Coordinated Demand Response By Data Centers Using Inverse Optimization

Athanasios Tsiligkaridis
Department of ECE
Boston University
Boston, MA, USA
atsili@bu.edu

Ioannis Ch. Paschalidis
Department of ECE
Boston University
Boston, MA, USA
yannisp@bu.edu

Ayse Coskun
Department of ECE
Boston University
Boston, MA, USA
acoskun@bu.edu

***Abstract*—Demand Response (DR) policies define the interactions between an energy supplier and its consumers and allow for customer energy regulation given a supplier request. Given the high flexibility and controllability of Data Centers (DC), they are promising candidates to participate in DR for power grid stabilization. In this work, we consider the setting where an energy supply deficit event occurs and must be addressed to avoid grid strain. We present two novel frameworks for DR, where a load aggregator offers price incentives to a set of consumer DCs so they can dynamically adjust their electricity consumption and provide DR to the grid via server usage reductions. Modeling DCs using realistic cost functions based on Quality of Service (QoS) requirements of the DC workloads, we present a data-driven inverse optimization method to estimate DC cost function parameters for precise and efficient pricing and provide an algorithm for solving the inverse problem. Experimental results on two test cases demonstrate the benefits of our proposed DR mechanisms for energy control.**

I. Introduction

DR is a burgeoning solution to the problem of power instability and peak reduction in the grid. Both energy supply and demand can vary greatly and their inequality can be costly to both a utility company/generator and the users it services. The standard practice is that utility providers must provision enough capacity to service peak demand. As usage amounts near the peak are achieved for only brief amounts of time, this results in inefficiencies and an overall underutilization of energy resources.

The creation of a robust smart grid armed with a design for handling supply and demand imbalances is vital and can allow for both smart energy resource control and the inclusion of a larger amount of renewable energy resources [1]. By default, renewables are inherently intermittent (i.e., wind and solar energy supply varies due to their dependence on weather occurrences); but, a DR program could be harnessed as a means of electricity demand management for service stability.

A DR policy requires a market model that encourages user participation. One such model is Supply Function Bidding (SFB), which is an incentive-based model frequently used in the electricity market [2]. In SFB, users submit supply functions as bids to an aggregator, which specify the energy quantities the users are committed to alter. The aggregator then uses these bids to determine a Market Clearing Price (MCP) that is used to provide a payment to each user for its participation. This framework has been studied along with the resulting supply function equilibrium and its uniqueness and existence conditions [3]. Market efficiency and price bounds have been studied using parameterized SFB [4]; linear SFB has also been used to determine the MCP and characterize the efficiency of the resulting equilibria [5]. A capacity bidding program has also been proposed to allow for reliable DC DR through aggregation and DC cooperation [6]. A pricing mechanism using parameterized SFB was also used for greening DC DR in the case of an energy supply deficit [7]. Inverse optimization approaches have previously been used for estimating cost functions in addition to bidding in a day-ahead market to find the MCP and equilibria [8]. Similar methods have also been employed in the estimation of market bids and forecasting of aggregate demands of electricity customers [9], [10].

In addition to SFB, price-based models are emerging where users directly respond to a given price from the aggregator without bidding. A prediction-based pricing approach was first used as a DR market design [11]. Online convex optimization methods have also been presented for DR real-time pricing in smart grids [12], and supply function bidding/pricing models have been investigated along with their resulting equilibria [13]. Coincident peak pricing and DC algorithms have also been demonstrated to avoid the coincident peak and reduce energy costs [14]. Additionally, real-time pricing approaches using a Stackelberg game have been presented for geo-distributed DCs [15]. We recently proposed a simple DC DR formulation using realistic QoS requirements and an inverse optimization method for pricing DCs [16].

In this work, we consider DCs as a DR resource due to their flexibility, capacity, and ease of monitoring and adjustment [17], [18]. DCs are entities owned by organizations used for storing, maintaining, and processing data, as well as processing computationally demanding jobs. As the amount of data in the world continues to grow (due to connected devices, burgeoning internet access in developing countries, artificial intelligence, etc. [19]), DCs will become more prevalent and their electricity usage will have to be examined and kept in check to avoid grid strain since they are heavy power consumers. United States DCs used over 90 TWh of electricity in 2013; by 2020, this usage is expected to increase to 140 TWh per year and will result in \$13 billion per year in electricity bill costs [20]. In addition, 1.62% and 3% of the world's energy was used by DCs in 2014 and 2017, respectively [21], [22], with their energy usage expected to continue to grow in the coming years. Due to today's COVID-19 crisis, DCs are experiencing increased surges of use due to the rise of online internet usage for items such as remote working, e-learning, and entertainment streaming [23], [24]; it is important to have a means of controlling DC usage while simultaneously offering much needed flexibility to the grid.

We argue that DR for DCs can accomplish delivering this flexibility, if provided the necessary policies for smart pricing. DR programs can harness DC flexibility to allow for the decrease of energy usage costs by providing financial benefits from participation [25]. A DR program allows an aggregator to handle power supply intermittency (e.g., due to renewable sources) by inducing desired customer responses through payment incentives.

The main contribution of this paper is to present two novel policies (Non-Aggregate (NAP) and Aggregate (AP)) of a coordinated DR mechanism utilizing DCs as a means of achieving desired target load reductions over all DC customers to provide grid relief in the case of an energy supply deficit. We construct realistic DC cost functions that capture the provisioning of QoS by the DC to the jobs it services. We present a means of reconstructing each cost function by using historical data observations of the interactions between a load aggregator and the DCs through an inverse optimization framework along with a data-driven algorithm for obtaining the best cost function parameter estimates from the solution of the inverse problem. We lastly present simulation results for both formulations that showcase their abilities to effectively achieve desired target energy load reductions in a critical situation of energy supply deficiency.

II. Problem Formulation

We focus on a DC DR setting which involves an Independent System Operator (ISO), an aggregator, and a number S of DCs. We consider a real-time pricing market model where DCs are given price incentives via the aggregator to dynamically reduce their power consumption during times of energy supply deficit. To circumvent the issue of imbalanced supply and demand, an overall energy usage decrease amount T must be achieved over all DCs; this amount is determined by an interaction between the ISO and the aggregator. With this target in mind, the aggregator attempts to induce each DC to offer DR to the grid by adjusting its energy consumption such that the overall energy consumption decrease by the DCs will match the target decrease value T. For DC i, $i \in [S] \triangleq \{1, ..., S\}$, to use less energy, it must reduce the number of servers that it has running from a nominal amount chosen at the upper limit N_i^H to a smaller amount N_i.

Participation in the DR scheme is incentivized via the broadcast of a price p_i (represents price per unit of energy reduction) for DC i that is used to calculate the compensation that DC i can receive for delaying/not completing jobs. The price vector $\boldsymbol{p} = [p_i] \;\forall\; i \in [S]$ is obtained via the minimization of a *social cost* metric, which is a metric that considers both how well the desired target load reduction is met and how much QoS cost the DCs are subject to. Setting the appropriate price requires knowledge of how each DC responds to a given set price, thus motivating the use of a cost function estimation method.

We model each DC as a G/G/1 queuing system, where jobs arrive and are carried out by that DC's running servers. The queueing model assumes a single server representing a pooled resource consisting of all available servers in the DC. We scale the service process parameters based on the number of available servers.

Let A_i and B_i denote discrete stochastic processes that correspond to the number of jobs that *arrive* and are *serviced* (and in turn, *depart*), respectively, at DC i. Specifically, $A_i = \{A_{i,1}, A_{i,2}, ...\}$ and $B_i = \{B_{i,1}, B_{i,2}, ...\}$ where $A_{i,t}$ (respectively, $B_{i,t}$) is the random variable representing the number of arrivals (respectively, departures) in DC i at time slot t. If we have N_i servers in DC i, the number of jobs that depart will be scaled based on this N_i, assuming that the number of jobs serviced per server per unit of time is constant, on average. We use these processes to obtain an individual QoS constraint for each DC. Qualitatively, QoS represents the effectiveness of a DC in completing query jobs. The QoS is modeled using the probability that the queue length L in the system exceeds or equals some value U: $P[L \geq U]$. We want this probability value to be as small as possible. For DC i, we can use Theorem 6.1.1 from prior work [26] to approximate this probability with an exponential as: $P[L \geq U] \sim e^{-\theta_i^* U}$, where $\theta_i^* > 0$ is a scalar that depends on the arrival and service distributions. Specifically, θ_i^* is the positive root to Eq. (1), where $\Lambda_{A_i} = \lim_{n\to\infty} \frac{1}{n} \log E[e^{\theta_i \sum_{i=1}^{n} A_i}]$ and $\Lambda_{B_i} = \lim_{n\to\infty} \frac{1}{n} \log E[e^{\theta_i \sum_{i=1}^{n} B_i}]$ are limits of log Moment Generating Functions (MGF):

$$\Lambda_{A_i}(\theta_i^*) + \Lambda_{B_i}(-\theta_i^*) = 0. \tag{1}$$

If we have N_i servers in a DC, the service process will get linearly scaled. In particular, let $D_{i,t} = N_i B_{i,t}$ denote the number of departing jobs in DC i during time slot t, where $B_{i,t}$ is the number of jobs departing from each server. Let $D_i = \{D_{i,1}, D_{i,2}, ...\}$ denote the stochastic process of departures from DC i. With this, θ_i^* will be the root of Eq. (2), where Λ_{A_i} is defined previously and $\Lambda_{D_i} = \lim_{n\to\infty} \frac{1}{n} \log E[e^{\theta_i \sum_{i=1}^{n} D_i}]$:

$$\Lambda_{A_i}(\theta_i^*) + \Lambda_{D_i}(-\theta_i^*) = 0. \tag{2}$$

For a QoS constraint, we would like $P[L \geq U]$ to be as small as possible; in turn, we want θ_i to be as large as possible. We use this parameter to design DC i's cost function such that it yields a small cost for a large θ_i, and vice versa. We model DC i's cost function, $C_i(\theta_i)$, as a *convex, non-increasing* function, which can be interpreted as providing better QoS as θ_i grows. Example valid cost functions could include scaled exponential or logarithmic functions.

III. Non-Aggregate Policy (NAP)

We first propose a NAP for describing the interations between a load aggregator and S DCs. In this non-aggregate formulation, each DC responds to a given price by the aggregator by solving its own greedy cost minimization problem; no collaboration

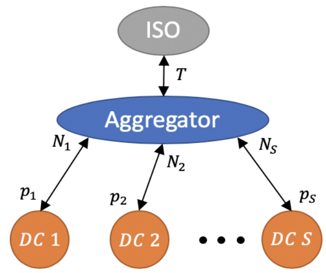


Fig. 1: A visualization of the NAP.

takes place between the DCs and no resource sharing is permitted.

Figure 1 presents the system where an ISO and aggregator decide on a target load reduction T and, afterwards, the aggregator then decides on prices $[p_i]$, $\forall\ i \in [S]$, to broadcast to its DCs to extract flexibility from them via energy usage decreases. We define a social cost in Eq. (3) that serves as the basis of how the aggregator determines its pricing scheme:

$$G(\boldsymbol{p}) = q\left(T - \sum_{i=1}^{S} \alpha_i (N_i^H - N_i^*(p_i))\right) + \sum_{i=1}^{S} C_i(\theta_i(N_i^*(p_i))). \tag{3}$$

The aggregator has its own goal of achieving a load reduction T over all S DCs due to the supply deficit; this is encompassed in the first term of Eq. (3) where q is a convex penalty function and the quantity $\alpha_i(N_i^H - N_i^*(p))$ represents DC i's energy usage decrease. $N_i^*(p)$ is the optimal number of servers DC i runs when presented with the price p_i. The aggregator also considers the overall DC QoS costs since it wants to encourage participation in the DR; the second term in Eq. (3) represents the sum of the QoS costs and it can be considered as a social welfare term. To find the best DR-inducing price $\boldsymbol{p}^*$, the aggregator solves (4):

$$\boldsymbol{p}^* \in \arg\min_{\boldsymbol{p} \geq \boldsymbol{0}} G(\boldsymbol{p}), \tag{4}$$

where each DC's cost function, $C_i(\theta_i)$, $\forall\ i \in [S]$, is needed to fully represent the social cost in Eq. (3). Complete cost function information is not directly accessible; thus, we propose the use of an inverse optimization framework to be used by the aggregator for cost function parameter estimation that will be discussed in Section III-B. With correctly estimated cost functions, the aggregator can fully express the social cost metric and can solve (4) to determine $\boldsymbol{p}^*$.

A. Forward Problem - NAP

Each DC i solves a cost minimization problem when responding to a price p_i given by the aggregator. The general forward optimization problem that DC i solves is defined as:

$$\begin{aligned} \min_{N_i, \theta_i} \quad & -p_i \alpha_i (N_i^H - N_i) + C_i(\theta_i) \\ \text{s.t.} \quad & \Lambda_{A_i}(\theta_i) + \Lambda_{D_i}(-\theta_i) = 0, \\ & N_i^L \leq N_i \leq N_i^H, \end{aligned} \tag{5}$$

where α_i represents the amount of energy use per server. We can interpret p_i as the price per unit of energy usage. The first term in the objective of (5) represents the benefit that DC i will obtain by reducing its server counts to N_i while the second term represents a QoS cost that is high when QoS is low (θ_i small) and vice versa. In addition, two constraints are placed to enforce the bounds on the number of servers a DC can run and ensure a QoS level as shown in Eq. (2).

Regarding the QoS cost, we expect there to be a relation between the number of servers DC i runs, N_i, and the cost, $C_i(\theta_i)$. We expect the QoS cost to decrease as the number of running servers increases since more running servers allows for more jobs to be competed, and thus, better QoS. Ultimately, we expect a direct correlation between N_i and θ_i; this behavior creates a desired two-way competition in the objective function of (5). From the objective, we see that a DC would want to reduce its servers as much as possible to receive the most benefit for its energy reduction; but, the reduction is limited based on the QoS cost since QoS cost increases as the number of running servers decreases.

We assume the arrival and service processes, A_i and B_i, $\forall\ i$, respectively, are Gaussian distributed. This choice results in the desired competitive behavior in the objective function of (5) as discussed in Theorem III.1.

Theorem III.1. *Assume A_i and B_i are Gaussian distributed processes in the QoS constraint of (5). Then, the optimal solutions $N_i^*(p_i)$ of (5), $\forall\ i \in [S]$, will lie somewhere in the interval specified by the bound constraint in (5), but not necessarily at the bounds.*

Proof. We aim to show that the two terms in the objective of (5) move in different directions when the input, the server responses, moves in a given direction.

Assume $[\theta_i^*(p),\ N_i^*(p)]$ is the optimal solution to problem (5) for a given price p. Then, this solution must satisfy the QoS constraint in (5). If we assume A_i and B_i are Gaussian processes for arrivals and services, the QoS constraint with scaled service rate becomes, after simplification,

$$\theta_i^*(N_i^*) = \frac{2\left(N_i^* \mu_B^{(i)} - \mu_A^{(i)}\right)}{N_i^* \sigma_B^{2^{(i)}} + \sigma_A^{2^{(i)}}}, \tag{6}$$

where $[\mu_A^{(i)}, \sigma_A^{2^{(i)}}]$ represent parameters of the arrival process, $[\mu_B^{(i)}, \sigma_B^{2^{(i)}}]$ represent parameters of the service process, and N_i is the amount of running servers in DC i.

From Eq. (6), it follows that there exists a strictly increasing function g^i that relates θ_i^* and N_i^* as $\theta_i^*(N_i^*) = g^i(N_i^*)$. With this, $C_i(\theta_i^*(N_i^*))$ will be a strictly decreasing function of N_i^*, as desired.

When solving the forward problem (5), the objective function has one term (i.e., the payment term) that wants to decrease server counts while the second term (QoS cost) influences the server counts to increase. Thus, there is competition in the objective and the optimal solution to (5) will lie somewhere in the bound constraints specified in (5) but not necessarily at the bounds.

$\square$

Choosing discrete Gaussian distributions for the arrival and service processes results in a nonlinear relation between θ_i and N_i shown in the QoS constraint of (5). If we take that constraint and go backwards from the service process 'per DC' quantities D_i to the 'per server' quantities B_i by scaling the service means and variances by the amount of running servers N_i, we obtain the relationship depicted in Eq. (6).

We can remove the QoS constraint from (5) since it will be solved with equality and can replace θ_i in (5) with Eq. (6). In addition, we define a new decision variable as $\tilde{N}_i = N_i - N_i^L$. DC i's forward problem then effectively becomes the following single variable convex optimization problem that is solved in response to a set price p_i where $N_i^U = N_i^H - N_i^L$:

$$\begin{aligned} \min_{\tilde{N}_i} \quad & -p\alpha_i(N_i^U - \tilde{N}_i) \\ & + C_i\left(\frac{2\left(\left(\tilde{N}_i + N_i^L\right)\mu_B^{(i)} - \mu_A^{(i)}\right)}{\left(\tilde{N}_i + N_i^L\right)\sigma_B^{2^{(i)}} + \sigma_A^{2^{(i)}}}\right) \\ \text{s.t.} \quad & 0 \le \tilde{N}_i \le N_i^U. \end{aligned} \tag{7}$$

B. Inverse Problem - NAP

To solve (4), the aggregator must know each DCs cost function. Through the use of the inverse variational inequality framework [27], we can use data observations to learn each cost function by solving an inverse optimization problem.

For a specific price $p_i|_m$, each DC i provides a response $\tilde{N}_i|_m$; we use a set of M observations of $\langle p_i|_m, \tilde{N}_i|_m\rangle$, $\forall\ i \in [S]$, $\forall\ m \in [M] \triangleq \{1, ..., M\}$ as data observations of the DR system interactions.

Let us look at our *convex, non-increasing* cost function, $C_i(\theta_i)$. We re-define it as: $C_i(\theta_i) = \boldsymbol{k}_i^T \boldsymbol{c}(\theta_i)$, a product of a vector of positive scaling parameters $\boldsymbol{k}_i \in \mathbb{R}^V_{>0}$ and a vector of *convex, non-increasing* constituent functions $\boldsymbol{c}(\theta_i) = (c_1(\theta_i), ..., c_V(\theta_i))$. One can think of these constituent cost functions as "features" influencing the DCs' cost associated with QoS and the specific Service Level Agreements (SLA) it has promised to its customers.

By using observations of DC responses to prices, we can solve an inverse optimization problem to learn the coefficients $\boldsymbol{k}_i$ for DC i's cost function. This parameter vector helps define the magnitude of the cost that DC i is subject to when reducing its servers from the nominal amount.

For observed price $p_i|_m$, define the objective function of the forward problem (7) as: $\phi_i(\tilde{N}_i|_m, \boldsymbol{k}_i, \boldsymbol{\gamma}_{i,m}) = -p_i|_m\alpha_i(N_i^U - \tilde{N}_i|_m) + \boldsymbol{k}_i^T \boldsymbol{c}(\theta_i(\tilde{N}_i|_m))$, where $\boldsymbol{\gamma}_{i,m} = (p_i|_m, \alpha_i, N_i^U, \mu_A^{(i)}, \mu_B^{(i)}, \sigma_A^{2^{(i)}}, \sigma_B^{2^{(i)}})$. With this notation, we can set up the inverse problem by applying Theorem 3 from [27] as:

$$\begin{aligned} \min_{\boldsymbol{k}_1,...,\boldsymbol{k}_S,\boldsymbol{y},\boldsymbol{\epsilon}} & \|\boldsymbol{\epsilon}\|_\infty \\ \text{s.t.} \ & -y_i^m \le 0,\ \forall i, \forall m, \\ & -y_i^m - \frac{\partial}{\partial \tilde{N}_i|_m}\phi_i(\tilde{N}_i|_m, \boldsymbol{k}_i, \boldsymbol{\gamma}_{i,m}) \le 0,\ \forall i, \forall m, \\ & \sum_{i=1}^{S}\left[\frac{\partial}{\partial \tilde{N}_i|_m}\phi_i(\tilde{N}_i|_m, \boldsymbol{k}_i, \boldsymbol{\gamma}_{i,m})\tilde{N}_i|_m + N_i^U y_i^m\right] - \epsilon_m \\ & \le 0,\ \forall m, \end{aligned} \tag{8}$$

where $||\boldsymbol{\epsilon}||_\infty = \max_m|\epsilon_m|$ is the infinity norm of $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_M)$. The inverse problem is a minimization of the norm of an error such that we have an approximate solution to a variational inequality problem.

The first constraint in (8) represents the non-negativity of the dual variable $\boldsymbol{y}$ while the second constraint is derived from weak duality. The final constraint represents the minimization of the duality gap $\boldsymbol{\epsilon}$ between the primal and dual problems used in deriving the inverse.

The inverse problem can be reformulated into a Linear Program and solved efficiently. By solving (8), the aggregator can obtain the cost coefficients for all DCs and can then solve (4) to find $\boldsymbol{p}^*$.

IV. Aggregate Policy

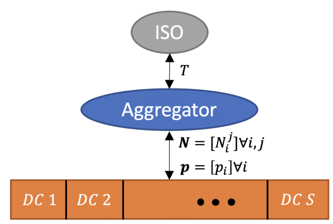


Fig. 2: A visualization of the AP.

Next, we propose an AP for describing the interations between a load aggregator and S DCs. In this aggregate formulation, DCs make a collective decision via the solution of a joint optimization problem when determining their running server counts and they are permitted to share resources in the form of servers. This policy is proposed as an alternative to the NAP with potential benefits due to the option of resource sharing; also, this AP without any sharing reduces to a problem whose solution is equivalent to the NAP forward problems' solutions.

Figure 2 presents the system where an ISO and aggregator decide on a target T and, afterwards, the aggregator finds a $\boldsymbol{p}$ to broadcast to its DCs for DR participation. Similar to the NAP, the aggregator determines the best price vector to broadcast via the solution of (4), where $G(\boldsymbol{p})$ is defined as:

$$\begin{aligned} G(\boldsymbol{p}) = q&\left(T - \sum_{i=1}^{S}\alpha_i(N_i^U - \mathbf{1}^\intercal \boldsymbol{N}_i^*(\boldsymbol{p}))\right) \\ &+ \sum_{i=1}^{S} C_i(\theta(\boldsymbol{N}_i^*(\boldsymbol{p}))), \end{aligned} \tag{9}$$

and $\boldsymbol{N}_i^*(\boldsymbol{p}) = [N_i^{j^*}]\ \forall\ j$.

The aggregator must still learn each DCs cost function in order to solve (4) using Eq. (9) as the social cost. An inverse optimization approach similar to the one in the NAP is proposed for cost function estimation.

A. Forward Problem - AP

A single joint optimization problem over all S data centers must be solved to yield each DCs running server responses where server sharing is permitted. The solution of this problem outputs decision variables $N_i^j,\ \forall\ i \in [S],\ \forall\ j \in [S]$, where N_i^j represents the number of servers in DC i that DC j uses.

As in the NAP, we consider each DC $i,\ \forall\ i \in [S]$, as a queuing system characterized by a discrete arrival and service process. Since DCs can share resources (i.e., servers), we can think of sharing as re-allocating servers to arrivals; thus, service parameters will be altered based on how sharing is done.

To model the DC cost functions, we use a convex and non-decreasing cost function $C_i(\theta_i)$ to model the QoS cost based on a QoS parameter θ_i for DC i. Similar to the NAP, we assume Gaussian arrival and service processes; θ_i can be expressed as a function of the number of running servers in a DC as:

$$\theta_i(\boldsymbol{N}) = \frac{2\left(\sum_{l=1}^{S} N_l^{(i)}\mu_B^{(l)} - \mu_A^{(i)}\right)}{\left(\sum_{l=1}^{S} N_l^{(i)}\sigma_B^{2(l)} + \sigma_A^{2(i)}\right)}, \tag{10}$$

which is similar to Eq. (6) except that the service process mean and variances are scaled based on how sharing is done.

The joint forward optimization problem is defined in (11) where we redefine the decision variables as $\tilde{N}_i^j = N_i^j - N_i^L/S$ and the argument in the cost function C_i represents the QoS parameter θ_i from Eq. (10). In the objective, the first term represents an aggregate payment for load reduction over all DCs, the second term represents the cost of foreign server usage where κ is a known parameter, and the final term represents the aggregate QoS cost over all DCs.

$$\begin{aligned}
\min_{\tilde{N}_i^j \forall i,j} \quad & -\sum_{i=1}^{S} p_i\alpha_i(N_i^U - \sum_{j=1}^{S}\tilde{N}_i^j) \\
& + \kappa\sum_{i=1}^{S}\sum_{j\in\{1,\ldots,S\}\setminus i}\tilde{N}_i^j \\
& + \sum_{i=1}^{S} C_i\left(\frac{2\left(\sum_{l=1}^{S}\left(\tilde{N}_l^i + \frac{N_l^L}{S}\right)\mu_B^{(l)} - \mu_A^{(i)}\right)}{\left(\sum_{l=1}^{S}\left(\tilde{N}_l^i + \frac{N_l^L}{S}\right)\sigma_B^{2(l)} + \sigma_A^{2(i)}\right)}\right) \\
\text{s.t.} \quad & 0 \le \sum_{j=1}^{S}\tilde{N}_i^j \le N_i^U\ \forall i.
\end{aligned} \tag{11}$$

B. Inverse Problem - AP

As in the NAP, an inverse formulation is proposed to learn the cost coefficients $k_i,\ \forall\ i$, from the DC cost functions C_i where each function is defined as $C_i(\theta_i) = \boldsymbol{k}_i^T\boldsymbol{c}(\theta_i)$. Since the constraint set of (11) is of similar form as that of (7), we can form the inverse in the same manner. In this case, the data observations we provide as input are sets of prices and DC responses. Specifically, we have M observations of prices $p_i|_m\ \forall\ i, m$ and responses $N_i^j|_m\ \forall\ i,j \in [S], m \in [M]$. For each observed price, define the objective function of (11) as:

$$\begin{aligned}
\phi(\tilde{\boldsymbol{N}}_i|_m, \boldsymbol{k}_1, \boldsymbol{\gamma}_{i,m}) = & -\sum_{i=1}^{S} p_i|_m\alpha_i(N_i^U - \sum_{j=1}^{S}\tilde{N}_i^j|_m) \\
& + \kappa\sum_{i=1}^{S}\sum_{j\in\{1,\ldots,S\}\setminus i}\tilde{N}_i^j|_m + \sum_{i=1}^{S} C_i(\theta_i(\tilde{\boldsymbol{N}}_i)),
\end{aligned}$$

where $\boldsymbol{\gamma}_{i,m} = (\boldsymbol{p}|_m, \boldsymbol{\alpha}, \boldsymbol{N}^U, \boldsymbol{\mu}_A, \boldsymbol{\mu}_B, \boldsymbol{\sigma}_A^2, \boldsymbol{\sigma}_B^2)$ and $\tilde{\boldsymbol{N}}_i^*|_m = [\tilde{N}_i^j|_m]\ \forall\ j$; the inverse problem is then formulated similarly as in Eq. (8) as:

$$\begin{aligned}
& \min_{\boldsymbol{k}_1, \boldsymbol{y}, \boldsymbol{\epsilon}} \|\boldsymbol{\epsilon}\|_\infty \\
& \quad \text{s.t. } y_i^m \le 0,\ \forall i, \forall m, \\
& y_i^m - \frac{\partial}{\partial \tilde{N}_i^j|_m}\phi(\tilde{\boldsymbol{N}}_i^*|_m, \boldsymbol{k}_i, \boldsymbol{\gamma}_{i,m}) \le 0, \forall i, \forall j, \forall m, \\
& \sum_{i=1}^{S}\left[-N_i^U y_i^m + \sum_{j=1}^{S}\frac{\partial}{\partial \tilde{N}_i^j|_m}\phi(\tilde{\boldsymbol{N}}_i^*|_m, \boldsymbol{k}_i, \boldsymbol{\gamma}_{i,m})\tilde{N}_i^j|_m\right] \\
& \qquad - \epsilon_m \le 0, \forall m.
\end{aligned} \tag{12}$$

V. Solving the Inverse Problem

The solutions obtained from solving the inverse problems in (8) and (12) have been observed to be dependent on the input data obtained from solving the respective forward problems. For some set of input observations, incorrect cost parameters can result from the inverse problem. To obtain the best solution set, we propose an algorithm that finds the most suitable parameter sets by using different subsets of input data to obtain parameter estimates that are evaluated using a validation set.

Algorithm 1 involves forming a training and validation set of samples where the training set is used to solve the inverse and obtain a parameter set while the validation data is used to evaluate the derived parameters, similar to the method used in [8]. If the validation set is not accurately reconstructed using the obtained parameters, we re-solve the inverse using a different allocation of data and continue with this procedure until the appropriate parameters are defined as based on a deviation metric dependent on the estimation of the correct number of running servers obtained from the solution of the forward problems presented in (7) and (11).

VI. Experimental Results

To compare the two proposed NAP and AP formulations, we are interested in seeing how well they can be used by an aggregator to achieve desired target energy reduction amounts. We look at three performance metrics: 1) social cost, 2) penalty cost (first terms in both social cost metrics in Eq. (3) and Eq. (9)), and 3) foreign server counts (unscaled version of the second term in the objective of (11)).

For our experiments, we consider the simple case of $S = 2$ data centers. Having a larger S would provide the benefit of achieving larger target load reduction values as there are more DCs present to provide relief in the DR. We assume each DC cost function has the following form: $C_i(\theta_i) = k_{i,1}E[A_i]e^{-k_{i,2}\theta_i}$, where θ_i is DC i's QoS parameter and $E[A_i] = \mu_A^{(i)}$ is used as an additional scaling parameter to incorporate job arrival information into the QoS cost function.

Algorithm 1 Obtaining True Cost Parameters In NAP/APs

1: **Input:** M (amount of sample points), τ (deviation threshold), L (split percentage).
2: **Output:** $\boldsymbol{\theta}^*$ (true cost parameter set).
3: **repeat**
4: (a) Partition the M sample points randomly into sets M_{TR} (for training) and M_{VA} (for validation) using a $L/(1-L)$ split.
5: (b) Solve the inverse problem using M_{TR} data as input and obtain the parameter set: $\hat{\boldsymbol{\theta}} = \{\hat{\theta}_i\},\ \forall\ i \in \{1, ..., S\}$.
6: (c) Solve the forward problem using the prices from the M_{VA} set and the $\hat{\boldsymbol{\theta}}$'s to obtain the response set: $\{\hat{\boldsymbol{N}}|_m\},\ \forall\ m \in \{1, ..., M_{VA}\}$.
7: (d) Compare the generated data, $\{\hat{\boldsymbol{N}}|_m\},\ \forall\ m$, to the real data, $\{\boldsymbol{N}|_m\},\ \forall\ m$, using the following deviation metric:

$$d = \max_m \left[\max \left| \boldsymbol{N}|_m - \hat{\boldsymbol{N}}|_m \right| \right]. \tag{13}$$

8: **until** $d \leq \tau$

As for the aggregator, it wants to learn the coefficients $k_{i,1},\ \forall\ i \in [S]$, in order to estimate each DC's cost function; we assume that it already knows the exponential constituent functions along with the extra scaling parameters $E[A_i]\ \forall\ i$. The true parameter estimates are obtained by solving the inverse problems for NAP and AP using Algorithm 1. With this, each DC's cost function can be estimated and used in finding the best prices via the minimization of the social cost metric in (4). For our simulations, we consider the *ground truth* scenario where the aggregator has access to DC interaction data needed to solve the inverse problems and has successfully done so for both NAP/AP formulations by carrying out Algorithm 1 and, in turn, has correct knowledge of the true DC cost functions.

We consider two experimental settings where the two DCs are defined as: 1) DC 1: low arrival rate, low efficiency, DC 2: high arrival rate, low efficiency, and 2) DC 1: low arrival rate, high efficiency, DC 2: high arrival rate, low efficiency. Arrival rates and efficiencies are designed by changing the mean arrival values $\mu_A^{(i)}$ and the α_i parameters, respectively, for DC i.

Figures 3a and 3b show the social cost and penalty costs, respectively, for both NAP and AP formulations over various target reduction values for Setting 1. Both policies perform similarly and achieve most target load reductions up to a point as both DCs are willing to participate in the DR. The costs heighten at target values close to $T = 8000$ units of energy due to the limits of energy reduction over the two present DCs being reached. If S was larger, more reductions would be possible and higher T's could be reached. The resource sharing aspect in the AP is not taken advantage of here as the values of the foreign server counts for Setting 1 in Figure 3c are all close to 0; this is expected as there is nothing to gain in sharing workloads since both DCs are similar in efficiency.

Figures 4a and 4b show the social cost and penalty costs, respectively, for both NAP and AP formulations over various targets for Setting 2. We see that the AP outperforms the NAP policy in terms of achieving high target reduction values with smaller social and penalty costs; this is due to the aspect of resource sharing. Figure 4c shows the amount of shared servers in Setting 2 and we see that sharing is increasingly taken advantage of as the target becomes larger since it becomes advantageous for DC 2 to transfer workload to DC 1 due to the efficiency benefit DC 1 can provide. The shared server curve grows up until a specific point that signifies the limit of the amount of servers that can be shared between the two DCs.

Ultimately, we see from these experiments that both proposed policies are able to achieve most desired load reduction values; but, in certain settings (i.e., Setting 2) the AP outperforms its non-sharing counterpart NAP policy and achieves a larger range of target load reduction values at smaller social and penalty costs. The AP serves as a more robust policy due to the permission to share servers, albeit at the expense of a resource transportation cost, and is a practical DR policy that can be deployed for real-world resource control and management.

VII. Conclusion

In this work, we proposed two novel frameworks for DC DR pricing, modeled DC cost functions using QoS requirements, presented a data-driven inverse optimization approach for cost parameter estimation, and displayed experimental results showcasing the benefits of using a DR program in dealing with a supply deficit event along with highlighting situations where one policy would yield stronger performance over the other.

As future work, the two-sided problem where DCs could have the option of both increasing and/or decreasing consumption for the case of either a supply deficit or supply surplus using negative prices sent via the aggregator can be studied in addition to a possible inclusion of more realistic costs based on cooling, temperature, and other critical DC operating components in our formulations.

References

[1] I. Hussain, S. Mohsin, A. Basit, Z. A. Khan, U. Qasim, and N. Javid. A review on demand response: Pricing, optimization, and appliance scheduling. In *The 5th International Conference on Sustainable Energy Information Technology - SEIT 2015*, London, UK, June 2015.

[2] Q. Zhang and J. Li. Demand response in electricity markets: A review. In *International Conference on the European Energy Market*, Florence, Italy, May 2012.

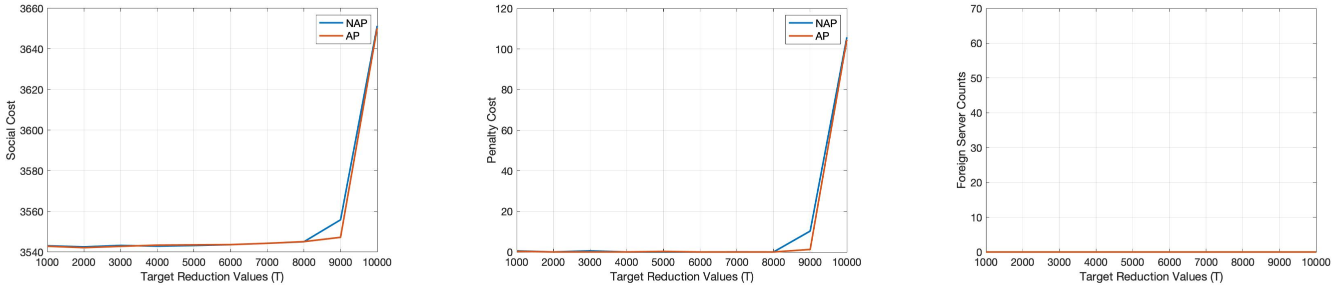
[3] P. D. Klemperer and M. A. Meyer. Supply function equilibria in oligopoly under uncertainty. *Econometrica*, 57(6):1243–1277, November 1989.

[4] R. Johari and J. N. Tsitsiklis. Parameterized supply function bidding: Equilibrium and efficiency. *Operations Research*, 59(5):1079–1089, September-October 2011.

[5] N. Li, L. Chen, and M. A. Dahleh. Demand response using linear supply function bidding. *IEEE Transactions on Smart Grid*, 6(4), July 2015.

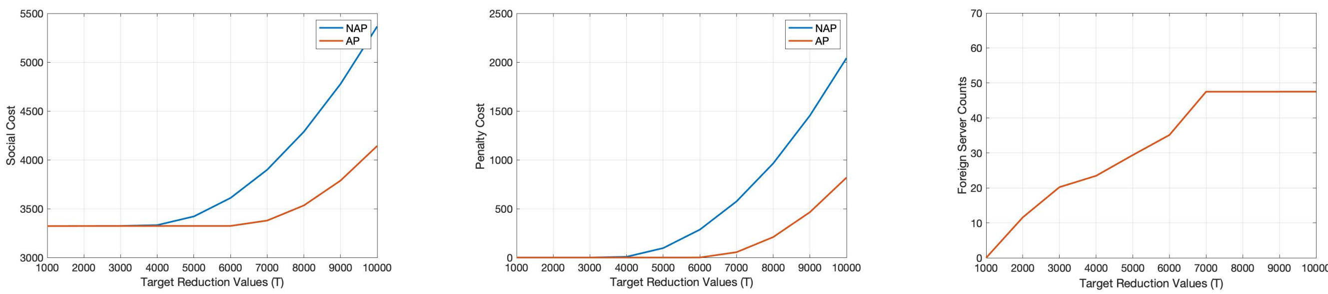
[6] L. Niu and Y. Guo. Enabling reliable data center demand response via aggregation. In *Seventh International Conference on Future Energy Systems - ACM e-Energy '16*, Waterloo, Ontario, Canada, June 2016.

[7] N. Chen, X. Ren, S. Ren, and A. Wierman. Greening multi-tenant data center demand response. *Performance Evaluation*, 43(2), September 2015.



(a) (b) (c)

Fig. 3: Experimental results for Setting 1. Both policies perform well and are able to achieve desired target load reductions with social and penalty costs close to zero up until very high T values where the $S = 2$ DCs cannot provide any more grid relief via their energy consumption reductions. There is no benefit in sharing resources since both DCs are similar in efficiencies.



(a) (b) (c)

Fig. 4: Experimental results for Setting 2. The AP outperforms the NAP in terms of social and penalty costs and can achieve higher target values at smaller social and penalty costs due to its sharing option, which is increasingly taken advantage of as there is a benefit in sharing resources. As seen in Setting 1, costs increase at very high targets due to the DR limits of only $S = 2$ DCs being reached.

[8] R. Chen, I. Ch. Paschalidis, M. C. Caramanis, and P. Andrianesis. Learning from past bids to participate strategically in day-ahead electricity markets. *IEEE Transactions on Smart Grid*, 2019. doi: 10.1109/TSG.2019.2891747.

[9] J. Saez-Gallego, J. M. Morales, M. Zugno, and H. Madsen. A data-driven bidding model for a cluster of price-responsive consumers of electricity. *IEEE Transactions on Power Systems*, 31(6), November 2016.

[10] J. Saez-Gallego and J. M. Morales. Short-term forecasting of price-responsive loads using inverse optimization. *IEEE Transactions on Smart Grid*, 9(5):4805–4814, September 2018.

[11] Z. Liu, I. Liu, S. Low, and A. Wierman. Pricing data center demand response. In *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Austin, TX, USA, June 2014.

[12] S. Kim and G. B. Giannakis. An online convex optimization approach to real-time energy pricing for demand response. *IEEE Transactions on Smart Grid*, 8(6), November 2017.

[13] L. Chen, N. Li, S. H. Low, and J. C. Doyle. Two market models for demand response in power networks. In *IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, USA, October 2010.

[14] Z. Liu, A. Wierman, Y. Chen, B. Razon, and N. Chen. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation*, 70(10), October 2013.

[15] N. Tran, S. Ren, Z. Han, S. M. Jang, S. Moon, and C. S. Hong. Demand response of data centers: A real-time pricing game between utilities in smart grid. In *9th International Workshop on Feedback Computing (Feedback Computing 14)*, Philadelphia, PA, 2014. USENIX Association.

[16] A. Tsiligkaridis, I. Ch. Paschalidis, and A. Coskun. Data center demand response pricing using inverse optimization. In *Tenth International Conference on Future Energy Systems - ACM e-Energy '19*, Phoenix, AZ, USA, June 2019.

[17] R. E. Brown et. al. Report to congress on server and data center energy efficiency: Public law 109-431. June 2008.

[18] G. Ghatikar, V. Ganti, N. Matson, and M. A. Piette. Demand response opportunities and enabling technologies for data centers: Findings from field studies. August 2012.

[19] J. Vidal. 'tsunami of data' could consume one fifth of global electricity by 2025. *Climate Home News*.

[20] Natural Resources Defense Council. Data center efficiency assessment. August 2014.

[21] J. M. Lima. Al Gore's 'an inconvenient sequel: Truth to power' lays it bare for data centers energy footprint. *Data Economy*.

[22] J. M. Lima. Data centres of the world will consume 1/5 of earth's power by 2025. *Data Economy*.

[23] Y. Sverdlik. Coronavirus is bad and good news for the data center switch market. *Data Center Knowledge*.

[24] S. Fulton III. Rules rewritten: Managing data centers through the pandemic. *Data Center Knowledge*.

[25] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad. Opportunities and challenges for data center demand response. In *International Green Computing Conference*, Dallas, TX, USA, November 2014.

[26] Ioannis Ch. Paschalidis. *Large Deviations in High Speed Communication Networks*. PhD thesis, Massachusetts Institute of Technology, 1996.

[27] D. Bertsimas, V. Gupta, and I. Ch. Paschalidis. Data-Driven Estimation in Equilibrium Using Inverse Optimization. *Mathematical Programming, Series A*, 153(2):595–633, 2015.