A Truthful FPTAS Mechanism for Emergency Demand Response in Colocation Data Centers

Jianhai Chen*, Deshi Ye*, Shouling Ji*†, Qinming He*, Yang Xiang‡, Zhenguang Liu§
* Institute of Cyberspace Research and College of Computer Science and Technology, Zhejiang University, Hangzhou 310027.
{chenjh919, yedeshi, hqm, sji}@zju.edu.cn
† Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China.
‡Swinburne Research, Swinburne University of Technology, Victoria 3122 Australia.
yxiang@swin.edu.au
§Dept. of Computer Science, National University of Singapore, Computing 1, 13 Computing Drive Singapore 117417.
liuzhenguang2008@gmail.com

***Abstract*—Demand response (DR) is a vital means of electricity market in maintaining power grid reliability, sustainability and stability. DR can enable consumers (e.g. data centers) to reduce their electricity consumption when the supply of electricity is a shortage. The consumers will be rewarded if they reduce or shift some of their energy usage during peak hours. Aiming at solving the efficiency of DR, in this paper, we present MEDR, a mechanism on emergency DR in colocation data center. First, we formalize the MEDR problem and propose a dynamic programming to solve the optimization version of the problem. We then design a deterministic mechanism to solve the MEDR. We prove that our mechanism is truthful and it is an FPTAS, i.e., it can be approximated within $1+\epsilon$ for any given $\epsilon > 0$, while the running time of our mechanism is polynomial in the number of tenants n and $1/\epsilon$. Furthermore, we also give an auction system covering the efficient FPTAS algorithm as bidding decision program for DR. Finally, we choose a real dataset to build a large number of simulation datasets in performance evaluation. The results show that our mechanism outperforms near-optimal and high utility demonstrate the effectiveness of our work.**

***Index Terms*—Emergency Demand Response; Knapsack Problem; Mechanism Design; Colocation Data Center; Auction**

I. Introduction

Demand response (DR) programs are widely adopted in many colocation data centers for improving the efficiency of power grids [1], [2]. It attempts to adjust the demand for power instead of adjusting the supply. In today's power grid, DR is a technique for regulating the energy consumption over time, which is one of the major reliability impacts for smart grids [3]. On the consumer side, DR is commonly utilized as a powerful tool for employing flexibility of using electricity in response to supply-demand conditions [4]. In smart grid market when electricity price rises or the system reliability are threatened, the electricity supplier will firstly deliver the notice of direct compensation, of inductively reducing power load or signal of power price rise to electricity consumers. The consumers will change their intrinsic power using mode so as to meet the demand of electricity supply, reducing or passing a special period of time of power load, ensuring the stability of the power grid and restraining the rise of electricity price. In addition, DR is an important tool of demand-side management [5] which refers to the way that countries use policy measures to guide power users to reduce electricity at peak time, use electricity in low valley, improve power supply efficiency and optimize the usage of electricity. When it comes to the emergency demand of using electricity, the demand-side management will be required to start an immediate response or incentive mechanism of the user report electricity, and declare the amount of electricity and the price of electricity.

However, there are few efficient mechanisms and systems to support efficient power management in current power grid and many colocation data centers, leading to high power cost and low efficiency [5]. For example, to reduce peak demand in a power grid, the DR is usually implemented manually by sending signals to large consumers, such as data centers. Besides, it is worthy to note that colocation of data centers is quite popular now. According to the website[1], there are $3,775$ colocation data centers from 112 countries. A colocation is the third-party leased placement that provides physical homes for many data centers, and provides lots of services such as fast Internet, stable power supply and cooling. Though colocation of data centers provided a nice solution for those enterprise tenants, it consumed huge electricity. As pointed out in [6], 91 billion kilowatt-hours of electricity was consumed in U.S. in 2013, and it emitted around 97 million metric tons of carbon pollution in that year. On the other hand, it is possible to close or migrate some tasks in a large data center such that some computing servers can be shutdown. This makes possible for data centers to be participant in DR. In case of emergency (for example, earthquake or extreme bad weather) or reaching the capacity of a grid, it requires to implement the DR.

Therefore, to improve the efficiency, an incentive reverse auction mechanism is employed to motivate power users to participate in DR activities. The basic process of the auction is as follows. The electricity DR and electricity quota are issued by the special power management department. After the power users receive the signal, the electricity price can be reported in

[1]Data collected from http://www.datacentermap.com/ in Jan 7, 2016.

the form of bidding. Then the bidding system runs a bidding mechanism to choose the group of users, in which users will perform the restricted electricity consumption behavior in the DR and achieve corresponding compensations.

In this work, we aim to design an efficient mechanism to solve the DR problem called MEDR. Actually, there are several challenges in solving MEDR. It is worth to note that we could apply the VCG mechanism [7]–[9] to MEDR problem, and a deterministic truthful mechanism was obtained. However, the optimization problem of MEDR is NP-hard, since it generalizes the min-knapsack problem. On the other hand, approximation algorithms for MEDR might not be truthful. To the best of our knowledge, Zhang et al. [4] was the first one to study approximated truthful mechanisms for MEDR problem. They provided a 2-approximated mechanism with truthful in expectation. The main technique of their work is to present a 2-approximated algorithm, and then turn the approximation algorithm into a mechanism with truthful in expectation, while keeping the approximation ratio of 2. The framework of their work is based on a convex decomposition technique [10], which will transfer an approximation algorithm into a truthful randomized mechanism.

Our main contribution is to propose a deterministic truthful mechanism with FPTAS approximated. To the best of our knowledge, we are not aware of other deterministic approximated truthful mechanisms. The main technique of our work is to design a monotone algorithm based on the framework of Archer and Tardös [11], which is the key idea of designing deterministic truthful mechanism for one parameter. In detail, we first design a dynamic programming for the optimization version of our studied problem, and then applying some rounding technique such that our design dynamic programming satisfies the monotone property. Furthermore, in our mechanism, both the cost c_i and the size e_i of each agent i are private information. We also implement all the algorithms in a reverse auction system and a simulation tool for performance evaluation. The system tool can be used for colocation data centers or can be extended to be adopted to some other applications such as auctions in smart grid. Extensive experiments are presented to evaluate the effectiveness of our method. Empirical results show that our mechanism achieves nearly optimal solution.

The rest of this paper is organized as follows. In Section II, we state the mechanism design problem called MEDR. In Section III, we propose a dynamic programming to solve the optimization version of the studied problem. Then we address an FPTAS deterministic truthful mechanism. In Section V, we implement a reverse auction system tool and all the algorithms. Extensive simulation experiments are taken to evaluate the effectiveness of our method. The related works are presented in Section VI, and concluding remarks are given in Section VII.

II. Problem statement

In this section we address the statement of a mechanism design problem on DR for colocation data center.

We study a mechanism design problem MEDR that arises in data center demand response. There are n tenants in a colocation data center. Each tenant $i \in \{1, 2, \ldots, n\}$ subscribes a certain amount of power supply from the colocation operator. In the event of Emergency Demand Response (EDR), the colocation operator is required to reduce W amount of energy. Given power-based contracts, tenants may not have incentive to participate in EDR unless they are awarded. Even if some tenants are interested in EDR, their reduction may not reach the reduction target W. In case of not reaching the target, the colocation operator can use backup energy storage (BES) to fulfill the shortage of EDR target. Let y be the amount of grid-power demand reduction due to the usage of BES, and α be the cost of BES usage per kWh.

Each tenant i submits a bid with two parameters (s_i, b_i), where s_i is the amount of planned energy reduction and b_i is the claimed cost due to such a reduction. However, each tenant i has its own true type (e_i, c_i), where c_i is the cost due to a reduction of e_i energy. The value e_i and c_i are only known to the tenant i. Moreover, each tenant is *single-minded* [12] such that each tenant is restricted to one single bid. Every tenant has freedom to choose participation in this EDR or not. If a tenant is not willing to participate in this EDR, we can suppose its bid is $(0, 0)$. Let $B = \{(s_1, b_1), \ldots, (s_n, b_n)\}$ be the set of bids by the n tenants. Based on this bidding B, the colocation operator will pay money $P_i(B)$ to each tenant i to encourage their participate in this EDR. Let $U_i(B) = P_i(B) - c_i$ be the utility of tenant i according to the biddings of B. Clearly each tenant i attempts to maximize his/her utility. According to [4], the power consumption in colocation data center consists of both the energy consumption of tenants and also consumption of management such as cooling. There is a ratio called Power Usage Effectiveness (PUE) γ between the total energy consumption to the energy consumed by tenants, which typically ranges from 1.1 to 2.0.

A tenant is a *winner* if her/his bidding is successful. Let N be the set of winners. To meet the energy reduction target W, we require that $y + \gamma \sum_{i \in N} s_i \geq W$. The social cost of the colocation operator is $\alpha y + \sum_{i \in N} P_i(B)$. The social cost of tenants is $\sum_{i \in N}(c_i - P_i(B))$. Thus, the total social cost is equivalent to aggregate tenant cost due to energy reduction plus the operator's cost for using BES, i.e., $\alpha y + \sum_{i \in N} c_i$. The goal of the mechanism design is to minimize the total social cost, meanwhile no tenant can benefit by proposing a false bidding. The optimization version of this problem can be formulated as an integer programming. Let $x_i = 1$ if tenant i is a winner, i.e. $i \in N$, otherwise $x_i = 0$.

$$min \quad \alpha y + \textstyle\sum_{i=1}^{n} x_i c_i \tag{1}$$

subject to

$$y + \gamma \textstyle\sum_{i=1}^{n} x_i s_i \geq W \tag{2}$$

$$x_i = \{1, 0\}, \forall i \in \{1, 2, \ldots, n\} \tag{3}$$

The studied MEDR problem is closely related to knapsack auction problems. According to objective functions, we define

two types of mechanism design for knapsack problems. In *max-knapsack* problem, each agent has a private valuation for having his/her objective in the knapsack. The problem is to find an allocation of the agents without exceeding the capacity of the knapsack as so to maximize the sum of each agent's value.In *min-knapsack* problem, each agent has a private cost for having his/her item in the knapsack. The problem is to find an allocation to cover the knapsack, while the sum of agents' cost is minimized.

For any instance I, we define by $\mathcal{C}(\mathcal{M}(I))$ the *social cost* of the mechanism $\mathcal{M}$, which is the total costs of tenants plus the operator's cost for using BES. A mechanism $\mathcal{M}$ is said to be ρ-approximated if $\mathcal{C}(\mathcal{M}(I)) \leq \rho \cdot \mathcal{C}(OPT(I))$, where OPT is an optimal algorithm.

Let $B_{-i} = \{B_1, \ldots, B_{i-1}, B_{i+1}, \ldots, B_n\}$ be the bids except tenant i's bid.

Definition 1: (Truthfulness): A mechanism $\mathcal{M}$ consisting of an allocation function $\mathcal{A}$ and a payment function $\mathcal{P}$ is truthful (or strategy-proof) if for every tenant i with the true cost c_i cannot increase his/her utility by declaring any other cost (s_i, b_i) regardless of every bidding of other agents B_{-i}, i.e., it satisfies

$$U_i((e_i, c_i), B_{-i}) \geq U_i((s_i, b_i), B_{-i}).$$

This definition implies that truthful reporting is a dominant strategy for every tenant.

Definition 2: (Individual rationality): A mechanism $\mathcal{M}$ is said to be individual rationality if every agent always obtains non-negative utility with bidding of the true cost, i.e., $U_i((e_i, c_i), B_{-i}) \geq 0$ for any i and any B_{-i}.

III. Approximated Truthful Mechanism

In this section, we will address an approximated truthful mechanism. We present a dynamic programming to solve the optimization version of our problem MEDR optimally, and then we explore a deterministic truthful mechanism while it is arbitrarily approximated for any given $\epsilon > 0$.

A. Dynamic Programming Model

Our dynamic programming requires to solve the min-knapsack problem as a subroutine. The min-knapsack problem consists in finding a subset of items, where each item i has a cost c_i and a size s_i, with the minimum cost such that the sum of their sizes is at least as large as a specified capacity. Based on the idea of the max-knapsack problem [13], Tauhidul [14] gave a dynamic programming for the min-knapsack problem. We adopt this dynamic programming in (4) [14] as a subroutine in the following.

Let $S(i, c)$ denote a subset of $\{1, \ldots, i\}$ whose cost is exactly c and whose total size is maximized. Let $A(i, c)$ be the size of $S(i, c)$ ($A(i, c) = 0$ if no such set exists). The recursive formula of the dynamic programming is given in (4). In this formula $A(i, c)$ gives a tabular of an optimal value for each subproblem (i, c).

$$A(i,c) = \begin{cases} \max\{A(i-1,c), s_i + A(i-1, c-c_i)\}, \\ \quad if\ c_i \leq c \\ A(i-1,c), \quad otherwise \end{cases} \tag{4}$$

In the following, we design the dynamic programming for our MEDR problem based on the recursive function (4). The Algorithm 1 (Algorithm $DOPT(I)$) gives the details of the dynamic programming for our problem MEDR.

Algorithm 1: Algorithm $DOPT(I)$: Dynamic Programming for MEDR

Input: The set of tenants I, and demand capacity W.

Run the dynamic programming based on the formula (4) for the input I, and obtain $A(i, c)$ for each (i, c), where $1 \leq i \leq n$ and $0 \leq c \leq \sum_i c_i$;

for *each* (i, c) **do**

  if $\gamma A(i, c) < W$ **then**

    $y(i, c) = \alpha(W - \gamma A(i, c)) + c$

  else

    $y(i, c) = c$

Output: Return $\min_{(i,c)} y(i, c)$

Theorem 1: The dynamic programming $DOPT(I)$ produces an optimal solution for any instance of tenants I with demand request W, and unit cost of BES α. The running time is pseudo-polynomial, which is $O(n^2 c_{\max})$, where $c_{\max}$ is the largest cost.

Proof. Any optimal solution consists of two parts, one is covered by BES, and another is covered by items from I. Let p, q be the cost due to tenants I and BES, respectively. Let $c_{\max} = \max_i c_i$ be the largest cost among all tenants.

In the dynamic programming we iterate all possible (i, c), where $c \in C = \{0, 1, 2, \ldots, nc_{\max}\}$. Any cost due to tenants I is in C, hence $p \in C$. Note that the dynamic programming $A(i, p)$ provides the maximize size whose cost is exactly p. If $\gamma A(n, p)$ is less than W, and we require at least $W - \gamma A(n, p)$ BES to cover the knapsack in the optimal. Therefore, $p + q \geq \alpha(W - \gamma A(n, p)) + p$. If $\gamma A(n, p) \geq W$, then $q = 0$. These two cases are both covered in the dynamic programming, which implies that the dynamic programming outputs an optimal solution.

The running time of dynamic programming is $O(n^2 c_{\max})$, since $i \leq n$, and $c \leq nc_{\max}$, and the running time is bounded by the iterative function of (i, c). □

B. Monotone FPTAS Model

Motivated by the truthful mechanism for max-knapsack problem [15], we will propose a deterministic truthful mechanism. To keep the truthful property, the idea of our mechanism is to give a monotone algorithm. To obtain an FPTAS, we need to design a monotone algorithm whose approximation ratio is arbitrarily close to 1. The detailed algorithm is given in Algorithm 3, which iteratively calls a subroutine Algorithm 2

(Algorithm $A_r(k, I)$). The motivation of Algorithm 2 is to keep the truthful, in which the rounding of each item is independent on the bidding of all tenants.

Algorithm 2: Algorithm $A_r(k, I)$

Input: Given parameter k, and the instance $I = (s_1, c_1), \ldots, (s_n, c_n)$.

Let $a_k = \frac{\epsilon 2^k}{n+1}$;

Let $T(k)$ be the subset of items with cost at most of 2^k; We construct a new instance I' based on $T(k)$, which is identical to $T(k)$, but the cost of each item c' is given as below.

for $i \in T(k)$ **do**

  $c'_i = \lfloor \frac{c_i}{a_k} \rfloor$

Run the dynamic programming $DOPT(I')$for the items in $T(k)$ with cost c'_i, and obtain $A(i, c')$;

for *each* (i, c') **do**

  if $\gamma A(i, c') < W$ **then**

    $y(i, c') = \lfloor \frac{\alpha(W - \gamma A(i,c'))}{a_k} \rfloor + c'$

  else

    $y(i, c') = c'$

Output: Return $\min_{(i,c')} y(i, c')$

Algorithm 3: Monotone FPTAS A_{FPTAS}

Input: Given $\epsilon > 0$, and the instance I.

Let $best \leftarrow \infty$, and $c_{\max} = \max_i c_i$.

for $k \leftarrow 1$ ***to*** $\log c_{\max}$ **do**

  $S'(k) \leftarrow A_r(k, I)$; /* call Algorithm 2 (Algorithm $A_r(k, I)$) */

  if $S'(k) < best$ **then**

    $best \leftarrow S'(k)$

    $\bar{S} \leftarrow$ the subset items that contained in the solution of $S'(k)$

Output: $\bar{S}$, and use BES $W - \gamma \sum_{i \in \bar{S}} s_i$

Lemma 1: For any $\epsilon > 0$, Algorithm A_{FPTAS} has approximation ratio of $1 + \epsilon$, and its running time is polynomial in $1/\epsilon, n, \log c_{max}$.

Proof. Let c_q be the largest cost among the items in an optimal algorithm to cover the knapsack. Define k^*, such that

$$2^{k^*-1} < c_q \leq 2^{k^*}.$$

Denote O^* to be the subset of items in the optimal solution. Let y^* be the size BES used in the optimal solution. Let $O^*(R) = O^* \bigcup \{R\}$, where R is a virtual item with size y^* and cost αy^*. Let OPT be the cost of the optimal solution. We have $OPT \geq c_q$.

Note that in $T(k^*)$ as denoted in the algorithm $A_r(k, I)$, we have $O^* \subseteq T(k^*)$. Let $\bar{S}$ be the subset of items returned by the algorithm $A_r(k, I)$ with k^* as the parameter, and let $(\bar{i}, \bar{c})$ be the pair of values that reaches the minimum of $A_r(k, I)$.

Let O' be the subset of items with costs rounded by 2^{k^*} from O^*. Let R' be a virtual item with size y^*/a_{k^*}.

Let ALG be the final cost incurred by the algorithm A_{FPTAS}, we can use the following inequalities to approximate the cost by the algorithm with the optimal solution.

$$\begin{aligned}
ALG &= \sum_{i \in \bar{S}} c_i + \max(\alpha(W - \gamma A(\bar{i}, \bar{c})), 0) \\
&\leq \sum_{i \in \bar{S}} c_i + \max(\lfloor \frac{\alpha(W - \gamma A(\bar{i}, \bar{c}))}{a^*_k} \rfloor, 0) a_{k^*} + a_{k^*} \\
&\leq \sum_{i \in \bar{S}} (c'_i \cdot a_{k^*} + a_{k^*}) + \\
&\quad \max(\lfloor \frac{\alpha(W - \gamma A(\bar{i}, \bar{c}))}{a^*_k} \rfloor, 0) \cdot a_{k^*} + a_{k^*} \\
&\leq \sum_{i \in \bar{S}} c'_i \cdot a_{k^*} + \max(\lfloor \frac{\alpha(W - \gamma A(\bar{i}, \bar{c}))}{a^*_k} \rfloor, 0) \cdot a_{k^*} \\
&\quad + (n+1) a_{k^*} \\
&\leq \sum_{i \in O' \bigcup \{R'\}} c'_i \cdot a_{k^*} + (n+1) a_{k^*} \\
&\leq \sum_{i \in O^* \bigcup \{R\}} c_i + (n+1) a_{k^*} \\
&\leq OPT + \epsilon 2^{k^*} \\
&\leq (1 + 2\epsilon) OPT.
\end{aligned}$$

The running time is $poly(1/\epsilon, n, \log c_{\max})$. In algorithm $A_r(k, I)$, the largest cost of $T(k)$ is 2^k, the running time of dynamic programming is $O(n^3/\epsilon)$. The total running time of A_{FPTAS} is $O(\frac{1}{\epsilon} n^3 \log c_{max})$. □

1) Monotone: A declaration $B'_i = (s'_i, b'_i)$ is said to be a *higher declaration* than the bidding $B_i = (s_i, b_i)$ if $s'_i \geq s_i$ and $b'_i \leq b_i$, i.e. $B_i \preceq B'_i$. A bid (s_i, b_i) is said to be a *winner declaration* if this item is selected in the knapsack.

Definition 3: (Monotone) We say that an algorithm A is monotone if, for any bidder (s_i, b_i) is a winning declaration then any higher declaration also wins.

Bitonic was introduced by Mu'Alem and Nisan [16] for maximize problems, such as multi-unit auction, and it was generalized by Briest, Krysta, and Vöcking [15].

In this work, we apply the technique of bitonic to the minimize problems.

Definition 4: (Bitonic) Given a function $f : \mathcal{A}^n \rightarrow$, a monotone algorithm A is bitonic with respect to the function f if for any agent i, the following hold:

1) If $i \in A(B)$, then $f(A(B_i, B_{-i})) \geq f(A(B'_i, B_{-i}))$ for any $B_i \preceq B'_i$.
2) If $i \notin A(B)$, then $f(A(B_i, B_{-i})) \geq f(A(B'_i, B_{-i}))$ for any $B'_i \preceq B_i$.

Intuitively, a monotone algorithm A is bitonic with respect to f if f is a monotone non-decreasing function of each agent's valuation while she is not selected for the solution, but becomes monotone non-increasing after she is selected for the solution. In this work, the function f is the objective

function, i.e., the social welfare. The bitonic is indeed required to guarantee the monotone for compositions of algorithms.

Algorithm 4: $MIN(A_1, A_2)$ Operator

Input: Bidding B

Run the algorithm A_1 and A_2;
Let $sw_{A_1}(B)$ and $sw_{A_2}(B)$ be the social welfare of Algorithm A_1 and A_2, respectively.
if $sw_{A_1}(B) \leq sw_{A_2}(B)$ **then**
  return $A_1(B)$;
else
  return $A_2(B)$.

Lemma 2: Let A_1 and A_2 be two monotone bitonic allocation algorithms. Then, $M = MIN(A_1, A_2)$ is a monotone bitonic allocation algorithm.

Proof. This can be easily extended from the proof of the Theorem 2 in [16], which was designed for the MAX operator. □

Lemma 3: Algorithm $A_r(k, I)$ is monotone and bitonic with respect to the objective function.

Proof. Algorithm $A_r(k, I)$ returns an optimal solution, if an agent reports a higher bidder, then the optimal algorithm will accept this item too. Suppose an agent i was not selected, and any lower declaration B'_i, if this item was accepted then the objective function shall be smaller, otherwise the objective remains, and hence the objective function is non-increasing for any lower bidders. Thus the property of bitonic follows. □

Lemma 4: Algorithm A_{FPTAS} is monotone and bitonic with respect to the objective function.

Proof. The lemma follows immediately according to Lemma 2 and Lemma 3. □

2) Payment:

Definition 5: (Critical payment) Let algorithm A be a monotone algorithm, if we fix the declaration B_{-i}, and then for any agent i and fixed bidding s_i, there exists a unique cost θ_i^A, called *critical payment*, such that $\forall b_i \leq \theta_i^A$, b_i is a winning declaration, and $\forall b_i > \theta_i^A$ is a losing declaration.

To calculate the critical value for any agent j, we fix the other agents' bids, and then use a binary search on interval $[b_j, \max_j b_j]$ and repeatedly run the allocation algorithm A to check whether the agent j is selected.

Definition 6: The payment p^A associated with the monotone allocation algorithm A that is based on the critical value is defined by $p_j^A = \theta_j^A$ if agent j wins with allocation $Alloc_i(B) = s_i$, and $p_j^A = 0$ otherwise.

A mechanism $M_A = (A, p^A)$ is *normalized* , if its payment p^A is defined as in Definition 6, i.e. agents that are not selected pay 0. We say that algorithm A is *exact* if $Alloc_i(B) = s_i$ or $Alloc_i(B) = \emptyset$ for each declaration (s_i, b_i).

In this work, we only consider a limited type of agent called *single-minded*, the cost function ∞ if $Alloc_i(B) > s_i$ and c_i otherwise. That will force each agent does not over bidding his/her size if allocation algorithm is an exact algorithm.

Theorem 2: [15] Let A be a monotone and exact algorithm for some minimization problems and single-minded agents. Then mechanism $M_A = (A, p^A)$ is truthful.

Proof. In [15], they gave the detailed proof for utilitarian problems, and thus it holds for MAX-knapsack problem. Moreover, in their paper, it was shown that the proof is valid for minimization problems, such as the reverse single-minded multi-unit auction problem, which is equivalent to the minimum knapsack problem. □

Algorithm 5: Algorithm $P^{\mathcal{A}}(B)$

Input: The bidding B of all tenants, and the allocation algorithm $\mathcal{A}$

for $i \leftarrow 1$ ***to*** n **do**
  Let $z_i = 1$ if the ith item is selected in the knapsack by the allocation problem $\mathcal{A}(B)$, and 0 otherwise.
  Let $h = \alpha\gamma s_i$ and $l = b_i$.
  while $h - l \geq 1$ **do**
    $b'_i = (h + l)/2$;
    $z_i = \mathcal{A}(B_{-i}, (s_i, b'_i))$;
    if $z_i == 1$ **then**
      $l = b'_i$
    else
      $h = b'_i$
  $P_i(B) \leftarrow l$.

Output: The payment $P_i(B)$ for each agent i.

Theorem 3: The mechanism $M_{A_{FPTAS}} = (A_{FPTAS}, p^{A_{FPTAS}})$ is truthful and it is an FPTAS mechanism, i.e. its approximation ratio is $1 + \epsilon$ for any given $\epsilon > 0$, and the total running time of the mechanism is polynomial in $1/\epsilon, n, \log c_{max}$.

Proof. Algorithm $P^{\mathcal{A}}(B)$ (Algorithm 5) is a critical payment, Algorithm A_{FPTAS} is an exact algorithm, and bitonic with respect to the objective function according to Lemma 4. Thus the mechanism $M_{A_{FPTAS}}$ is truthful followed by Theorem 2. The FPTAS is achieved in Lemma 1.

□

IV. Implementation

In this section, we provide a reverse auction system (RAS) to solve the MEDR problem and implement all the algorithms for DR on colocation data centers.

A. Reverse Auction System

The architecture of the RAS is shown in Fig. 1. The whole system framework consists of two parts, namely, client and server. The client refers to users including the bidding tenants and the colocation data center operator. The server is a MEDR auction system, which is implemented and run as a service. This service program involves three primary function modules,

that is, *auction controller (AC)*, *bidding decision (BD)* and *power controller (PC)*, respectively.

The *AC* module is responsible for interaction with client users, including receiving the client tenants' bidding application, invoking bidding decision (BD) to make a bidding decision, and returning back the bidding decision results to client users. The *BD* module implements all the MEDR algorithms presented in Section III, including the dynamic programming algorithm A_{DOPT}, the monotone FPTAS algorithm A_{FPTAS} and the payment algorithm $A_{Payment}$. All the algorithms are developed by C/C++ language, and are integrated together as an independent service program. The *PC* module takes charge of executing power supply policies for balancing power using according to the bidding results. In particular, the RAS runs in a colocation data center for all tenants and the colocation operator.

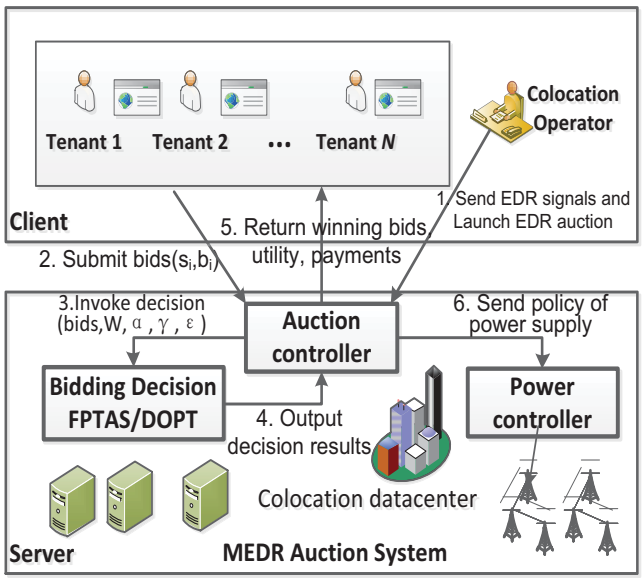


Fig. 1. Architecture of MEDR auction system

B. Reverse Auction Process

As shown in Fig. 1, the basis process of reverse auction is as follows. At first, upon arrival of DR or EDR signals in client, the colocation operator (Auctioneer) will firstly launch an auction in RAS and send the EDR signal to tenants. The tenants who are willing to participate then submit their own bids with the size of power and costs to the RAS server.

Then, the **AC** receives the bids from client tenant users, and then ask for BD to process the bidding decision when the system needs to make decision of bidding. In bidding decision, BD invokes FPTAS to make a decision. Before invoking, BD needs to input the bidding data and the value of decision algorithm required parameters, such as EDR W, α, γ, and ϵ, etc. BD will input all data from the AC and calls a decision algorithm to get bidding decision results. After running the decision algorithms, it outputs the bidding decision results to the AC.

Further, the AC returns back the bidding decision results to the client. The results include the winning bids, and payments. In the end, not only the winning tenants will achieve compensation rewards from the colocation data center operator, but also the new policy of power supply will be submitted to the PC and be executed for the DR.

V. Performance Evaluation

In this section, we firstly generate simulation datasets. We then do holistic simulation experiments to evaluate the efficiency of FPTAS mechanism which further validates the effectiveness of our RAS system performance.

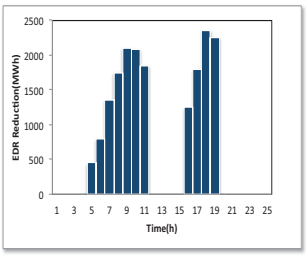
A. Simulation Datasets

We consider a colocation data center, with $N(N = 9)$ participating tenants (denoted as Tenant #1, Tenant #2, ...,Tenant #9). Each tenant i has $m_i(= 100,000)$ homogeneous servers. The power of each server includes $150W$ computing power and $d_0 = 100W$ idle power. That is to say, if a server is turned off, then we can obtain 100W power. If a server is running a workload, it needs $150 + 100 = 250W$ power.

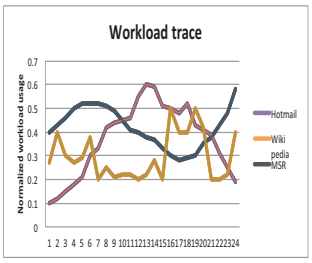
For simplicity, we produce simulation datasets according to the datasets used in [4], including total EDR energy reduction by PJM on January 7, 2014 and normalized workload traces which are collected from [17] ("Hotmail" and "MSR") and [18] ("Wikipedia"), as shown in Fig. 2(a) and Fig. 2(b). The EDR energy reduction data consists of eleven event numbers. Each number represents an energy reduction of demand response arise in one hour period of time. The eleven time periods are made up of a range in hour from 5 to 11 and from 16 to 19. Moreover, we get the specific data from the two figures and list in Table I in detail. Like Zhang [4], we further generate EDR energy reduction targets as 15% of the total EDR, which lists in the W column of Table I.

The tenants' bids data are generated randomly with consideration of the workload traces and some data center factors, such as the general cost of a server or whole data center and the energy price in market. We duplicate each workload for all tenants with randomness up to 20%. For the total 9 tenants, we firstly generate 9 random numbers r_i, where $i = 1, 2, \ldots, 9$, between 0.01 and 0.20. In our simulation, the random numbers are $0.11, 0.06, 0.02, 0.13, 0.05, 0.01, 0.15, 0.16, 0.02$. Then we duplicate the workload for three tenants by Hotmail, three by MSR and three by Wikipedia, respectively. We assume that each size of the workload denotes a ratio of the total number of servers which is in running status. If we turn off some servers then we can slash energy consumption. The total energy reduction by tenant i is $s_i = n_i.d_{0,i}.T$, where $T = 1$ hour in simulation.

We conclude the size of tenants' bid by a formula: $s_i = workloads * M * r_i * d_0/1,000,000(MW)$, where $M = 100,000$ and $d_0 = 100$. The final range of size s is between $10 \sim 80$. The workloads use data shown in Table I. Besides, according to Zhang [4], tenants can reasonably save $6.7 \sim 13.3$ cents/kWh (depending on electricity price) power when they house servers in their data centers. Equivalently,



(a) Total EDR energy reduction by PJMon January 7, 2014.



(b) Normalized workloads in distinct time.

Fig. 2. The source dataset for bidding decision simulation.

tenants can save 67 $\sim$ 133 \$/MWh. Hence we produce several random numbers rb between 67\$ and 133\$ for each tenant as a bid power price which tenants are willing to take part in the auction activity. The ith tenant bid is concluded by formula $b_i = s_i * rb_i$. All produced simulation data are in Table II. Each tenant bid Bi has two numbers s_i, b_i, which denote a size of energy reduction and cost for supplying its responding size of power, respectively.

TABLE I
EDR TARGET & WORKLOAD

Time(h)	EDR	W(15%EDR)	Hotmail	MSR	Wikipedia
5	450	68	0.21	0.52	0.29
6	800	120	0.30	0.52	0.20
7	1,350	203	0.33	0.52	0.20
8	1,750	263	0.42	0.51	0.25
9	2,100	315	0.44	0.49	0.21
10	2,080	312	0.45	0.45	0.22
11	1,850	278	0.46	0.41	0.22
16	1,250	188	0.50	0.30	0.50
17	1,800	270	0.48	0.28	0.40
18	2,350	353	0.52	0.29	0.40
19	2,250	338	0.43	0.30	0.50

The PUE of colocation γ is set to 1.6 (typical for colocation), and the default cost for using BES, α is considered 150\$/MWh which we will vary later depending on the BES energy source.

B. Experiment and Result Analysis

To evaluate the performance of the MEDR method, we consider the approximation ratio of the FPTAS mechanism, tenants' utility and social cost reductions. For simplification, we prepare all the simulation data and related parameter data of algorithms together into a file which is used as input for the bidding decision program. During the test running process, our bidding decision program will read the data in a batch, process all the algorithms execution, and then output all results to a file for analysis.

1) Approximation Ratios: Our FPTAS mechanism can achieve $1 + \epsilon$ approximation ratio. Though FPTAS ensures the running time is polynomial in n, and $1/\epsilon$. To evaluate the efficiency of the FPTAS mechanism in practice, we use approximation ratio metric which is concluded by the ratio between the FPTAS-approximation algorithm and the DOPT optimal, i.e. $AR = y(FPTAS)/y(DOPT)$. In the simulation, we vary the parameters α, γ, and ϵ to run the FPTAS and DOPT algorithm to obtain the results respectively. In each test, we fix two parameters and change one parameter regularly. Obviously we have three cases of ratio testing, which are named as α-ratio test, γ-ratio test and ϵ-ratio test, respectively. In α-ratio test, we set γ to 1.6 and set ϵ to 0.5 and vary the parameter α with a range from 140 to 320 with an increasing step 20. In γ-ratio test, ϵ is set to 0.5, α is set to 180\$, and γ is changed from 1.1 to 2 with an increasing step 0.1. In ϵ-ratio test, α is set to 180\$, γ is set to 1.6, and γ is changed from 1.1 to 2 with an increasing step 0.1.

We run the tests for all the 11 EDR reduction instances. The results are shown in Fig. 3. We observe that all ratios are between 1 and $1 + \epsilon$. Moreover, the ratios are closer to the line with ratio 1. It means that the FPTAS solution is very close to the optimal solution. Most of the line with ratios of varying α seems to be more flat which means the α parameter has little impact on the ratios. There are several broken lines in Fig. 3(b) and Fig. 3(c). It demonstrates that the ratio exists a special sensitivity to parameters γ and ϵ. The largest impact factor is the parameter ϵ. With the increasing of ϵ, the ratio increases as well as in theory.

2) Agents' Utilities: We study each agent's utility in all the experiments. Note that agents' utilities are concluded from the payment for each tenant subtracts his/her actual cost. Actually, we got that each tenant obtains a non-negative utility in our experiment. We only show the result in Fig. 4(a) by letting $\alpha = 180\$, \gamma = 1.6, \epsilon = 0.5$. We illustrate each winner tenant's utility in each hour. Fig. 4(a) shows that the case of the 18th hour time has the largest utility and the 5th hour time has the minimum utility.

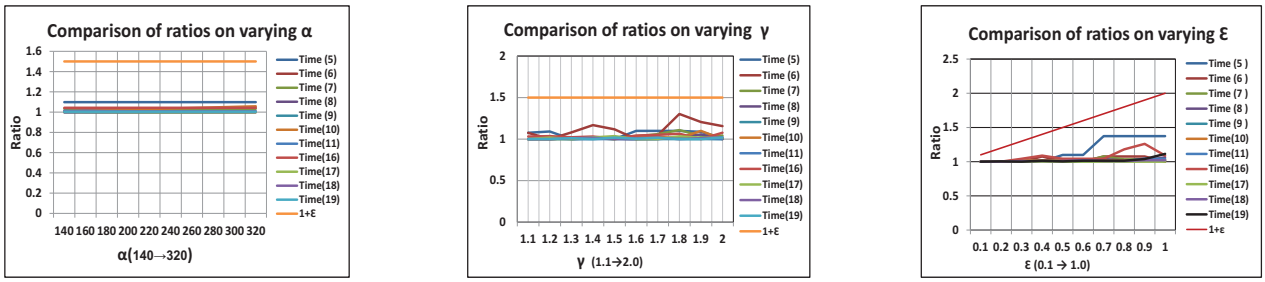
3) Social Cost Reduction Compared to BES only: Each winner tenant has obtained a non-negative utility, which implies that the colocation operator will pay a lot of money to these tenants. To study whether this payment is too much, we investigate the social costs when we only use BES. The results are illustrated in Fig. 4(b) and Fig. 4(c) when the parameters α and γ vary. The results reveal that the social costs given by the FPTAS mechanism is much smaller than that one of BES only. The percentages are declined when α or γ increases. So it's very important for the colocation data center to enable more tenants to attend the energy demand response activities.

VI. RELATED WORKS

Besides the work in [4], there are several existing research on mechanism design on DR. Ren and Islam [19] studied the mechanism design for colocation demand response, but their mechanism is not truthful and may not meet the target of EDR. Chen et al. [20] studied the green colocation data centers by designing a pricing mechanism to fulfill energy reduction requirement for EDR. The energy reduction from tenants is calculated by the price-taking and price-anticipating equilibrium. Zhou et al. [21] studied demand resource on geo-distributed cloud through VCG-based mechanism, in which the

TABLE II
THE TENANT BID DATA

Hour	Tenant#1		Tenant#2		Tenant#3		Tenant#4		Tenant#5		Tenant#6		Tenant#7		Tenant#8		Tenant#9	
	s_1	b_1	s_2	b_2	s_3	b_3	s_4	b_1	s_5	b_5	s_6	b_6	s_7	b_7	s_8	b_8	s_9	b_9
5	23	2,737	12	1,284	4	352	67	4,623	26	2,704	5	555	43	3,569	46	5,888	5	570
6	33	3,531	18	1,782	6	648	67	8,710	26	2,392	5	470	57	4,617	60	5,100	7	588
7	36	3,960	19	1,653	6	462	67	6,700	26	1,976	5	605	30	3,960	32	3,136	4	364
8	46	5,336	25	1,950	8	784	66	6,864	25	2,350	5	585	37	4,440	40	4,240	5	565
9	48	3,600	26	2,262	8	960	63	8,064	24	1,656	4	476	31	3,162	33	3,564	4	516
10	49	4,018	27	3,159	9	792	58	5,510	22	2,112	4	332	33	3,267	35	2,590	4	456
11	50	6,000	27	3,078	9	693	53	5,353	20	2,440	4	340	33	3,102	35	3,115	4	384
16	55	3,960	30	2,160	10	1,150	39	3,159	15	1,995	3	399	75	7,950	80	8,320	10	1,290
17	52	4,420	28	2,016	9	1,170	36	4,068	14	1,484	2	178	60	7,620	64	4,352	8	952
18	57	7,581	31	3,100	10	1,030	37	4,144	14	1,358	2	212	60	6,540	64	7,360	8	712
19	47	4,136	25	3,275	8	744	39	5,031	15	1,380	3	270	75	7,725	80	5,760	10	930



(a) α-ratio test: the ratio between FPTAS and DOPT for eleven EDR instances on varying α, where $\gamma = 1.6, \epsilon = 0.5$

(b) γ-ratio test: the ratio between FPTAS and DOPT for eleven EDR instances on varying γ, where $\alpha = 180\$, \epsilon = 0.5$

(c) ϵ-ratio test: the ratio between FPTAS and DOPT for eleven EDR instances on varying ϵ, where $\alpha = 180\$, \gamma = 1.6$

Fig. 3. The performance evaluation on approximation ratios in distinct time of EDR instances.

utility of each agent depends highly on its interactive workload. Sun et al. [22] considered fairness among the mechanism design and provided online mechanism with competitive ratio of 3.2 in expectation. Ahmed et al. [23] proposed a contract-based mechanism, in which the colocation operator offers a set of contracts (i.e., a pair of energy reduction and rewards) to tenants and tenants can voluntarily select none or one of the contracts to accept, while the objective is to minimize the operator's cost, the sum of rewards plus the cost of BES. Islam et al. [24] reduced the operator's cost by learning tenants' response to reward. Tran et al. [25] used a two-stage Stackelberg game to model the economic demand response where the operator can adjust an elastic energy reduction target.

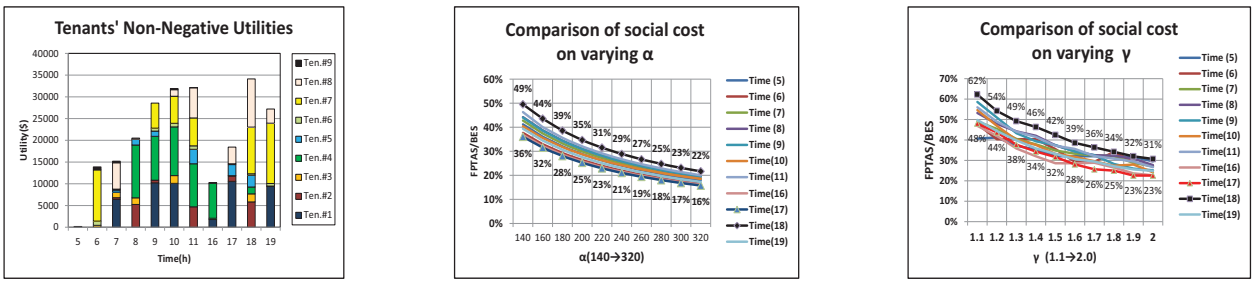
A closely related work is DR in smart grids. Zhou et al. [6] studied the mechanism design on DR in smart grids. Let $\alpha = 1$, and there is an upper bound on the BES, i.e. $y \leq z_{\max}$. A randomized FPTAS mechanism was given in [6]. Their idea is to combine with smooth analysis and randomize auction. Actually, Dough and Roughgarden [26] showed that if there exists an FPTAS approximation, then this algorithm can be transformed into a truthful in expectation mechanism that retains the FPTAS property. The work in [26] does not require the existence of FPTAS. However, it still remains open whether there exists a deterministic FPTAS.

A vast amount of work has been done for mechanism design on multi-unit auction problem [27], [28], in which there is a set of identical items among bidders, and every bidder has a private valucation function on the number of items, and the problem is to find an allocation of the items to the bidders so as to maximize the sum of bidders' valuations.

Briest et al. [15] presented a truthful FPTAS for the max-knapsack problem. Our problem differs from the min-knapsack in which we have BES such that the capacity of the knapsack we need to cover is soft.

VII. CONCLUSIONS

In this paper, we have proposed a deterministic truthful FPTAS mechanism with 1+ϵ approximation ratio for a reverse auction of EDR in colocation data centers. We have developed an auction system and implemented a bidding decision tool for simulation experiments. The experimental results demonstrate the effectiveness of our methods. In future work, we study the more practical utility of our mechanism and the performance optimization with parallelization techniques for the dynamical programming. Besides, the provided technique allows us to deal with single minded agents, and both the size of energy and the cost are private information. Many open problems arise in the area of demand response. For example, agents are multi-minded.



(a) Comparison of tenants' non-negative utilities when $\alpha = 180, \gamma = 1.6, \epsilon = 0.5$.

(b) Comparison of social cost ratio between FPTAS and BES, where $\gamma = 1.6$, $\epsilon = 0.5$ and α varying from 140 to 320 with a growing step 20.

(c) Comparison of social cost ratio between FPTAS and BES, where $\alpha = 180$, $\epsilon = 0.5$ and γ varying from 1.1 to 2.0 with a growing step 0.1.

Fig. 4. The performance evaluation on tenants' utility and social costs.

ACKNOWLEDGMENT

This research was in part supported by NSFC (11671355, 61772466, U1836202, 60970125, 61272303, 61472359), the National Key R&D Program of China (2017YFB0203202), the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under No. LR19F020003, the Provincial Key Research and Development Program of Zhejiang, China under No. 2017C01055, and the Alibaba-ZJU Joint Research Institute of Frontier Technologies.

REFERENCES

[1] F. Kamyab, M. Amini, S. Sheykhha, M. Hasanpour, and M. M. Jalali, "Demand response program in smart grid using supply function bidding mechanism," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1277–1284, 2016.

[2] M. H. Y. Moghaddam, A. Leon-Garcia, and M. Moghaddassian, "On the performance of distributed and cloud-based demand response in smart grid," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[3] W. Wei, F. Liu, and S. Mei, "Energy pricing and dispatch for smart grid retailers under demand response and market price uncertainty," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1364–1374, 2017.

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

[5] P. Jacquot, O. Beaude, S. Gaubert, and N. Oudjane, "Demand side management in the smart grid: An efficiency and fairness tradeoff," in *IEEE Pes Innovative Smart Grid Technologies Conference Europe*, 2018, pp. 1–6.

[6] R. Zhou, Z. Li, C. Wu, and M. Chen, "Demand response in smart grids: A randomized auction approach," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2540–2553, 2015.

[7] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of finance*, vol. 16, no. 1, pp. 8–37, 1961.

[8] E. H. Clarke, "Multipart pricing of public goods," *Public choice*, vol. 11, no. 1, pp. 17–33, 1971.

[9] T. Groves, "Incentives in teams," *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

[10] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," *Journal of the ACM (JACM)*, vol. 58, no. 6, p. 25, 2011.

[11] A. Archer and E. Tardos, "Truthful mechanisms for one-parameter agents," in *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*. IEEE Computer Society, 2001, pp. 482–491.

[12] D. Lehmann, L. I. Oćallaghan, and Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions," *Journal of the ACM (JACM)*, vol. 49, no. 5, pp. 577–602, 2002.

[13] E. L. Lawler, "Fast approximation algorithms for knapsack problems," *Mathematics of Operations Research*, vol. 4, no. 4, pp. 339–356, 1979.

[14] I. M. Tauhidul, "Approximation algorithms for minimum knapsack problem," *Master's degree Thesis, university of lethbridge*, 2009.

[15] P. Briest, P. Krysta, and B. Vöcking, "Approximation techniques for utilitarian mechanism design," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1587–1622, 2011.

[16] A. Mu'Alem and N. Nisan, "Truthful approximation mechanisms for restricted combinatorial auctions," *Games and Economic Behavior*, vol. 64, no. 2, pp. 612–631, 2008.

[17] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, 2013.

[18] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. G. Andersen, G. R. Ganger, and Gibson, *Safe and effective fine-grained TCP retransmissions for datacenter communication*. ACM, 2009.

[19] S. Ren and M. Islam, "Colocation demand response: Why do i turn off my servers?" in *Proc. of 11th International Conference on Autonomic Computing (ICAC 2014)*. USENIX Association, 2014.

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

[21] Z. Zhou, F. Liu, Z. Li, and H. Jin, "When smart grid meets geo-distributed cloud: An auction approach to datacenter demand response," in *Proc. of IEEE INFOCOM*, 2015.

[22] Q. Sun, C. Wu, S. Ren, and Z. Li, "Fair rewarding in colocation data centers: Truthful mechanism for emergency demand response," in *IEEE 23rd IWQoS 2015*, 2015, pp. 363–372.

[23] K. Ahmed, M. A. Islam, and S. Ren, "A contract design approach for colocation data center demand response," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2015)*. IEEE Press, 2015, pp. 635–640.

[24] M. Islam, H. Mahmud, S. Ren, X. Wang *et al.*, "Paying to save: Reducing cost of colocation data center via rewards," in *Proc. of IEEE 21st International Symposium on High Performance Computer Architecture (HPCA 2015)*. IEEE, 2015, pp. 235–245.

[25] N. H. Tran, C. T. Do, S. Ren, Z. Han, and C. S. Hong, "Incentive mechanisms for economic and emergency demand responses of colocation datacenters," *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 12, pp. 2892–2905, 2015.

[26] S. Dughmi and T. Roughgarden, "Black-box randomized reductions in algorithmic mechanism design," *SIAM Journal on Computing*, vol. 43, no. 1, pp. 312–336, 2014.

[27] J. Lessan and S. Karabatı, "A preference-based, multi-unit auction for pricing and capacity allocation," *Computers & Operations Research*, vol. 91, pp. 237–246, 2018.

[28] E. Anderson and P. Holmberg, "Price instability in multi-unit auctions," *Journal of Economic Theory*, vol. 175, pp. 318–341, 2018.