to high energy consumption, colocations play an irreplaceable role in the EDR program. By analysing the special management pattern of colocations, we showed that solving the “uncoordinated relationship” issue is the key to improve the colocations’ energy efficiency. In this paper, we firstly proposed a joint incentive mechanism Co-Colo. We showed that it can better improve the energy efficiency and resource utilization in colocations compared with LocalOpt and GlobalOpt. Secondly, considering whether the public resources are sufficient to satisfy tenants’ total demands, Co-Colo was formulated as two problems. We designed a $(1+\epsilon)$-approximation algorithm and a 2-approximation algorithm to solve them, respectively. Meanwhile, we also explained the robustness of proposed algorithms. Then, we proved the feasibility and the truthfulness of Co-Colo. Finally, based on the real traces, we evaluated the effectiveness of Co-Colo and the performance of developed algorithms.

REFERENCES


