Optimal Power Cost Management Using Stored Energy in Data Centers

Rahul Urgaonkar, Bhuvan Urgaonkar†, Michael J. Neely‡, Anand Sivasubramaniam†
Advanced Networking Dept., Dept. of CSE†, Dept. of EE‡
Raytheon BBN Technologies, The Pennsylvania State University†, University of Southern California‡
Cambridge MA, University Park PA†, Los Angeles CA‡
rahul@bbn.com, {bhuvan,anand}@cse.psu.edu†, mjneely@usc.edu‡

ABSTRACT

Since the electricity bill of a data center constitutes a significant portion of its overall operational costs, reducing this has become important. We investigate cost reduction opportunities that arise by the use of uninterrupted power supply (UPS) units as energy storage devices. This represents a deviation from the usual use of these devices as mere transitional fail-over mechanisms between utility and captive sources such as diesel generators. We consider the problem of opportunistically using these devices to reduce the time average electric utility bill in a data center. Using the technique of Lyapunov optimization, we develop an online control algorithm that can optimally exploit these devices to minimize the time average cost. This algorithm operates without any knowledge of the statistics of the workload or electricity cost processes, making it attractive in the presence of workload and pricing uncertainties. An interesting feature of our algorithm is that its deviation from optimality reduces as the storage capacity is increased. Our work opens up a new area in data center power management.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques; Design studies

General Terms

Algorithms, Performance, Theory

Keywords

Power Management, Data Centers, Stochastic Optimization, Optimal Control

1. INTRODUCTION

Data centers spend a significant portion of their overall operational costs towards their electricity bills. As an example, one recent case study suggests that a large 15MW
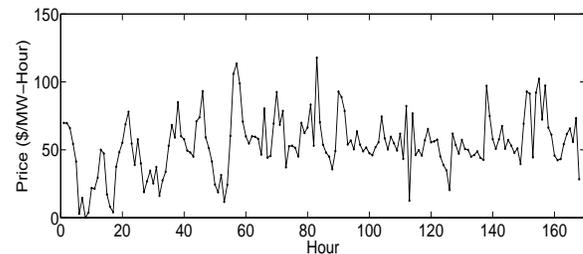
Figure 1: Avg. hourly spot market price during the week of 01/01/2005-01/07/2005 for LA1 Zone [1].

data center (on the more energy-efficient end) might spend about $1M on its monthly electricity bill. In general, a data center spends between 30-50% of its operational expenses towards power [10]. A large body of research addresses these expenses by reducing the energy consumption of these data centers. This includes designing/employing hardware with better power/performance trade-offs [9,17,20], software techniques for power-aware scheduling [12], workload migration, resource consolidation [6], among others. Power prices exhibit variations along time, space (geography), and even across utility providers. As an example, consider Fig. 1 that shows the average hourly spot market prices for the Los Angeles Zone LA1 obtained from CAISO [1]. These correspond to the week of 01/01/2005-01/07/2005 and denote the average price of 1 MW-Hour of electricity. Consequently, minimization of energy consumption need not coincide with that of the electricity bill.

Given the diversity within power price and availability, attention has recently turned towards *demand response* (DR) within data centers. DR within a data center (or a set of related data centers) attempts to optimize the electricity bill by adapting its needs to the temporal, spatial, and cross-utility diversity exhibited by power price. The key idea behind these techniques is to preferentially shift power draw (i) to times and places or (ii) from utilities offering cheaper prices. Typically some constraints in the form of performance requirements for the workload (e.g., response times offered to the clients of a Web-based application) limit the cost reduction benefits that can result from such DR. Whereas existing DR techniques have relied on various forms of workload scheduling/shifting, a complementary knob to facilitate such movement of power needs is offered by *energy storage devices*, typically uninterrupted power supply (UPS) units, residing in data centers.

A data center deploys captive power sources, typically diesel generators (DG), that it uses for keeping itself powered up when the utility experiences an outage. The UPS units serve as a bridging mechanism to facilitate this transition from utility to DG: upon a utility failure, the data center is kept powered by the UPS unit using energy stored within its batteries, before the DG can start up and provide power. Whereas this transition takes only 10-20 seconds, UPS units have enough battery capacity to keep the entire data center powered at its maximum power needs for anywhere between 5-30 minutes. Tapping into the energy reserves of the UPS unit can allow a data center to improve its electricity bill. Intuitively, the data center would store energy within the UPS unit when prices are low and use this to augment the draw from the utility when prices are high.

In this paper, we consider the problem of developing an online control policy to exploit the UPS unit along with the presence of delay-tolerance within the workload to optimize the data center's electricity bill. This is a challenging problem because data centers experience time-varying workloads and power prices with possibly unknown statistics. Even when statistics can be approximated (say by learning using past observations), traditional approaches to construct optimal control policies involve the use of Markov Decision Theory and Dynamic Programming [5]. It is well known that these techniques suffer from the "curse of dimensionality" where the complexity of computing the optimal strategy grows with the system size. Furthermore, such solutions result in hard-to-implement systems, where significant re-computation might be needed when statistics change.

In this work, we make use of a different approach that can overcome the challenges associated with dynamic programming. This approach is based on the recently developed technique of Lyapunov optimization [8] [15] that enables the design of online control algorithms for such time-varying systems. These algorithms operate *without requiring any knowledge of the system statistics* and are easy to implement. We design such an algorithm for optimally exploiting the UPS unit and delay-tolerance of workloads to minimize the time average cost. We show that our algorithm can get within $O(1/V)$ of the optimal solution where the maximum value of V is limited by battery capacity. We note that, for the same parameters, a dynamic programming based approach (if it can be solved) will yield a better result than our algorithm. However, this gap reduces as the battery capacity is increased. Our algorithm is thus most useful when such scaling is practical.

2. RELATED WORK

One recent body of work proposes online algorithms for using UPS units for cost reduction via shaving workload "peaks" that correspond to higher energy prices [3, 4]. This work is highly complementary to ours in that it offers a worst-case competitive ratio analysis while our approach looks at the average case performance. Whereas a variety of work has looked at workload shifting for power cost reduction [20] or other reasons such as performance and availability [6], our work differs both due to its usage of energy storage as well as the cost optimality guarantees offered by our technique. Some research has considered consumers with access to multiple utility providers, each with a different carbon profile, power price and availability and looked at optimizing cost subject to performance and/or carbon emissions
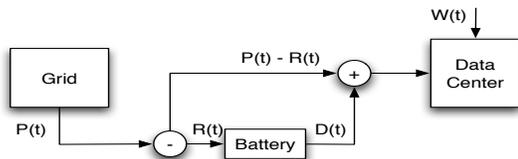

Figure 2: Block diagram for the basic model.

constraints [11]. Another line of work has looked at cost reduction opportunities offered by geographical variations within utility prices for data centers where portions of workloads could be serviced from one of several locations [11,18]. Finally, [7] considers the use of rechargeable batteries for maximizing system utility in a wireless network. While all of this research is highly complementary to our work, there are three key differences: (i) our investigation of energy storage as an enabler of cost reduction, (ii) our use of the technique of Lyapunov optimization which allows us to offer a provably cost optimal solution, and (iii) combining energy storage with delay-tolerance within workloads.

3. BASIC MODEL

We consider a time-slotted model. In the basic model, we assume that in every slot, the total power demand generated by the data center in that slot must be met in the current slot itself (using a combination of power drawn from the utility and the battery). Thus, any buffering of the workload generated by the data center is not allowed. We will relax this constraint later in Sec. 6 when we allow buffering of some of the workload while providing worst case delay guarantees. In the following, we use the terms UPS and battery interchangeably.

3.1 Workload Model

Let $W(t)$ be total workload (in units of power) generated in slot t. Let $P(t)$ be the total power drawn from the grid in slot t out of which $R(t)$ is used to recharge the battery. Also, let $D(t)$ be the total power discharged from the battery in slot t. Then in the basic model, the following constraint must be satisfied in every slot (Fig. 2):

$$W(t) = P(t) - R(t) + D(t) \qquad (1)$$

Every slot, a control algorithm observes $W(t)$ and makes decisions about how much power to draw from the grid in that slot, i.e., $P(t)$, and how much to recharge and discharge the battery, i.e., $R(t)$ and $D(t)$. Note that by (1), having chosen $P(t)$ and $R(t)$ completely determines $D(t)$.

Assumptions on the statistics of $W(t)$: The workload process $W(t)$ is assumed to vary randomly taking values from a set $\mathcal{W}$ of non-negative values and is not influenced by past control decisions. The set $\mathcal{W}$ is assumed to be finite, with potentially arbitrarily large size. The underlying probability distribution or statistical characterization of $W(t)$ is not necessarily known. We only assume that its maximum value is finite, i.e., $W(t) \leq W_{max}$ for all t.

For simplicity, in the basic model we assume that $W(t)$ evolves according to an i.i.d. process noting that the algorithm developed for this case can be applied without any modifications to non-i.i.d. scenarios as well. The analysis and performance guarantees for the non-i.i.d. case can be

obtained using the delayed Lyapunov drift and T slot drift techniques developed in [8] [15].

3.2 Battery Model

Ideally, we would like to incorporate the following idiosyncrasies of battery operation into our model. First, batteries become unreliable as they are charged/discharged, with higher depth-of-discharge (DoD) - percentage of maximum charge removed during a discharge cycle - causing faster degradation in their reliability. This dependence between the useful lifetime of a battery and how it is discharged/charged is expressed via battery lifetime charts [13]. For example, with lead-acid batteries that are commonly used in UPS units, 20% DoD yields 1400 cycles [2]. Second, batteries have conversion loss whereby a portion of the energy stored in them is lost when discharging them (e.g., about 10-15% for lead-acid batteries). Furthermore, certain regions of battery operation (high rate of discharge) are more inefficient than others. Finally, the storage itself maybe "leaky", so that the stored energy decreases over time, even in the absence of any discharging.

For simplicity, in the basic model we will assume that there is no power loss either in recharging or discharging the batteries, noting that this can be easily generalized to the case where a fraction of $R(t), D(t)$ is lost. We will also assume that the batteries are not leaky, so that the stored energy level decreases only when they are discharged. This is a reasonable assumption when the time scale over which the loss takes place is much larger than that of interest to us. To model the effect of repeated recharging and discharging on the battery's lifetime, we assume that with each recharge and discharge operation, a fixed cost (in dollars) of C_{rc} and C_{dc} respectively is incurred. The choice of these parameters would affect the trade-off between the cost of the battery itself and the cost reduction benefits it offers. For example, suppose a new battery costs B dollars and it can sustain N discharge/charge cycles (ignoring DoD for now). Then setting $C_{rc} = C_{dc} = B/N$ would amount to expecting the battery to "pay for itself" by augmenting the utility N times over its lifetime.

In any slot, we assume that one can either recharge or discharge the battery or do neither, but not both. This means that for all t, we have:

$$R(t) > 0 \implies D(t) = 0, \; D(t) > 0 \implies R(t) = 0 \quad (2)$$

Let $Y(t)$ denote the battery energy level in slot t. Then, the dynamics of $Y(t)$ can be expressed as:

$$Y(t+1) = Y(t) - D(t) + R(t) \quad (3)$$

The battery is assumed to have a finite capacity Y_{max} so that $Y(t) \leq Y_{max}$ for all t. Further, for the purpose of reliability, it may be required to ensure that a minimum energy level $Y_{min} \geq 0$ is maintained at all times. For example, this could represent the amount of energy required to support the data center operations until a secondary power source (such as DG) is activated in the event of a grid outage. Recall that the UPS unit is integral to the availability of power supply to the data center upon utility outage. Indiscriminate discharging of UPS can leave the data center in situations where it is unable to safely fail-over to DG upon a utility outage. Therefore, discharging the UPS must be done carefully so that it still possesses enough charge so reliably carry out its role as a transition device between utility

and DG. Thus, the following condition must be met in every slot under any feasible control algorithm:

$$Y_{min} \leq Y(t) \leq Y_{max} \quad (4)$$

The effectiveness of the online control algorithm we present in Sec. 5 will depend on the magnitude of the difference $Y_{max} - Y_{min}$. In most practical scenarios of interest, this value is expected to be at least moderately large: recent work suggests that storing energy Y_{min} to last about a minute is sufficient to offer reliable data center operation [14], while Y_{max} can vary between 5-20 minutes (or even higher) due to reasons such as UPS units being available only in certain sizes and the need to keep room for future IT growth. Furthermore, the UPS units are sized based on the *maximum provisioned* capacity of the data center, which is itself often substantially (up to twice [10]) higher than the maximum actual power demand.

The initial charge level in the battery is given by Y_{init} and satisfies $Y_{min} \leq Y_{init} \leq Y_{max}$. Finally, we assume that the maximum amounts by which we can recharge or discharge the battery in any slot are bounded. Thus, we have $\forall t$:

$$0 \leq R(t) \leq R_{max}, \; 0 \leq D(t) \leq D_{max} \quad (5)$$

We will assume that $Y_{max} - Y_{min} > R_{max} + D_{max}$ while noting that in practice, $Y_{max} - Y_{min}$ is much larger than $R_{max} + D_{max}$. Note that any feasible control decision on $R(t), D(t)$ must ensure that both of the constraints (4) and (5) are satisfied. This is equivalent to the following:

$$0 \leq R(t) \leq \min[R_{max}, Y_{max} - Y(t)] \quad (6)$$
$$0 \leq D(t) \leq \min[D_{max}, Y(t) - Y_{min}] \quad (7)$$

3.3 Cost Model

The cost per unit of power drawn from the grid in slot t is denoted by $C(t)$. In general, it can depend on both $P(t)$, the total amount of power drawn in slot t, and an auxiliary state variable $S(t)$, that captures parameters such as time of day, identity of the utility provider, etc. For example, the per unit cost may be higher during business hours, etc. Similarly, for any fixed $S(t)$, it may be the case that $C(t)$ increases with $P(t)$ so that per unit cost of electricity increases as more power is drawn. This may be because the utility provider wants to discourage heavier loading on the grid. Thus, we assume that $C(t)$ is a function of both $S(t)$ and $P(t)$ and we denote this as:

$$C(t) = \hat{C}(S(t), P(t)) \quad (8)$$

For notational convenience, we will use $C(t)$ to denote the per unit cost in the rest of the paper noting that the dependence of $C(t)$ on $S(t)$ and $P(t)$ is implicit.

The auxiliary state process $S(t)$ is assumed to evolve independently of the decisions taken by any control policy. For simplicity, we assume that every slot it takes values from a finite but arbitrarily large set $\mathcal{S}$ in an i.i.d. fashion according to a potentially unknown distribution. This can again be generalized to non i.i.d. Markov modulated scenarios using the techniques developed in [8] [15]. For each $S(t)$, the unit cost is assumed to be a non-decreasing function of $P(t)$. Note that it is not necessarily convex or strictly monotonic or continuous. This is quite general and can be used to model a variety of scenarios. A special case is when $C(t)$ is only a function of $S(t)$. The optimal control action for this

case has a particularly simple form and we will highlight this in Sec. 5.1.1. The unit cost is assumed to be non-negative and finite for all $S(t), P(t)$.

We assume that the maximum amount of power that can be drawn from the grid in any slot is upper bounded by P_{peak}. Thus, we have for all t:

$$0 \leq P(t) \leq P_{peak} \qquad (9)$$

Note that if we consider the original scenario where batteries are not used, then P_{peak} must be such that all workload can be satisfied. Therefore, $P_{peak} \geq W_{max}$.

Finally, let C_{max} and C_{min} denote the maximum and minimum per unit cost respectively over all $S(t), P(t)$. Also let $\chi_{min} > 0$ be a constant such that for any $P_1, P_2 \in [0, P_{peak}]$ where $P_1 \leq P_2$, the following holds for all $\chi \geq \chi_{min}$:

$$P_1(-\chi + C(P_1, S)) \geq P_2(-\chi + C(P_2, S)) \qquad \forall S \in \mathcal{S} \quad (10)$$

For example, when $C(t)$ does not depend on $P(t)$, then $\chi_{min} = C_{max}$ satisfies (10). This follows by noting that $(-C_{max} + C(t)) \leq 0$ for all t. Similarly, suppose $C(t)$ does not depend on $S(t)$, but is continuous, convex, and increasing in $P(t)$. Then, it can be shown that $\chi_{min} = C(P_{peak}) + P_{peak}C'(P_{peak})$ satisfies (10) where $C'(P_{peak})$ denotes the derivative of $C(t)$ evaluated at P_{peak}. In the following, we assume that such a finite χ_{min} exists for the given cost model. We further assume that $\chi_{min} > C_{min}$. The case of $\chi_{min} = C_{min}$ corresponds to the degenerate case where the unit cost is fixed for all times and we do not consider it in this paper.

What is known in each slot?: We assume that the value of $S(t)$ and the form of the function $C(P(t), S(t))$ for that slot is known. For example, this may be obtained beforehand using pre-advertised prices by the utility provider. We assume that given an $S(t) = s$, $C(t)$ is a deterministic function of $P(t)$ and this holds for all s. Similarly, the amount of incoming workload $W(t)$ is known at the beginning of each slot.

Given this model, our goal is to design a control algorithm that minimizes the time average cost while meeting all the constraints. This is formalized in the next section.

4. CONTROL OBJECTIVE

Let $P(t), R(t)$ and $D(t)$ denote the control decisions made in slot t by any feasible policy under the basic model as discussed in Sec. 3. These must satisfy the constraints (1), (2), (6), (7), and (9) every slot. We define the following indicator variables that are functions of the control decisions regarding a recharge or discharge operation in slot t:

$$1_R(t) = \begin{cases} 1 & \text{if } R(t) > 0 \\ 0 & \text{else} \end{cases} \quad 1_D(t) = \begin{cases} 1 & \text{if } D(t) > 0 \\ 0 & \text{else} \end{cases}$$

Note that by (2), at most one of $1_R(t)$ and $1_C(t)$ can take the value 1. Then the total cost incurred in slot t is given by $P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}$. The time-average cost under this policy is given by:

$$\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc}\right\} \quad (11)$$

where the expectation above is with respect to the potential randomness of the control policy. Assuming for the time being that this limit exists, our goal is to design a control algorithm that minimizes this time average cost subject to the constraints described in the basic model. Mathematically, this can be stated as the following *stochastic optimization problem*:

P1 :

Minimize: $\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc}\right\}$

Subject to: Constraints $(1), (2), (6), (7), (9)$

The finite capacity and underflow constraints (6), (7) make this a particularly challenging problem to solve even if the statistical descriptions of the workload and unit cost process are known. For example, the traditional approach based on Dynamic Programming [5] would have to compute the optimal control action for all possible combinations of the battery charge level and the system state $(S(t), W(t))$. Instead, we take an alternate approach based on the technique of Lyapunov optimization, taking the *finite* size queues constraint explicitly into account.

Note that a solution to the problem **P1** is a control policy that determines the sequence of feasible control decisions $P(t), R(t), D(t)$, to be used. Let ϕ_{opt} denote the value of the objective in problem **P1** under an optimal control policy. Define the time-average rate of recharge and discharge under any policy as follows:

$$\overline{R} = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{R(\tau)\right\}, \ \overline{D} = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{D(\tau)\right\} \quad (12)$$

Now consider the following problem:

P2 :

Minimize: $\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc}\right\}$

Subject to: Constraints $(1), (2), (5), (9)$

$$\overline{R} = \overline{D} \qquad (13)$$

Let $\hat{\phi}$ denote the value of the objective in problem **P2** under an optimal control policy. By comparing **P1** and **P2**, it can be shown that **P2** is less constrained than **P1**. Specifically, any feasible solution to **P1** would also satisfy **P2**. To see this, consider any policy that satisfies (6) and (7) for all t. This ensures that constraints (4) and (5) are always met by this policy. Then summing equation (3) over all $\tau \in \{0, 1, 2, \ldots, t-1\}$ under this policy and taking expectation of both sides yields:

$$\mathbb{E}\left\{Y(t)\right\} - Y_{init} = \sum_{\tau=0}^{t-1} \mathbb{E}\left\{R(\tau) - D(\tau)\right\}$$

Since $Y_{min} \leq Y(t) \leq Y_{max}$ for all t, dividing both sides by t and taking limits as $t \to \infty$ yields $\overline{R} = \overline{D}$. Thus, this policy satisfies constraint (13) of **P2**. Therefore, any feasible solution to **P1** also satisfies **P2**. This implies that the optimal value of **P2** cannot exceed that of **P1**, so that $\hat{\phi} \leq \phi_{opt}$.

Our approach to solving **P1** will be based on this observation. We first note that it is easier to characterize the optimal solution to **P2**. This is because the dependence on $Y(t)$ has been removed. Specifically, it can be shown that the optimal solution to **P2** can be achieved by a station-

ary, randomized control policy that chooses control actions $P(t), D(t), R(t)$ every slot purely as a function (possibly randomized) of the current state $(W(t), S(t))$ and *independent of* the battery charge level $Y(t)$. This fact is presented in the following lemma:

LEMMA 1. *(Optimal Stationary, Randomized Policy): If the workload process $W(t)$ and auxiliary process $S(t)$ are i.i.d. over slots, then there exists a stationary, randomized policy that takes control decisions $P^{stat}(t), R^{stat}(t), D^{stat}(t)$ every slot purely as a function (possibly randomized) of the current state $(W(t), S(t))$ while satisfying the constraints (1), (2), (5), (9) and providing the following guarantees:*

$$\mathbb{E}\left\{R^{stat}(t)\right\} = \mathbb{E}\left\{D^{stat}(t)\right\} \qquad (14)$$

$$\mathbb{E}\left\{P^{stat}(t)C(t) + 1_R^{stat}(t)C_{rc} + 1_D^{stat}(t)C_{dc}\right\} = \hat{\phi} \qquad (15)$$

where the expectations above are with respect to the stationary distribution of $(W(t), S(t))$ and the randomized control decisions.

PROOF. This result follows from the framework in [8,15] and is omitted for brevity. □

It should be noted that while it is possible to characterize and potentially compute such a policy, it may not be feasible for the original problem **P1** as it could violate the constraints (6) and (7). However, the existence of such a policy can be used to construct an approximately optimal policy that meets all the constraints of **P1** using the technique of Lyapunov optimization [8] [15]. This policy is dynamic and does not require knowledge of the statistical description of the workload and cost processes. We present this policy and derive its performance guarantees in the next section. This dynamic policy is approximately optimal where the approximation factor improves as the battery capacity increases. Also note that the distance from optimality for our policy is measured in terms of $\hat{\phi}$. However, since $\hat{\phi} \le \phi_{opt}$, in practice, the approximation factor is better than the analytical bounds.

5. OPTIMAL CONTROL ALGORITHM

We now present an *online* control algorithm that approximately solves **P1**. This algorithm uses a control parameter $V > 0$ that affects the distance from optimality as shown later. This algorithm also makes use of a "queueing" state variable $X(t)$ to track the battery charge level and is defined as follows:

$$X(t) = Y(t) - V\chi_{min} - D_{max} - Y_{min} \qquad (16)$$

Recall that $Y(t)$ denotes the actual battery charge level in slot t and evolves according to (3). It can be seen that $X(t)$ is simply a shifted version of $Y(t)$ and its dynamics is given by:

$$X(t+1) = X(t) - D(t) + R(t) \qquad (17)$$

Note that $X(t)$ can be negative. We will show that this definition enables our algorithm to ensure that the constraint (4) is met.

We are now ready to state the dynamic control algorithm. Let $(W(t), S(t))$ and $X(t)$ denote the system state in slot t. Then the dynamic algorithm chooses control action $P(t)$ as
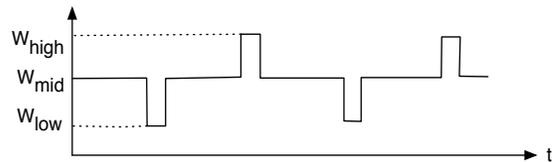

Figure 3: Periodic $W(t)$ process in the example.

the solution to the following optimization problem:

P3 :

Minimize: $X(t)P(t) + V\left[P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}\right]$

Subject to: Constraints $(1), (2), (5), (9)$

The constraints above result in the following constraint on $P(t)$:

$$P_{low} \le P(t) \le P_{high} \qquad (18)$$

where
$P_{low} = \max[0, W(t) - D_{max}]$ and $P_{high} = \min[P_{peak}, W(t) + R_{max}]$. Let $P^*(t), R^*(t)$, and $D^*(t)$ denote the optimal solution to **P3**. Then, the dynamic algorithm chooses the recharge and discharge values as follows.

$$R^*(t) = \begin{cases} P^*(t) - W(t) & \text{if } P^*(t) > W(t) \\ 0 & \text{else} \end{cases}$$

$$D^*(t) = \begin{cases} W(t) - P^*(t) & \text{if } P^*(t) < W(t) \\ 0 & \text{else} \end{cases}$$

Note that if $P^*(t) = W(t)$, then both $R^*(t) = 0$ and $D^*(t) = 0$ and all demand is met using power drawn from the grid. It can be seen from the above that the control decisions satisfy the constraints $0 \le R^*(t) \le R_{max}$ and $0 \le D^*(t) \le D_{max}$. That the finite battery constraints and the constraints (6), (7) are also met will be shown in Sec. 5.3.

After computing these quantities, the algorithm implements them and updates the queueing variable $X(t)$ according to (17). This process is repeated every slot. Note that in solving **P3**, the control algorithm only makes use of the current system state values and does not require knowledge of the statistics of the workload or unit cost processes. Thus, it is *myopic* and *greedy* in nature. From **P3**, it is seen that the algorithm tries to recharge the battery when $X(t)$ is negative and per unit cost is low. And it tries to discharge the battery when $X(t)$ is positive. That this is sufficient to achieve optimality will be shown in Theorem 1. The queueing variable $X(t)$ plays a crucial role as making decisions purely based on prices is not necessarily optimal.

To get some intuition behind the working of this algorithm, consider the following simple example. Suppose $W(t)$ can take three possible values from the set $\{W_{low}, W_{mid}, W_{high}\}$ where $W_{low} < W_{mid} < W_{high}$. Similarly, $C(t)$ can take three possible values in $\{C_{low}, C_{mid}, C_{high}\}$ where $C_{low} < C_{mid} < C_{high}$ and does not depend on $P(t)$. We assume that the workload process evolves in a frame-based periodic fashion. Specifically, in every odd numbered frame, $W(t) = W_{mid}$ for all except the last slot of the frame when $W(t) = W_{low}$. In every even numbered frame, $W(t) = W_{mid}$ for all except the last slot of the frame when $W(t) = W_{high}$. This is il-

Y_{max}	20	30	40	50	75	100
V	0	1.25	2.5	3.75	6.875	10.0
Avg. Cost	94.0	92.5	91.1	88.5	88.0	87.0

Table 1: Average Cost vs. Y_{max}

lustrated in Fig. 3. The $C(t)$ process evolves similarly, such that $C(t) = C_{low}$ when $W(t) = W_{low}$, $C(t) = C_{mid}$ when $W(t) = W_{mid}$, and $C(t) = C_{high}$ when $W(t) = W_{high}$.

In the following, we assume a frame size of 5 slots with $W_{low} = 10$, $W_{mid} = 15$, and $W_{high} = 20$ units. Also, $C_{low} = 2$, $C_{mid} = 6$, and $C_{high} = 10$ dollars. Finally, $R_{max} = D_{max} = 10$, $P_{peak} = 20$, $C_{rc} = C_{dc} = 5$, $Y_{init} = Y_{min} = 0$ and we vary $Y_{max} > R_{max} + D_{max}$. In this example, intuitively, an optimal algorithm that knows the workload and unit cost process beforehand would recharge the battery as much as possible when $C(t) = C_{low}$ and discharge it as much as possible when $C(t) = C_{high}$. In fact, it can be shown that the following strategy is feasible and achieves minimum average cost:

- If $C(t) = C_{low}, W(t) = W_{low}$, then $P(t) = W_{low} + R_{max}$, $R(t) = R_{max}$, $D(t) = 0$.

- If $C(t) = C_{mid}, W(t) = W_{mid}$, then $P(t) = W_{mid}$, $R(t) = 0$, $D(t) = 0$.

- If $C(t) = C_{high}, W(t) = W_{high}$, then $P(t) = W_{high} - D_{max}$, $R(t) = 0$, $D(t) = D_{max}$.

The time average cost resulting from this strategy can be easily calculated and is given by 87.0 dollars/slot for all $Y_{max} > 10$. Also, we note that the cost resulting from an algorithm that does not use the battery in this example is given by 94.0 dollars/slot.

Now we simulate the dynamic algorithm for this example for different values of Y_{max} for 1000 slots (200 frames). The value of V is chosen to be $\frac{Y_{max} - Y_{min} - R_{max} - D_{max}}{C_{high} - C_{low}} = \frac{Y_{max} - 20}{8}$ (this choice will become clear in Sec. 5.2 when we relate V to the battery capacity). Note that the number of slots for which a fully charged battery can sustain the data center at maximum load is Y_{max}/W_{high}.

In Table 1, we show the time average cost achieved for different values of Y_{max}. It can be seen that as Y_{max} increases, the time average cost approaches the optimal value (this behavior will be formalized in Theorem 1). This is remarkable given that the dynamic algorithm operates without any knowledge of the future workload and cost processes. To examine the behavior of the dynamic algorithm in more detail, we fix $Y_{max} = 100$ and look at the sample paths of the control decisions taken by the optimal offline algorithm and the dynamic algorithm during the first 200 slots. This is shown in Figs. 4 and 5. It can be seen that initially, the dynamic tends to perform suboptimally. But eventually it *learns* to make close to optimal decisions.

It might be tempting to conclude from this example that an algorithm based on a price threshold is optimal. Specifically, such an algorithm makes a recharge vs. discharge decision depending on whether the current price $C(t)$ is smaller or larger than a threshold. However, it is easy to construct examples where the dynamic algorithm outperforms such a threshold based algorithm. Specifically, suppose that the $W(t)$ process takes values from the interval $[10, 90]$ uniformly at random every slot. Also, suppose
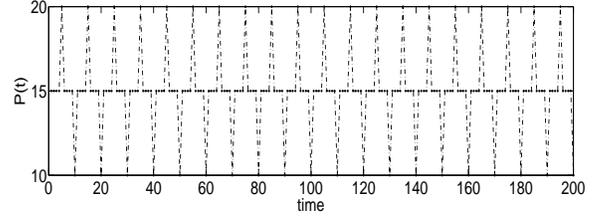

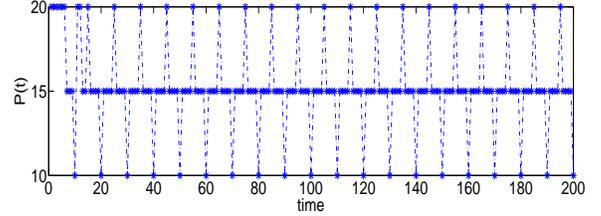**Figure 4:** $P(t)$ **under the offline optimal solution with** $Y_{max} = 100$**.**

Figure 5: $P(t)$ **under the Dynamic Algorithm with** $Y_{max} = 100$**.**

$C(t)$ takes values from the set $\{2, 6, 10\}$ dollars uniformly at random every slot. We fix the other parameters as follows: $R_{max} = D_{max} = 10$, $P_{peak} = 90$, $C_{rc} = C_{dc} = 1$, $Y_{init} = Y_{min} = 0$ and $Y_{max} = 100$. We then simulate a threshold based algorithm for different values of the threshold in the set $\{2, 6, 10\}$ and select the one that yields the smallest cost. This was found to be 280.7 dollars/slot. We then simulate the dynamic algorithm for 10000 slots with $V = \frac{Y_{max} - 20}{10 - 2} = 10.0$ and it yields an average cost of 275.5 dollars/slot. We also note that the cost resulting from an algorithm that does not use the battery in this example is given by 300.73 dollars/slot.

We now establish two properties of the structure of the optimal solution to **P3** that will be useful in analyzing its performance later.

LEMMA 2. *The optimal solution to **P3** has the following properties:*

1. *If $X(t) > -VC_{min}$, then the optimal solution always chooses $R^*(t) = 0$.*

2. *If $X(t) < -V\chi_{min}$, then the optimal solution always chooses $D^*(t) = 0$.*

PROOF. See [19]. □

5.1 Solving P3

In general, the complexity of solving **P3** depends on the structure of the unit cost function $C(t)$. For many cases of practical interest, **P3** is easy to solve and admits closed form solutions that can be implemented in real time. We consider two such cases here. Let $\theta(t)$ denote the value of the objective in **P3** when there is no recharge or discharge. Thus $\theta(t) = W(t)(X(t) + VC(t))$.

5.1.1 $C(t)$ *does not depend on* $P(t)$

Suppose that $C(t)$ depends only on $S(t)$ and not on $P(t)$. We can rewrite the expression in the objective of **P3** as

$P(t)(X(t) + VC(t)) + 1_R(t)VC_{rc} + 1_D(t)VC_{dc}$. Then, the optimal solution has the following simple threshold structure.

1. If $X(t) + VC(t) > 0$, then $R^*(t) = 0$ so that there is no recharge and we have the following two cases:

   (a) If $P_{low}(X(t) + VC(t)) + VC_{dc} < \theta(t)$, then discharge as much as possible, so that we get $D^*(t) = \min[W(t), D_{max}]$, $P^*(t) = \max[0, W(t) - D_{max}]$.

   (b) Else, draw all power from the grid. This yields $D^*(t) = 0$ and $P^*(t) = W(t)$.

2. Else if $X(t) + VC(t) \leq 0$, then $D^*(t) = 0$ so that there is no discharge and we have the following two cases:

   (a) If $P_{high}(X(t) + VC(t)) + VC_{rc} < \theta(t)$, then recharge as much as possible. This yields $R^*(t) = \min[P_{peak} - W(t), R_{max}]$ and $P^*(t) = \min[P_{peak}, W(t) + R_{max}]$.

   (b) Else, draw all power from the grid. This yields $R^*(t) = 0$ and $P^*(t) = W(t)$.

We will show that this solution is feasible and does not violate the finite battery constraint in Sec. 5.3.

5.1.2 $C(t)$ convex, increasing in $P(t)$

Next suppose for each $S(t)$, $C(t)$ is convex and increasing in $P(t)$. For example, $\hat{C}(S(t), P(t))$ may have the form $\alpha(S(t))P^2(t)$ where $\alpha(S(t)) > 0$ for all $S(t)$. In this case, **P3** becomes a standard convex optimization problem in a single variable $P(t)$ and can be solved efficiently. The full solution is provided in [19].

5.2 Performance Theorem

We first define an upper bound V_{max} on the maximum value that V can take in our algorithm.

$$V_{max} \triangleq \frac{Y_{max} - Y_{min} - R_{max} - D_{max}}{\chi_{min} - C_{min}} \quad (19)$$

Then we have the following result.

THEOREM 1. *(Algorithm Performance) Suppose the initial battery charge level Y_{init} satisfies $Y_{min} \leq Y_{init} \leq Y_{max}$. Then implementing the algorithm above with any fixed parameter V such that $0 < V \leq V_{max}$ for all $t \in \{0, 1, 2, \ldots\}$ results in the following performance guarantees:*

1. *The queue $X(t)$ is deterministically upper and lower bounded for all t as follows:*

$$-V\chi_{min} - D_{max} \leq X(t) \leq Y_{max} - Y_{min}$$
$$- D_{max} - V\chi_{min} \quad (20)$$

2. *The actual battery level $Y(t)$ satisfies $Y_{min} \leq Y(t) \leq Y_{max}$ for all t.*

3. *All control decisions are feasible.*

4. *If $W(t)$ and $S(t)$ are i.i.d. over slots, then the time-average cost under the dynamic algorithm is within B/V of the optimal value:*

$$\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc}\}$$
$$\leq \phi_{opt} + B/V \quad (21)$$

*where B is a constant given by $B = \frac{\max[R_{max}^2, D_{max}^2]}{2}$ and ϕ_{opt} is the optimal solution to **P1** under any feasible control algorithm (possibly with knowledge of future events).*

Theorem 1 part 4 shows that by choosing larger V, the time-average cost under the dynamic algorithm can be pushed closer to the minimum possible value ϕ_{opt}. However, V_{max} limits how large V can be chosen. We prove Theorem 1 in the next section.

5.3 Proof of Theorem 1

PROOF. (Theorem 1 part 1) We first show that (20) holds for $t = 0$. We have that

$$Y_{min} \leq Y(0) = Y_{init} \leq Y_{max} \quad (22)$$

Using the definition (16), we have that $Y(0) = X(0) + V\chi_{min} + D_{max} + Y_{min}$. Using this in (22), we get:

$$Y_{min} \leq X(0) + V\chi_{min} + D_{max} + Y_{min} \leq Y_{max}$$

This yields

$$-V\chi_{min} - D_{max} \leq X(0) \leq Y_{max} - Y_{min} - D_{max} - V\chi_{min}$$

Now suppose (20) holds for slot t. We will show that it also holds for slot $t + 1$. First, suppose $-VC_{min} < X(t) \leq Y_{max} - Y_{min} - D_{max} - V\chi_{min}$. Then, from Lemma 2, we have that $R^*(t) = 0$. Thus, using (17), we have that $X(t + 1) \leq X(t) \leq Y_{max} - Y_{min} - D_{max} - V\chi_{min}$. Next, suppose $X(t) \leq -VC_{min}$. Then, the maximum possible increase is R_{max} so that $X(t + 1) \leq -VC_{min} + R_{max}$. Now for all V such that $0 < V \leq V_{max}$, we have that $-VC_{min} + R_{max} \leq Y_{max} - Y_{min} - D_{max} - V\chi_{min}$. This follows from the definition (19) and the fact that $\chi_{min} > C_{min}$. Thus, we have $X(t + 1) \leq Y_{max} - Y_{min} - D_{max} - V\chi_{min}$.

Next, suppose $-V\chi_{min} - D_{max} \leq X(t) < -V\chi_{min}$. Then, from Lemma 2, we have that $D^*(t) = 0$. Thus, using (17) we have that $X(t + 1) \geq X(t) \geq -V\chi_{min} - D_{max}$. Next, suppose $-V\chi_{min} \leq X(t)$. Then the maximum possible decrease is D_{max} so that $X(t + 1) \geq -V\chi_{min} - D_{max}$ for this case as well. This shows that $X(t + 1) \geq -V\chi_{min} - D_{max}$. Combining these two bounds proves (20). □

PROOF. (Theorem 1 parts 2 and 3) Part 2 directly follows from (20) and (16). Using $Y(t) = X(t) + V\chi_{min} + D_{max} + Y_{min}$ in the lower bound in (20), we have: $-V\chi_{min} - D_{max} \leq Y(t) - V\chi_{min} - D_{max} - Y_{min}$, i.e., $Y_{min} \leq Y(t)$. Similarly, using $Y(t) = X(t) + V\chi_{min} + D_{max} + Y_{min}$ in the upper bound in (20), we have: $Y(t) - V\chi_{min} - D_{max} - Y_{min} \leq Y_{max} - Y_{min} - D_{max} - V\chi_{min}$, i.e., $Y(t) \leq Y_{max}$.

Part 3 now follows from part 2 and the constraint on $P(t)$ in **P3**. □

PROOF. (Theorem 1 part 4) We make use of the technique of Lyapunov optimization to show (21). We start by defining a Lyapunov function as a scalar measure of congestion in the system. Specifically, we define the following Lyapunov function: $L(X(t)) \triangleq \frac{1}{2}X^2(t)$. Define the conditional 1-slot Lyapunov drift as follows:

$$\Delta(X(t)) \triangleq \mathbb{E}\{L(X(t+1)) - L(X(t))|X(t)\} \quad (23)$$

Using (17), $\Delta(X(t))$ can be bounded as follows (see [19] for details):

$$\Delta(X(t)) \leq B - X(t)\mathbb{E}\{D(t) - R(t)|X(t)\} \quad (24)$$

where $B = \frac{\max[R_{max}^2, D_{max}^2]}{2}$. Following the Lyapunov optimization framework of [8], we add to both sides of (24) the penalty term $V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}|X(t)\}$ to get the following:

$$\Delta(X(t)) + V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}|X(t)\}$$
$$\leq B - X(t)\mathbb{E}\{D(t) - R(t)|X(t)\}$$
$$+ V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}|X(t)\} \quad (25)$$

Using (1), we can rewrite the above as:

$$\Delta(X(t)) + V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}|X(t)\} \leq$$
$$B - X(t)\mathbb{E}\{W(t)|X(t)\} + X(t)\mathbb{E}\{P(t)|X(t)\}$$
$$+ V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}|X(t)\} \quad (26)$$

Comparing this with **P3**, it can be seen that given any queue value $X(t)$, our control algorithm is designed to *minimize* the right hand side of (26) over all possible feasible control policies. This includes the optimal, stationary, randomized policy given in Lemma 1. Then, plugging the control decisions corresponding to the stationary, randomized policy, the following holds for the dynamic algorithm:

$$\Delta(X(t)) + V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}|X(t)\} \leq$$
$$B + V\mathbb{E}\{P^{stat}(t)C^{stat}(t) + 1_R^{stat}(t)C_{rc} + 1_D^{stat}(t)C_{dc}|X(t)\}$$
$$= B + V\hat{\phi} \leq B + V\phi_{opt}$$

Taking the expectation of both sides and using the law of iterated expectations and summing over $t \in \{0, 1, 2, \ldots, T-1\}$, we get:

$$\sum_{t=0}^{T-1} V\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}\} \leq$$
$$BT + VT\phi_{opt} - \mathbb{E}\{L(X(T))\} + \mathbb{E}\{L(X(0))\}$$

Dividing both sides by VT and taking limit as $T \to \infty$ yields:

$$\lim_{T \to \infty} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc}\} \leq \phi_{opt} + B/V$$

where we have used the fact that $\mathbb{E}\{L(X(0))\}$ is finite and that $\mathbb{E}\{L(X(T))\}$ is non-negative. $\square$

6. EXTENSIONS TO BASIC MODEL

In this section, we extend the basic model of Sec. 3 to the case where portions of the workload are delay-tolerant in the sense they can be postponed by a certain amount without affecting the utility the data center derives from executing them. We refer to such postponement as buffering the workload. Specifically, we assume that the total workload consists of both delay tolerant and delay intolerant components. Similar to the workload in the basic model, the delay intolerant workload cannot be buffered and must be served immediately. However, the delay tolerant component may be buffered and served later. As an example, data centers run virus scanning programs on most of their servers routinely (say once per day). As long as a virus scan is executed once a day, their purpose is served - it does not matter what time of the day is chosen for this. The ability to delay some of the workload gives more opportunities to reduce the average power cost in addition to using the battery. We assume

that our data center has system mechanisms to implement such buffering of specified workloads.

In the following, we will denote the total workload generated in slot t by $W(t)$. This consists of the delay tolerant and intolerant components denoted by $W_1(t)$ and $W_2(t)$ respectively, so that $W(t) = W_1(t) + W_2(t)$ for all t. Similar to the basic model, we use $P(t), R(t), D(t)$ to denote the total power drawn from the grid, the total power used to recharge the battery and the total power discharged from the battery in slot t, respectively. Thus, the total amount available to serve the workload is given by $P(t) - R(t) + D(t)$. Let $\gamma(t)$ denote the fraction of this that is used to serve the delay tolerant workload in slot t. Then the amount used to serve the delay intolerant workload is $(1 - \gamma(t))(P(t) - R(t) + D(t))$. Note that the following constraint must be satisfied every slot:

$$0 \leq \gamma(t) \leq 1 \quad (27)$$

We next define $U(t)$ as the unfinished work for the delay tolerant workload in slot t. The dynamics for $U(t)$ can be expressed as:

$$U(t+1) = \max[U(t) - \gamma(t)(P(t) - R(t) + D(t)), 0] + W_1(t) \quad (28)$$

We assume that $U(t)$ is served in FIFO order. For the delay intolerant workload, there are no such queues since all incoming workload must be served in the same slot. This means:

$$W_2(t) = (1 - \gamma(t))(P(t) - R(t) + D(t)) \quad (29)$$

The block diagram for this extended model is shown in Fig. 6. Similar to the basic model, we assume that for $i = 1, 2$, $W_i(t)$ varies randomly in an i.i.d. fashion, taking values from a set $\mathcal{W}_i$ of non-negative values. We assume that $W_1(t) + W_2(t) \leq W_{max}$ for all t. We also assume that $W_1(t) \leq W_{1,max} < W_{max}$ and $W_2(t) \leq W_{2,max} < W_{max}$ for all t. We further assume that $P_{peak} \geq W_{max} + \max[R_{max}, D_{max}]$. We use the same model for battery and unit cost as in Sec. 3.

Our objective is to minimize the time-average cost subject to meeting all the constraints (such as finite battery size and (29)) and ensuring finite average delay for the delay tolerant workload. This can be stated as:

P4 :

Minimize: $\lim_{t \to \infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc}\}$

Subject to: Constraints $(2), (5), (6), (7), (9), (27), (29)$
Finite average delay for $W_1(t)$

Similar to the basic model, we consider the following *relaxed* problem:

P5 :

Minimize: $\lim_{t \to \infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc}\}$

Subject to: Constraints $(2), (5), (9), (27), (29)$

$$\overline{R} = \overline{D} \quad (30)$$
$$\overline{U} < \infty \quad (31)$$

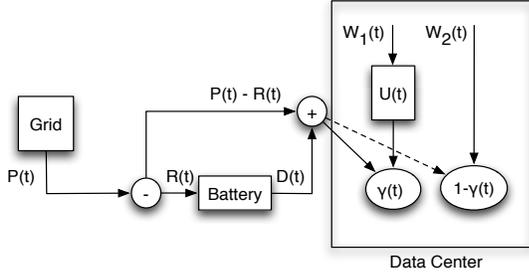where $\overline{U}$ is the time average expected queue backlog for the

Figure 6: Block diagram for the extended model with delay tolerant and delay intolerant workloads.

delay tolerant workload and is defined as:

$$\overline{U} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{U(\tau)\} \quad (32)$$

Let ϕ_{ext} and $\hat{\phi}_{ext}$ denote the optimal value for problems **P4** and **P5** respectively. Since **P5** is less constrained than **P4**, we have that $\hat{\phi}_{ext} \leq \phi_{ext}$. Similar to Lemma 1, the following holds:

LEMMA 3. *(Optimal Stationary, Randomized Policy): If the workload process $W_1(t), W_2(t)$ and auxiliary process $S(t)$ are i.i.d. over slots, then there exists a stationary, randomized policy that takes control decisions $\hat{P}(t), \hat{R}(t), \hat{D}(t), \hat{\gamma}(t)$ every slot purely as a function (possibly randomized) of the current state $(W_1(t), W_2(t), S(t))$ while satisfying the constraints (29), (2), (5), (9), (27) and providing the following guarantees:*

$$\mathbb{E}\left\{\hat{R}(t)\right\} = \mathbb{E}\left\{\hat{D}(t)\right\} \quad (33)$$

$$\mathbb{E}\left\{\hat{\gamma}(t)(\hat{P}(t) - \hat{R}(t) + \hat{D}(t))\right\} \geq \mathbb{E}\{W_1(t)\} \quad (34)$$

$$\mathbb{E}\left\{\hat{P}(t)\hat{C}(t) + \hat{1}_R(t)C_{rc} + \hat{1}_D(t)C_{dc}\right\} = \hat{\phi}_{ext} \quad (35)$$

where the expectations above are with respect to the stationary distribution of $(W_1(t), W_2(t), S(t))$ and the randomized control decisions.

PROOF. This result follows from the framework in [8,15] and is omitted for brevity. □

The condition (34) only guarantees queueing stability, not bounded worst case delay. We will now design a dynamic control algorithm that will yield bounded worst case delay while guaranteeing an average cost that is within $O(1/V)$ of $\hat{\phi}_{ext}$ (and therefore ϕ_{ext}).

6.1 Delay-Aware Queue

In order to provide worst case delay guarantees to the delay tolerant workload, we will make use of the technique of ϵ-persistent queue [16]. Specifically, we define a virtual queue $Z(t)$ as follows:

$$Z(t+1) = [Z(t) - \gamma(t)(P(t) - R(t) + D(t)) + \epsilon 1_{U(t)}]^+ \quad (36)$$

where $\epsilon > 0$ is a parameter to be specified later, $1_{U(t)}$ is an indicator variable that is 1 if $U(t) > 0$ and 0 otherwise, and $[x]^+ = \max[x, 0]$. The objective of this virtual queue

is to enable the provision of worst-case delay guarantee on any buffered workload $W_1(t)$. Specifically, if any control algorithm ensures that $U(t) \leq U_{max}$ and $Z(t) \leq Z_{max}$ for all t, then the worst case delay can be bounded. This is shown in the following:

LEMMA 4. *(Worst Case Delay) Suppose a control algorithm ensures that $U(t) \leq U_{max}$ and $Z(t) \leq Z_{max}$ for all t, where U_{max} and Z_{max} are some positive constants. Then the worst case delay for any delay tolerant workload is at most δ_{max} slots where:*

$$\delta_{max} \triangleq \lceil (U_{max} + Z_{max})/\epsilon \rceil \quad (37)$$

PROOF. Consider a new arrival $W_1(t)$ in any slot t. We will show that this is served on or before time $t + \delta_{max}$. We argue by contradiction. Suppose this workload is not served by $t + \delta_{max}$. Then for all slots $\tau \in \{t+1, t+2, \ldots, t+\delta_{max}\}$, it must be the case that $U(\tau) > 0$ (else $W_1(t)$ would have been served before τ). This implies that $1_{U(\tau)} = 1$ and using (36), we have:

$$Z(\tau + 1) \geq Z(\tau) - \gamma(\tau)(P(\tau) - R(\tau) + D(\tau)) + \epsilon$$

Summing for all $\tau \in \{t+1, t+2, \ldots, t+\delta_{max}\}$, we get:

$$Z(t + \delta_{max} + 1) - Z(t+1) \geq \delta_{max}\epsilon$$
$$- \sum_{\tau=t+1}^{t+\delta_{max}} [\gamma(\tau)(P(\tau) - R(\tau) + D(\tau))]$$

Using the fact that $Z(t+\delta_{max}+1) \leq Z_{max}$ and $Z(t+1) \geq 0$, we get:

$$\sum_{\tau=t+1}^{t+\delta_{max}} [\gamma(\tau)(P(\tau) - R(\tau) + D(\tau))] \geq \delta_{max}\epsilon - Z_{max} \quad (38)$$

Note that by (28), $W_1(t)$ is part of the backlog $U(t+1)$. Since $U(t+1) \leq U_{max}$ and since the service is FIFO, it will be served on or before time $t + \delta_{max}$ whenever at least U_{max} units of power is used to serve the delay tolerant workload during the interval $(t+1, \ldots, t+\delta_{max})$. Since we have assumed that $W_1(t)$ is not served by $t + \delta_{max}$, it must be the case that $\sum_{\tau=t+1}^{t+\delta_{max}} [\gamma(\tau)(P(\tau) - R(\tau) + D(\tau))] < U_{max}$. Using this in (38), we have:

$$U_{max} > \delta_{max}\epsilon - Z_{max}$$

This implies that $\delta_{max} < (U_{max} + Z_{max})/\epsilon$, that contradicts the definition of δ_{max} in (37). □

In Sec. 6.4, we will show that under the dynamic control algorithm (to be presented next), there are indeed constants U_{max}, Z_{max} such that $U(t) \leq U_{max}, Z(t) \leq Z_{max}$ for all t.

6.2 Optimal Control Algorithm

We now present an online control algorithm that approximately solves **P4**. Similar to the algorithm for the basic model, this algorithm also makes use of the following queueing state variable $X(t)$ to track the battery charge level and is defined as follows:

$$X(t) = Y(t) - Q_{max} - D_{max} - Y_{min} \quad (39)$$

where Q_{max} is a constant to be specified in (44). Recall that $Y(t)$ denotes the actual battery charge level in slot t

and evolves according to (3). It can be seen that $X(t)$ is simply a shifted version of $Y(t)$ and its dynamics is given by:

$$X(t+1) = X(t) - D(t) + R(t) \qquad (40)$$

We will show that this definition enables our algorithm to ensure that the constraint (4) is met.

We are now ready to state the dynamic control algorithm. Let $(W_1(t), W_2(t), S(t))$ be the system state in slot t. Define $\mathbf{Q}(t) \triangleq (U(t), Z(t), X(t))$ as the queue state that includes the workload queue as well as auxiliary queues. Then the dynamic algorithm chooses control decisions $P(t), R(t), D(t)$ and $\gamma(t)$ as the solution to the following problem:

P6 :

$$\text{Max:} [U(t) + Z(t)] P(t) - V \Big[P(t)C(t) + 1_R(t)C_{rc} + 1_D(t)C_{dc} \Big]$$
$$+ [X(t) + U(t) + Z(t)](D(t) - R(t))$$

Subject to: Constraints $(27), (29), (2), (5), (9)$

where $V > 0$ is a control parameter that affects the distance from optimality. Let $P^*(t), R^*(t), D^*(t)$ and $\gamma^*(t)$ denote the optimal solution to **P6**. Then, the dynamic algorithm allocates $(1 - \gamma^*(t))(P^*(t) - R^*(t) + D^*(t))$ power to service the delay intolerant workload and the remaining is used for the delay tolerant workload.

After computing these quantities, the algorithm implements them and updates the queueing variable $X(t)$ according to (40). This process is repeated every slot. Note that in solving **P6**, the control algorithm only makes use of the current system state values and does not require knowledge of the statistics of the workload or unit cost processes.

We now establish two properties of the structure of the optimal solution to **P6** that will be useful in analyzing its performance later.

LEMMA 5. *The optimal solution to **P6** has the following properties:*

1. *If $X(t) > -VC_{min}$, then the optimal solution always chooses $R^*(t) = 0$.*

2. *If $X(t) < -Q_{max}$ (where Q_{max} is specified in (44)), then the optimal solution always chooses $D^*(t) = 0$.*

PROOF. See [19]. ☐

6.3 Solving P6

Similar to **P3**, the complexity of solving **P6** depends on the structure of the unit cost function $C(t)$. For many cases of practical interest, **P6** is easy to solve and admits closed form solutions that can be implemented in real time. We consider one such case here.

6.3.1 $C(t)$ *does not depend on* $P(t)$

For notational convenience, let $Q_1(t) = [U(t) + Z(t) - VC(t)]$ and $Q_2(t) = [X(t) + U(t) + Z(t)]$.

Let $\theta_1(t)$ denote the optimal value of the objective in **P6** when there is no recharge or discharge. When $C(t)$ does not depend on $P(t)$, this can be calculated as follows: If $U(t) + Z(t) \geq VC(t)$, then $\theta_1(t) = Q_1(t)P_{peak}$. Else, $\theta_1(t) = Q_1(t)W_2(t)$.

Next, let $\theta_2(t)$ denote the optimal value of the objective in **P6** when the option of recharge is chosen, so that $R(t) > 0, D(t) = 0$. This can be calculated as follows:

1. If $Q_1(t) \geq 0, Q_2(t) \geq 0$, then $\theta_2(t) = Q_1(t)P_{peak} - VC_{rc}$.

2. If $Q_1(t) \geq 0, Q_2(t) < 0$, then $\theta_2(t) = Q_1(t)P_{peak} - Q_2(t)R_{max} - VC_{rc}$.

3. If $Q_1(t) < 0, Q_2(t) \geq 0$, then $\theta_2(t) = Q_1(t)W_2(t) - VC_{rc}$.

4. If $Q_1(t) < 0, Q_2(t) < 0$, then we have two cases:

   (a) If $Q_1(t) \geq Q_2(t)$, then $\theta_2(t) = Q_1(t)(R_{max} + W_2(t)) - Q_2(t)R_{max} - VC_{rc}$.

   (b) If $Q_1(t) < Q_2(t)$, then $\theta_2(t) = Q_1(t)W_2(t) - VC_{rc}$.

Finally, let $\theta_3(t)$ denote the optimal value of the objective in **P6** when when the option of discharge is chosen, so that $D(t) > 0, R(t) = 0$. This can be calculated as follows:

1. If $Q_1(t) \geq 0, Q_2(t) \geq 0$, then $\theta_3(t) = Q_1(t)P_{peak} + Q_2(t)D_{max} - VC_{dc}$.

2. If $Q_1(t) \geq 0, Q_2(t) < 0$, then $\theta_3(t) = Q_1(t)P_{peak} - VC_{dc}$.

3. If $Q_1(t) < 0, Q_2(t) \geq 0$, then $\theta_3(t) = Q_1(t)\max[0, W_2(t) - D_{max}] + Q_2(t)D_{max} - VC_{dc}$.

4. If $Q_1(t) < 0, Q_2(t) < 0$, then we have two cases:

   (a) If $Q_1(t) \leq Q_2(t)$, then $\theta_3(t) = Q_1(t)\max[0, W_2(t) - D_{max}] + Q_2(t)\min[W_2(t), D_{max}] - VC_{dc}$.

   (b) If $Q_1(t) > Q_2(t)$, then $\theta_3(t) = Q_1(t)W_2(t) - VC_{dc}$.

After computing $\theta_1(t), \theta_2(t), \theta_3(t)$, we pick the mode that yields the highest value of the objective and implement the corresponding solution.

6.4 Performance Theorem

We define an upper bound V_{ext}^{max} on the maximum value that V can take in our algorithm for the extended model.

$$V_{ext}^{max} \triangleq \frac{Y_{max} - Y_{min} - (R_{max} + D_{max} + W_{1,max} + \epsilon)}{\chi_{min} - C_{min}} \qquad (41)$$

Then we have the following result.

THEOREM 2. *(Algorithm Performance) Suppose $U(0) = 0$, $Z(0) = 0$ and the initial battery charge level Y_{init} satisfies $Y_{min} \leq Y_{init} \leq Y_{max}$. Then implementing the algorithm above with any fixed parameter $\epsilon \geq 0$ such that $\epsilon \leq W_{max} - W_{2,max}$ and a parameter V such that $0 < V \leq V_{ext}^{max}$ for all $t \in \{0, 1, 2, \ldots\}$ results in the following performance guarantees:*

1. *The queues $U(t)$ and $Z(t)$ are deterministically upper bounded by U_{max} and Z_{max} respectively for all t where:*

$$U_{max} \triangleq V\chi_{min} + W_{1,max} \qquad (42)$$
$$Z_{max} \triangleq V\chi_{min} + \epsilon \qquad (43)$$

Further, the sum $U(t) + Z(t)$ is also deterministically upper bounded by Q_{max} where

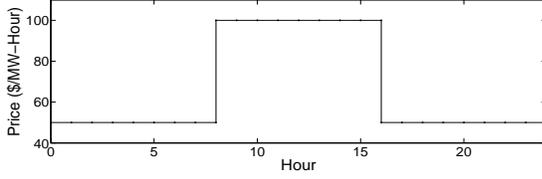$$Q_{max} \triangleq V\chi_{min} + W_{1,max} + \epsilon \qquad (44)$$

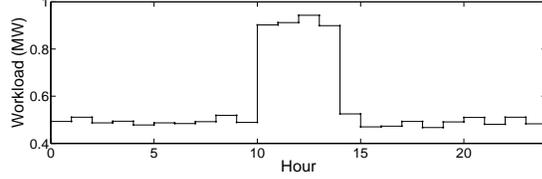**Figure 7: One period of the unit cost process.**

Figure 8: One period of the workload process.

2. The queue $X(t)$ is deterministically upper and lower bounded for all t as follows:

$$-Q_{max} - D_{max} \leq X(t) \leq Y_{max} - Y_{min} - Q_{max}$$
$$- D_{max} \quad (45)$$

3. The actual battery level $Y(t)$ satisfies $Y_{min} \leq Y(t) \leq Y_{max}$ for all t.

4. All control decisions are feasible.

5. The worst case delay experienced by any delay tolerant request is given by:

$$\left\lceil \frac{2V\chi_{min} + W_{1,max} + \epsilon}{\epsilon} \right\rceil \quad (46)$$

6. If $W_1(t), W_2(t)$ and $S(t)$ are i.i.d. over slots, then the time-average cost under the dynamic algorithm is within B_{ext}/V of the optimal value:

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E} \left\{ P(\tau)C(\tau) + 1_R(\tau)C_{rc} + 1_D(\tau)C_{dc} \right\}$$

$$\leq \hat{\phi}_{ext} + B_{ext}/V \quad (47)$$

where B_{ext} is a constant given by $B_{ext} = (P_{peak} + D_{max})^2 + \frac{(W_{1,max})^2 + \epsilon^2}{2} + B$ and $\hat{\phi}_{ext}$ is the optimal solution to **P4** under any feasible control algorithm (possibly with knowledge of future events).

Thus, by choosing larger V, the time-average cost under the dynamic algorithm can be pushed closer to the minimum possible value ϕ_{opt}. However, this increases the worst case delay bound yielding a $O(1/V, V)$ utility-delay tradeoff. Also note that V_{ext}^{max} limits how large V can be chosen.

PROOF. See [19]. □

7. SIMULATION-BASED EVALUATION

We evaluate the performance of our control algorithm using both synthetic and real pricing data. To gain insights into the behavior of the algorithm and to compare with the optimal offline solution, we first consider the basic model
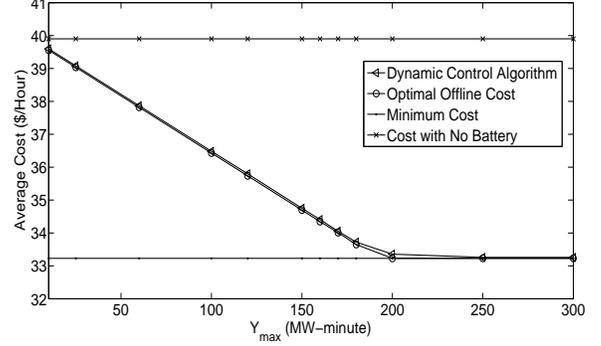

Figure 9: Average Cost per Hour vs. Y_{max}.

and use a simple periodic unit cost and workload process as shown in Figs. 7 and 8. These values repeat every 24 hours and the unit cost does not depend on $P(t)$. From Fig. 7, it can be seen that $C_{max} = \$100$ and $C_{min} = \$50$. Further, we have that $\chi_{min} = C_{max} = 100$. We assume a slot size of 1 minute so that the control decisions on $P(t), R(t), D(t)$ are taken once every minute. We fix the parameters $R_{max} = 0.2$ MW-slot, $D_{max} = 1.0$ MW-slot, $C_{rc} = C_{dc} = 0, Y_{min} = 0$. We now simulate the basic control algorithm of Sec. 5.1.1 for different values of Y_{max} and with $V = V_{max}$. For each Y_{max}, the simulation is performed for a duration of 4 weeks.

In Fig. 9, we plot the average cost per hour under the dynamic algorithm for different values of battery size Y_{max}. It can be seen that the average cost reduces as Y_{max} is increased and converges to a fixed value for large Y_{max}, as suggested by Theorem 1. For this simple example, we can compute the minimum possible average cost per hour over all battery sizes (this corresponds to ϕ of Sec. 4), and this is given by \$33.23 which is also the value to which the dynamic algorithm converges as Y_{max} is increased. Moreover, in this example, we can also compute the optimal offline cost for *each value* of Y_{max} (corresponding to ϕ_{opt}). These are also plotted in Fig. 9. It can be seen that, for each Y_{max}, the dynamic algorithm performs quite close to the corresponding optimal value, even for smaller values of Y_{max}. Note that Theorem 1 provides such guarantees only for sufficiently large values of Y_{max}. Finally, the average cost per hour when no battery is used is given by \$39.90.

We next consider a six-month data set of average hourly spot market prices for the Los Angeles Zone LA1 obtained from CAISO [1]. These prices correspond to the period $01/01/2005-06/30/2005$ and each value denotes the average price of 1 MW-Hour of electricity. A portion of this data corresponding to the first week of January is plotted in Fig. 1. We fix the slot size to 5 minutes. The unit cost $C(t)$ obtained from the data set for each hour is assumed to be fixed for that hour. Furthermore, we assume that the unit cost does not depend on the total power drawn $P(t)$.

In our experiments, we assume that the data center receives workload in an i.i.d fashion. Specifically, every slot, $W(t)$ takes values from the set [0.1,1.5] MW uniformly at random. We fix the parameters D_{max} and R_{max} to 0.5 MW-slot, $C_{dc} = C_{rc} = \$0.1$, and $Y_{min} = 0$. Also, $P_{peak} = W_{max} + R_{max} = 2.0$ MW. We now simulate four algorithms on this setup for different values of Y_{max}. The length of time the battery can power the data center if the draw were

Y_{max}	15	30	50
Battery, No WP	95%	92%	89%
WP, No Battery	96%	92%	88%
WP, Battery	92%	85%	79%

Table 2: Ratio of total cost under schemes (B), (C), (D) to the total cost under (A) for different values of Y_{max} with i.i.d. $W(t)$ over the 6 month period.

W_{max} starting from fully charged battery is given by $\frac{Y_{max}}{W_{max}}$ slots, each of length 5 minutes. We consider the following four schemes: (A) "No battery, No WP," which meets the demand in every slot using power from the grid and without postponing any workload, (B) "Battery, No WP," which employs the algorithm in the basic model without postponing any workload, (C) "No Battery, WP," which employs the extended model for WP but without any battery, and (D) "Complete," the complete algorithm of the extended model with both battery and WP. For (C) and (D), we assume that during every slot, half of the total workload is delay-tolerant.

We simulate these algorithms to obtain the total cost over the 6 month period for $Y_{max} \in \{15, 30, 50\}$ MW-slot. For (B), we use $V = V_{max}$ while for (C) and (D), we use $V = V_{ext}^{max}$ with $\epsilon = W_{max}/2$. Note that an increased battery capacity does not have any effect on the performance under (C). In order to get a fair comparison with the other schemes, we assume that the worst case delay guarantee that case (C) must provide for the delay tolerant traffic is the same as that under (D).

In Table 2, we show the ratio of the total cost under schemes (B), (C), (D) to the total cost under (A) for these values of Y_{max} over the 6 month period. The total cost over the 6 month period under (A) was found to be $143, 141.11. It can be seen that (D) combines the benefits of both (B) and (C) and provides the most cost savings over the baseline case. For example, with $Y_{max} = 50$ MW-slot, the total savings provided by (B), (C), and (D) are $15, 745, $17, 176 and $30, 000, respectively.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of opportunistically using energy storage devices to reduce the time average electricity bill of a data center. Using the technique of Lyapunov optimization, we designed an online control algorithm that achieves close to optimal cost as the battery size is increased.

We would like to extend our current framework along several important directions including: (i) multiple utilities (or captive sources such as DG) with different price variations and availability properties (e.g., certain renewable sources of energy are not available at all times), (ii) tariffs where the utility bill depends on peak power draw in addition to the energy consumption, and (iii) devising online algorithms that offer solutions whose proximity to the optimal has a smaller dependence on battery capacity than currently. We also plan to explore implementation and feasibility related concerns such as: (i) what are appropriate trade-offs between investments in additional battery capacity and cost reductions that this offers? (ii) what is the extent of cost reduction benefits for realistic data center workloads? and (iii) does stored energy make sense as a cost optimization knob in other domains besides data centers? Our technique could be viewed as a design tool which, when parameterized well, can

assist in determining suitable configuration parameters such as battery size, usage rules-of-thumb, time-scale at which decisions should be made, etc. Finally, we believe that our work opens up a whole set of interesting issues worth exploring in the area of consumer-end (not just data centers) demand response mechanisms for power cost optimization.

Acknowledgments

9. REFERENCES

[1] California ISO Open Access Same-time Information System (OASIS) Hourly Average Energy Prices. http://oasisis.caiso.com.
[2] Lead-acid batteries: Lifetime vs. Depth of discharge. http://www.windsun.com/Batteries/Battery_FAQ.htm.
[3] A. Bar-Noy, Y. Feng, M. P. Johnson, and O. Liu. When to reap and when to sow: Lowering peak usage with realistic batteries. In *Proc. 7th International Conference on Experimental Algorithms*, 2008.
[4] A. Bar-Noy, M. P. Johnson, and O. Liu. Peak shaving through resource buffering. In *Proc. WAOA*, 2008.
[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control, vols. 1 and 2*. Athena Scientific, 2007.
[6] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. *SIGOPS Oper. Syst. Rev.*, 35:103–116, Oct. 2001.
[7] M. Gatzianas, L. Georgiadis, and L. Tassiulas. Control of wireless networks with rechargeable batteries. *IEEE Trans. Wireless. Comm.*, 9:581–593, Feb. 2010.
[8] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Found. and Trends in Networking*, 1:1–144, 2006.
[9] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. Drpm: Dynamic speed control for power management in server class disks. In *Proc. ISCA '03*, 2003.
[10] U. Hoelzle and L. A. Barroso. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool, 2009.
[11] K. Le, R. Bianchini, M. Martonosi, and T. Nguyen. Cost- and energy-aware load distribution across data centers. In *Proc. HOTPOWER*, 2009.
[12] A. R. Lebeck, X. Fan, H. Zeng, and C. Ellis. Power aware page allocation. *SIGOPS Oper. Syst. Rev.*, 34:105–116, Nov. 2000.
[13] D. Linden and T. B. Reddy. *Handbook of Batteries*. McGraw Hill Handbooks, 2002.
[14] M. Marwah, P. Maciel, A. Shah, R. Sharma, T. Christian, V. Almeida, C. Araújo, E. Souza, G. Callou, B. Silva, S. Galdino, and J. Pires. Quantifying the sustainability impact of data center availability. *SIGMETRICS Perform. Eval. Rev.*, 37:64–68, March 2010.
[15] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
[16] M. J. Neely, A. S. Tehrani, and A. G. Dimakis. Efficient algorithms for renewable energy allocation to delay tolerant consumers. In *Proc. IEEE SmartGridComm*, 2010.
[17] S. Park, W. Jiang, Y. Zhou, and S. Adve. Managing energy-performance tradeoffs for multithreaded applications on multiprocessor architectures. In *Proc. ACM SIGMETRICS*, 2007.
[18] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *Proc. SIGCOMM*, 2009.
[19] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam. Optimal power cost management using stored energy in data centers. *arXiv Technical Report: arXiv:1103.3099v2*, March 2011.
[20] Q. Zhu, F. David, C. Devaraj, Z. Li, Y. Zhou, and P. Cao. Reducing energy consumption of disk storage using power-aware cache management. In *Proc. HPCA*, 2004.