# Datacenter Net Profit Optimization with Deadline Dependent Pricing

Wei Wang[1], Peng Zhang[1], Tian Lan[1], and Vaneet Aggarwal[2]
[1]Department of ECE, George Washington University, DC 20052, USA
[2]AT&T Labs - Research, Florham Park, NJ 07932, USA

***Abstract*—This paper considers deadline dependent pricing and its impact to datacenter net profit optimization. We formulate the problem by jointly maximizing total revenue as a function of individual job deadlines and minimizing electricity cost through job scheduling over different time and locations. These two complementary objectives—the maximization of revenue and the minimization of cost—are mutually-dependent due to the coupling of job completion time and scheduling decisions. Leveraging a new approximation method for job completion time, we develop two low-complexity, distributed algorithms for the net profit optimization. Through numerical evaluations, we show the efficacy of the proposed algorithms as well as the net profit improvements.**

## I. Introduction

Large investments have been made in recent years in data centers and cloud computing. A survey by KPMG International [1] in February 2012 shows that over 50% of senior executives find the most important impact of cloud computing to their business models to be its cost benefits. Accordingly, much research has been focused on demystifying various economic issues in data center and cloud computing.

Due to the fast growing power consumption of data centers, reducing the total electricity cost and therefore maximizing data center net profit is becoming ever more urgent and important. In contrast to the adoption of low power devices and energy saving solutions in [2], [3], a cost-optimization approach in [4], [5] exploits the diversities of electricity prices over time and geographic regions. In particular, a load-balancing algorithm is proposed in [5] to coordinate cloud computing workload with electricity prices at distributed data centers, in order to achieve the goal of minimizing the total electricity cost while guaranteeing the average service delay experienced by all jobs.

However, achieving an average service delay is unsatisfactory to cloud users, not only because they may have heterogeneous job requirements and spending budgets, but also due to a fundamental limitation of the average delay approach - individual job completion times are still randomly scattered over a wide range of values. Delayed response may frustrate users, and consequently, result in revenue loss. Therefore, the ability to deliver according to pre-defined Service Level Agreements (SLAs) increasingly becomes a competitive requirement [6], [7]. Cloud providers and users can negotiate SLAs (e.g., individual job deadlines) to determine costs and penalties based on the desired performance and budget. This presents an opportunity to offer deadline dependent pricing, which generates an additional source of revenue for cloud providers. It immediately raises the following question: How to jointly optimize both electricity cost of distributed data centers and total revenue from deadline dependent pricing, in order to maximize the net profit cloud providers receive?

In this paper, we systematically study the problem of data center net profit optimization, which aims to maximize the different between total revenue from deadline dependent pricing and electricity cost of distributed data centers. Time-dependent pricing, which charges users based on not only *how much* resources are consumed, but also *when* they are consumed, has been widely studied in the electricity industry [8], [9], [10], [11] and the Internet Service Provider (ISP) industry [12], [13], [14] as an effective solution to even out resource consumption peaks and reduce operating costs. However our problem is largely different because the prices are determined by job deadlines (i.e., completion times), whereas job duration or progress are irrelevant.

Our goal is to maximize the data center net profit through job scheduling. It requires (1) to maximize the total revenue as a function of individual job completion deadlines, (2) to minimize the electricity cost by scheduling jobs to different time and locations, and (3) to satisfy a capacity constraint at each distributed data center. We formulate this problem as a constrained optimization, whose solutions characterizes an interesting tradeoff between delay and cost - while completing a job earlier generates a higher revenue, it restricts the set of feasible scheduling decisions, causing higher electricity cost. While being a complex, mixed-integer optimization due to the scheduling formulation, the net profit optimization problem remains to be hard due to the coupling of scheduling decisions and job completion deadlines. There is no closed-form, differentiable function that computes job completion deadlines by scheduling decisions, which is often necessary if standard optimization techniques are to be applied.

Toward this end, we propose a new approximation method for job completion deadlines. Quantifying its gap in closed-form, we then leverage the approximation and the primal-dual decomposition [20] to design two efficient, distributed algorithmic solutions for the net profit optimization. Through numerical evaluations, we show the efficacy of the proposed algorithms as well as the total electricity cost reduction. Due to space limitations, all proofs are omitted in this paper and can be obtained in our online technical report in [21].

## II. System Model

In this section, we give the detailed modeling of the net profit optimization with deadline-dependent pricing over multiple regional data centers. We will first introduce the deadline dependent pricing mechanism and derive the total revenue, electricity cost, and workload constraints. Then we formulate the net profit optimization as a constrained integer maximization problem. The main notations are summarized in Table I as follows.

| Symbol | Meaning |
|---|---|
| $N$ | Number of distributed data centers, indexed by $j$ |
| $T$ | Number of periods, indexed by $t$ |
| $K$ | Number for jobs, indexed by $i$ |
| $U_i(t)$ | Payment for completing job $i$ at time $t$ |
| $\eta_i, r_i$ | Subscription time and demanded VM number of job $i$ |
| $m_j(t)$ | Number of active servers at data center $j$ |
| $M_j$ | Available servers at data center $j$ |
| $c_j$ | Electricity consumption per server over one period |
| $\mu_j$ | Number of VMs rate per server |
| $P_j(t)$ | Electricity price for data center $j$ at time $t$ |
| $x_{i,j}(t)$ | Number of VMs data center $j$ assigns to job $i$ at time $t$ |
| $d_i$ | Scheduled completion time of Job $i$ |
| $\mathcal{X}(d_i)$ | The set of all feasible job scheduling decisions |

TABLE I
MAIN NOTATIONS.

### A. Deadline Dependent Pricing

Deadline dependent pricing charges users based on not only *how much* resources are consumed, but also *when* they are consumed. Suppose that a cloud computing service consists of a set of $N$ distributed data centers, and that its billing cycle is divided into $T$ periods. Let $K$ be the total number of jobs submitted for the next cycle. To enable deadline-dependent SLA and pricing, each user request $i$ not only contains the number of demanded VM instances (i.e., $r_i$) and the total subscription time (i.e., $\eta_i$), but is also associated with a bid function $U_i(t)$, which measures the payment user $i$ is willing to make if his job is accepted and scheduled to complete by period $t$. For example, a bid function is flat when a user is indifferent to job deadlines, whereas it becomes strictly decreasing when a user is willing to pay more to get his job completed early.

All user requests for the next cycle are collectively handled by a job scheduler, which decides the location and time for each job to be processed in a judiciary manner. This scheme can be viewed as an generalization of existing on-spot VM instances in Amazon EC2, which allows users to bid for multiple periods and different job deadlines (or job completion times). The system model is illustrated in Fig. 1. Let $d_i$ denote the scheduled completion time of job $i$, the total revenue received by the cloud provider over the next cycle is given as:

$$U_{\text{total}} = \sum_{i=1}^{K} U_i(d_i) \quad (1)$$

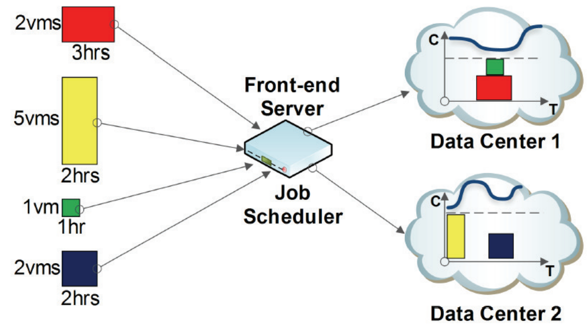


Fig. 1. An illustration of our system model. 4 jobs with different parameters are submitted to a front-end server, where a job scheduler decides the location and time for each job to be processed, by maximizing the net profit a data center operator receives.

### B. Electricity Cost

In the electricity market of North American, Regional Transmission Organization (RTO) is responsible for transmit electricity over large interstate areas and electricity prices are regionally different. Electricity prices remain the same for a relatively long period in some regions, while they may change every hour, even 15 Minutes in the regions who have wholesale electricity markets [5], [8], [9]. In this paper, we consider cloud computing services consisting of distributed regional data centers, which are subject to different electricity prices.

We make the assumption that the servers at each data center are homogeneous, so that the total electricity consumption at each data center $j$ over period $t$ can be calculated directly by multiplying the number of active servers $m_j(t)$ and the electricity consumption per server $c_j$. Let $P_j(t)$ be the electricity price of data center $j$ at time $t$. We also assume that $P_j(t)$ for $t = 1, \ldots, T$ is known non-causally at the beginning of each billing cycle. In practice, this can be achieved by stochastic modeling of electricity prices [15], [16] or by purchasing forward electricity contracts in the whole sale market [17], [22]. Thus, the total electricity cost for $N$ data centers over the next cycle is given as:

$$C_{\text{total}} = \sum_{j=1}^{N} \sum_{t=1}^{T} m_j(t) c_j P_j(t) \quad (2)$$

### C. Workload Constraints

We denote the number of VM instances received by job $i$ from data center $j$ at period $t$ as $x_{i,j}(t)$. We consider two types of jobs: divisible jobs and indivisible jobs. An indivisible job that requires $r_i$ VM instances and a subscription time $\eta_i$ cannot be interrupted and must have $r_i$ VM running continuously for $\eta_i$ periods, whereas a divisible job can be partitioned arbitrarily into any number of portions, as long as the total VM instance-hour is equal to the demand $r_i \eta_i$. Each portion of a divisible job can ran independently from other portions. Examples of applications that satisfy this divisibility property include image processing, database search, Monte Carlo simulations, computational fluid dynamics, and matrix computations [18].

Given the scheduling decisions $x_{i,j}(t)$ of all jobs, the aggregate workload for each regional data center must satisfy a service rate constraint:

$$\sum_{i=1}^{K} x_{i,j}(t) \le \mu_j m_j(t),\ \forall j,t \tag{3}$$

where $\mu_j$ is the service rate per server at regional data center $j$, and $m_j(t)$ is the number of active server at time $t$. There is also a limitation on the number of servers at each location. Therefore, we have

$$m_j(t) \le M_j,\ \forall t \tag{4}$$

### *D. Net Profit Maximization*

The goal of this paper is to design a job scheduler that maximizes the net profit $U_{\text{total}} - C_{\text{total}}$ over feasible scheduling decisions $x_{i,j}(t)$, subject to the workload constraints. For a given job completion time $d_i$, we denote the set of all feasible job scheduling decisions by a set $\mathcal{X}(d_i)$. In particular, for a divisible job, a user is only concerned with the total VM instance-hour received before time $d_i$. This implies

$$\mathcal{X}(d_i) = \left\{ x_{i,j}(t) : \sum_{j=1}^{N}\sum_{t=1}^{d_i} x_{i,j}(t) = r_i\eta_i \right\} \tag{5}$$

where $r_i\eta_i$ is the total demand of job $i$. On the other hand, an indivisible job must be assigned to a single regional data center and run continuously before completion. It will have a feasible scheduling set as

$$\mathcal{X}(d_i) = \left\{ x_{i,j}(t) : x_{i,j}(t) = 0 \text{ or } r_i \cdot \mathbf{1}_{(d_i-\eta_i \le t \le d_i)} \right\} \tag{6}$$

where $\mathbf{1}_{(d_i-\eta_i \le t \le d_i)}$ is an indicator function and equals to 1 if $t$ is belongs to the scheduled execution interval $[d_i - \eta_i, d_i]$, and 0 otherwise. We suppress the positivity constraints of $x_{i,j}(t)$ for simplicity of expressions.

We formulate the data center Net Profit Optimization (NPO) problem as follows:

**Problem NPO** :

$$\text{maximize} \quad \sum_{i=1}^{K} U_i(d_i) - \sum_{j=1}^{N}\sum_{t=1}^{T} m_j(t) c_j P_j(t) \tag{7}$$

$$\text{subject to} \quad \sum_{i=1}^{K} x_{i,j}(t) \le \mu_j m_j(t), \forall j, t \tag{8}$$

$$m_j(t) \le M_j,\ \forall j \tag{9}$$

$$\{x_{i,j}(t)\} \in \mathcal{X}(d_i), \forall i \tag{10}$$

$$\text{variables} \quad d_i, x_{i,j}(t), m_j(t) \tag{11}$$

The above problem can be looked at graphically as illustrated in Fig. 1. It requires a joint optimization of revenue and electricity cost, which is a mixed-integer optimization since the scheduling decisions $x_{i,j}(t)$ are discrete for indivisible jobs. Further, due to our deadline dependent pricing mechanism, the maximization of revenue over completion time $d_i$ is coupled with the minimization of electricity cost over feasible scheduling decisions $\mathcal{X}(d_i)$. However, there is no closed-form, differentiable function that represents $d_i$ by $\mathcal{X}(d_i)$. Therefore, off-the-shelf optimization algorithms cannot be directly applied.

It is worth noticing that our formulation of Problem NPO in (7) also incorporates sharp job deadlines. For instance, if job $i$ must be completed before time $t^*$, we can impose a utility function with $U_i(t) = -\infty$, for all $t > t^*$, so that scheduling decisions with a completion time later than $t^*$ become infeasible in Problem NPO.

## III. Approximation of Job Completion Time

Problem NPO is a complex joint optimization over both revenue and electricity cost, whose optimization variables, scheduling decisions $\{x_{i,j}(t)\} \in \mathcal{X}(d_i)$ and job completion time $d_i$, are not independent of each other. There is no closed-form, differentiable function that relates the two optimization variables. Let $x_i(t) = \sum_j x_{i,j}(t)$ be the aggregate service rate received by job $i$ at time $t$. For given scheduling decisions $x_{i,j}(t)$, we could have replaced constraint (10) by a supremum:

$$d_i = \sup_{\tau} \left\{ \tau : \sum_{t=1}^{\tau} x_i(t) < r_i\eta_i \right\}. \tag{12}$$

where $r_i\eta_i$ is the total demand of job $i$. However, it still requires evaluating supremum and inequalities, and therefore does not yield a differentiable function to represent $d_i$ by scheduling decisions $\{x_{i,j}(t)\}$.

In this section, we propose an approximation of the job completion time function in (12) by a differentiable function as follows:

$$\hat{d}_i = \frac{1}{\beta} \cdot \log\left( \frac{1}{r_i\eta_i} \sum_{t=1}^{T} e^{\beta t} \cdot x_i(t) \right), \tag{13}$$

where $\beta$ is a positive constant, and $1/(r_i\eta_i)$ normalizes the service rate $x_i(t)$. The accuracy of this approximation is given as follows.

*Theorem 1:* For a positive constant $\beta$ and a bounded service rate $0 \le x_i(t) \le x_{\max}$, we have

$$r_i\eta_i \cdot \left[ 1 - \frac{\log(1+\beta/x_{\max})}{\beta/x_{\max}} \right] \le \sum_{t=1}^{\hat{d}_i} x_i(t) \le r_i\eta_i. \tag{14}$$

Hence, $\lim_{\beta\to\infty} \hat{d}_i = d_i$, i.e., the approximation in (13) becomes exact.

*Proof:* It is easy to verify that $\hat{d}_i$ defined in (13) is the geometric mean of time $t$ with normalized weights $x_i(t)/(r_i\eta_i)$. Therefore, $\hat{d}_i$ is no more than the maximum $t$ that is associated with a positive weight. This implies $\hat{d}_i \le d_i$, which proves the last inequality.

To prove the first inequality, we consider $F = \sum_{t=\hat{d}_i+1}^{T} x_i(t)$, that is the cumulative mass of $x_i(t)$ after $\hat{d}_i$.

From (12), we have

$$
\begin{aligned}
e^{\beta \hat{d}_i} &= \frac{1}{r_i\eta_i}\sum_{t=1}^{T} e^{\beta t}\cdot x_i(t) \\
&\geq \frac{1}{r_i\eta_i}\sum_{t=\hat{d}_i+1}^{T} e^{\beta t}\cdot x_i(t) \\
&\geq \frac{1}{r_i\eta_i}\sum_{t=\hat{d}_i+1}^{T} e^{\beta t}\cdot f_{\max}\cdot 1_{(\hat{d}_i+1\leq t\leq \hat{d}_i+F/f_{\max})} \\
&= \frac{1}{r_i\eta_i}\sum_{t=1}^{F/f_{\max}} e^{\beta (t+\hat{d}_i)}\cdot f_{\max} \\
&= \frac{f_{\max}}{r_i\eta_i} e^{\beta \hat{d}_i}\left[e^{\beta(F/f_{\max}+1)}-e^{\beta}\right]/(e^{\beta}-1) \\
&\geq \frac{f_{\max}}{r_i\eta_i} e^{\beta \hat{d}_i}\left[e^{\beta F/f_{\max}}-1\right] \qquad (15)
\end{aligned}
$$

Here the second step uses positivity of $x_i(t)$, and the fourth step absorbs the indicator function into the summation. The third step bounds the weighted sum of $x_i(t)$ by that of another distribution $x_{\max}1_{(\hat{d}_i+1\leq t\leq \hat{d}_i+F/x_{\max})}$, which has the same cumulative mass $F$ over $[\hat{d}_i+1, T]$, and is construct by skewing distribution $x_i(t)$ to $\hat{d}_i$ as much as possible. It provides a lower bound because $e^{\beta t}$, for $\beta > 0$, is monotonically increasing over $t \geq \hat{d}_i$.

Finally, we take $\beta \to \infty$ in (14) and make use of $x_{\max}\log(1+\beta/x_{\max})/\beta \to 0$, which implies $\hat{d}_i \to d_i$ according to the definition in (12). This means that the approximation in (13) becomes exact. ■

**Remark:** By the approximation in (13) , we obtain a closed-form, differentiable expression for $\hat{d}_i$, which guarantees an approximated completion of job $i$, off by a logarithmic term $x_{\max}\log(1+\beta/x_{\max})/\beta$ in the worst case. The approximation becomes exact as $\beta$ approaches infinity. However, often there are practical constraints or overhead concerns on using large $\beta$. We choose an appropriate $\beta$ such that the resulting optimality gap is sufficiently small.

## IV. Algorithmic Solution for Divisible Jobs

In this section, we present a solution method for Problem NPO with divisible jobs, which have a feasible set:

$$
\mathcal{X}(d_i) = \left\{ x_{i,j}(t) : \sum_{j=1}^{N}\sum_{t=1}^{d_i} x_{i,j}(t) = r_i\eta_i \right\} \qquad (16)
$$

In order to solve the problem, we leverage the approximation of job completion time in Section III to obtain an approximated version of Problem NPO. The resulting problem has a easy-to-handle analytical structure, and is convex for certain choices of $U_i$ functions. Further, when the problem is non-convex, we then convert it into a sequence of linear programming and solve it using an efficient algorithm. It provides useful insights for solving Problem NPO with indivisible jobs.

Rewriting (10) in Problem NPO using the approximation in (13) and the feasible set in (15), we obtain a net profit optimization problem for divisible jobs (named Problem NPOD) as follows:

**Problem NPOD** :

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{K} U_i(\hat{d}_i) - \sum_{j=1}^{N}\sum_{t=1}^{T} m_j(t)c_jP_j(t) && (17)\\
\text{subject to} \quad & \sum_{i=1}^{K} x_{i,j}(t) \leq \mu_j m_j(t), \forall j,t && (18)\\
& m_j(t) \leq M_j,\ \forall j && (19)\\
& \hat{d}_i = \frac{1}{\beta}\log\left(\frac{1}{r_i\eta_i}\sum_{t=1}^{T}\sum_{j=1}^{N} e^{\beta t}x_{i,j}(t)\right) && (20)\\
& \sum_{j=1}^{N}\sum_{t=1}^{T} x_{i,j}(t) = r_i\eta_i && (21)\\
\text{variables} \quad & \hat{d}_i, x_{i,j}(t), m_j(t) && (22)
\end{aligned}
$$

where completion time becomes a differentiable function of $x_{i,j}(t)$. The optimization variables in Problem NPOD may still be integer variables, i.e., the number of active servers $m_j(t)$ and the number of demanded VM instances $x_{i,j}(t)$. We can leverage rounding techniques (e.g., [19]) to relax Problem NPOD so that its optimization variables become continuous.

Since our goal is to maximize the net profit, we need to balance deadline-dependent revenue and electricity cost in Problem NPOD. Since inequality constraints in (17, 18, 20) are linear, we investigate the convexity of the optimization objective in (16), which is a function of $x_{i,j}(t)$ and $m_j(t)$, by plugging in the equality constraint (19).

*Proposition 1:* For $\hat{d}_i$ in (19) and a positive, differentiable $U_i(\cdot)$, the objective function of Problem NPOD is convex if $U''(y) \geq \beta U'(y)$ for all $y > 0$, and concave if $U''(y) \leq \beta U'(y)$ for all $y > 0$.

The two conditions of $U_i(\cdot)$ in Proposition 1 captures a wide range of functions in practice. Let $a > 0$ and $b$ be arbitrary constants. Examples of $U_i(\cdot)$ that result in a concave objective function include certain exponential and logarithmic functions. Examples that result in a convex objective function include linear $U_i(y) = b - ax$ and logarithm $U_i(y) = a - b\log(y)$. We remark that when $U''(y) \leq \beta U'(y)$ for all $y > 0$, Problem NPOD is a concave maximization. It can be solved efficiently by off-the-shelf convex optimization algorithms, e.g., primal-dual algorithm and interior point algorithm [20].

Next, we develop an iterative algorithm to solve Problem NPOD when it maximizes a convex objective function. We show that the problem can be converted into a sequence of linear programming. We then leverage a primal-dual algorithm to obtain a distributed solution to each linear programming.

Let $x_{i,j}^{(k-1)}(t)$ be a set of given scheduling decisions, and $\hat{d}_i^{(k-1)}$ be the corresponding completion time approximation using (13). We linearize revenue $U_i(\hat{d}_i)$ by its first order Taylor expansion:

$$
U_i(\hat{d}_i^{(k-1)}) + \frac{U_i'(\hat{d}_i^{(k-1)})}{r_i\eta_i\beta e^{\beta \hat{d}_i^{(k-1)}}}\sum_{j=1}^{N}\sum_{t=1}^{T} e^{\beta t}\left[x_{i,j}(t) - x_{i,j}^{(k-1)}(t)\right]
$$

Plugging this approximation into the objective function of Problem NPOD in (16), it reduces to the following linear programing:

$$\begin{array}{ll} \text{maximize} & \sum_{i,j,t} \frac{U_i'(\hat{d_i}^{(k-1)})}{r_i\eta_i\beta e^{\beta \hat{d_i}^{(k-1)}}} e^{\beta t} x_{i,j}(t) - m_j(t) c_j P_j(t) \\ \text{subject to} & (17),\ (18),\ (20) \\ \text{variables} & x_{i,j}(t), m_j(t) \end{array}$$

where additive terms that only depend on $x_{i,j}^{(k-1)}(t)$ are dropped, since they have no effect in the optimization over $x_{i,j}(t), m_j(t)$.

Relying on this linearization, we propose the following iterative algorithm to solve Problem NPOD by solving a sequence of linear programming. It starts at some random initial point $x_{i,j}^{(0)}(t)$, solves the linear programming with $x_{i,j}^{(k-1)}(t)$, and therefore generates a sequence $\{x_{i,j}^{(k)}(t)\}_{k=0}^{\infty}$. This procedure is indeed a special case of the *difference of convex programming*, has been extensively used in solving many non-convex programs of similar forms in machine learning [23]. It is shown that the sequence $\{x_{i,j}^{(k)}(t)\}_{k=0}^{\infty}$ converges to a local minimum or a saddle of Problem NPOD, as in Theorem 2. We further apply a primal-dual algorithm in [20] to obtain a distributed solution to each linear programming. The algorithm is summarized in Fig. 2.

*Theorem 2:* (*Difference of convex programming*) The sequence $\{x_{i,j}^{(k)}(t)\}_{k=0}^{\infty}$ generated by our proposed interative algorithm satisfies the monotonic ascent property, i.e.,

$$U_{\text{total}}^{(k)} - C_{\text{total}}^{k} \geq U_{\text{total}}^{(k-1)} - C_{\text{total}}^{k-1},\ \forall k. \tag{23}$$

which insures the convergence to a local minimum or a saddle of Problem NPOD.

## V. Algorithmic Solution for Indivisible Jobs

The Problem NPO with indivisible jobs is a mixed-integer optimization because its feasible set is discrete, i.e.,

$$\mathcal{X}(d_i) = \left\{x_{i,j}(t) : x_{i,j}(t) = 0 \text{ or } r_i \cdot \mathbf{1}_{(d_i - \eta_i \leq t \leq d_i)}\right\}, \tag{24}$$

which implies that the job requires $r_i$ VM to run continuously for $\eta_i$ periods and cannot be interrupted. Although we can use the technique in Section III to approximate job completion time, Problem NPO with indivisible job and a feasible set in (23) could still be very challenging to solve. In practice, it is often acceptable to solve the problem sub-optimally , but in a distributed manner. In the following, we leverage the insights obtained from solving Problem NPOD and the primal-dual algorithm in [20] to derive a heuristic, distributed algorithm for Problem NPO with indivisible jobs.

We first notice that for indivisible jobs, completion time $d_i$ can also be obtained by approximating the scheduled starting time of job $i$, plus its subscription time $\eta_i$, i.e.,

$$\hat{d_i} = -\frac{1}{\beta} \cdot \log\left(\frac{1}{r_i\eta_i}\sum_{t=1}^{T}\sum_{j=1}^{n} e^{-\beta t} \cdot x_{i,j}(t)\right) + \eta_i, \tag{25}$$

where a negative exponent $-\beta$ places higher weights at smaller $t$. Using Theorem 1, we can show that $\lim_{\beta\to\infty} \hat{d_i} = d_i$, i.e., the approximation in (13) becomes exact. The major advantage of this new approximation $\hat{d_i}$ is that it generates a convex $U_i(\hat{d_i})$ even if $U''(y) \leq U'(y)$ for all $y > 0$, which would otherwise result in a concave function if the approximation in (13) were used.

*Proposition 2:* For $\hat{d_i}$ in (24) and a positive, differentiable $U_i(\cdot)$, the objective function $\hat{d_i}$ is concave if $U''(y) \geq U'(y)$ for $y > 0$, and convex if $U''(y) \leq U'(y)$ for all $y > 0$.

Rewriting (10) in Problem NPO using the approximation in (24) and the feasible set in (23), we obtain a net profit optimization problem for indivisible jobs (named Problem NPOI) as follows:

**Problem NPOI** :

$$\begin{array}{lll} \text{maximize} & \sum_{i=1}^{K} U_i(\hat{d_i}) - \sum_{j=1}^{N}\sum_{t=1}^{T} m_j(t)c_jP_j(t) & (26) \\ \text{subject to} & \sum_{i=1}^{K} x_{i,j}(t) \leq \mu_j m_j(t), \forall j, t & (27) \\ & m_j(t) \leq M_j,\ \forall j & (28) \\ & \hat{d_i} = \eta_i - \frac{1}{\beta}\log\left(\sum_{t=1}^{T}\sum_{j=1}^{N} \frac{e^{-\beta t}}{r_i\eta_i} x_{i,j}(t)\right) & (29) \\ & x_{i,j}(t) = 0 \text{ or } r_i \cdot \mathbf{1}_{(d_i-\eta_i\leq t\leq d_i)} & (30) \\ \text{variables} & \hat{d_i}, x_{i,j}(t), m_j(t) & (31) \end{array}$$

Using the same derivation in Section IV, we can convert Problem NPOI into a sequence of mixed-integer linear programming, by linearizing a convex $U_i(\hat{d_i})$ with its Taylor expansion. For a given initial point $x_{i,j}^{(k-1)}(t)$ , we have

$$\begin{array}{ll} \text{maximize} & \sum_{i,j,t} \frac{-U_i'(\hat{d_i}^{(k-1)})}{r_i\eta_i\beta e^{\beta \hat{d_i}^{(k-1)}}} e^{-\beta t} x_{i,j}(t) - m_j(t) c_j P_j(t) \\ \text{subject to} & (26),\ (27),\ (29) \\ \text{variables} & x_{i,j}(t), m_j(t) \end{array}$$

Therefore, Problem NPOI can be solved by an iterative algorithm, which generates a sequence, $\{x_{i,j}^{(k)}(t)\}_{k=0}^{\infty}$, by the solution of the above mixed-integer linear programming.

Toward this end, we leverage the primal-dual algorithm in [20] to derive a distributed, heuristic algorithm for the mixed-integer linear programming. It is easy to see that constraint (26) is satisfied with equality at optimum. We introduce a set of Lagrangian multipliers $\lambda_j(t)$ (i.e., data center congestion prices) for constraint (27) and decompose the mixed-integer linear programming into individual user problems. The detailed derivation is omitted here due to space limitation. The proposed algorithm is summarized in Fig. 2.

## VI. Simulations

In this section, we evaluate our algorithmic solution for Problem NPOD and NPOI over a 24-hour cycle, divided into $T = 144$ 10-minute periods. The goal is to obtain empirically-validated insights about the feasibility/efficiency of our solution using synthesized data from a recent study [5]. We construct $N = 3$ regional data centers, which reside

**Initialize** a feasible point $\{x_{i,j}^{(0)}(t)\}$, $k=0$, and stepsize $\delta > 0$.
**for** each $k$, iteratively solve the linearization:
  Find effective profit for each data center $j$ and period $t$:
  $\Gamma_j(t) = \frac{U_i'(\hat{d}_i^{(k-1)})}{r_i \eta_i \beta e^{\beta \hat{d}_i^{(k-1)}}} e^{\beta t} - \frac{c_j P_j(t) + \lambda_j^{(k)}(t)}{\mu_j}$
  **for** each job $i$
    Schedule job $i$ to maximize $\sum_{j,t} \Gamma_j(t) \cdot x_{i,j}^{(k)}(t)$, satisfying
      NPOD: $\sum_{i,j,t} x_{i,j}^{(k)}(t) = r_i \eta_i$
      NPOI: $x_{i,j}^{(k)}(t) = 0$ or $r_i \cdot \mathbf{1}_{(y - \eta_i \le t \le y)}$ for some $y$
  **end for**
  Obtain $m_j^{(k)}(t) = \sum_i x_{i,j}^{(k)}(t)$
  Update price $\lambda_j^{(k)}(t) = \left[\lambda_j^{(k-1)}(t) + \delta(m_j^{(k)}(t) - M_j)\right]^+$
  $k \leftarrow k+1$
**end for**

Fig. 2. Algorithm for Problem NPOD and NPOI.

in different electricity markets and have electricity prices $P_1(t)$ =\$56/$MWh$, $P_2(t)$ =\$$(51 + 15\cos 0.05t)/MWh$, and $P_3(t)$ =\$$(45 + 20\sin 0.06t)/MWh$. The data centers host $M_1 = 2000$, $M_2 = 1000$, $M_3 = 1000$ servers, respectively. Each server is operating at $c_j = 1200$ Watts with $\mu_j = 4$ VMs per server for $j = 1, 2, 3$.

We construct two types of jobs: elephant jobs that subscribes $r_i \in [50, 100]$ VMs for $\eta_i = [10 - 20]$ periods, and mice jobs that subscribes $r_i \in [5, 20]$ VMs for $\eta_i = [1 - 10]$ periods. In our simulations, both $r_i$ and $\eta_i$ are uniformly-randomly generated from their ranges. We fix the total workload to be $K = 1200$ jobs, each being an elephant job with probability $20\%$ and a mice job with probability $80\%$. Job $i$ is associated with a non-linear bid function, given by

$$U_i(d_i) = r_i \eta_i \cdot (a - b \log t) \text{ (dollars)} \quad (32)$$

where $a \in [0.01, 0.02]$ and $b \in [0.001, 0.002]$ are uniformly distributed. Using this non-linear bid function, the approximation of job completion time will result in a convex objective function in (16) and (25). All numerical results shown in this section are averaged over 5 realizations of random parameters.

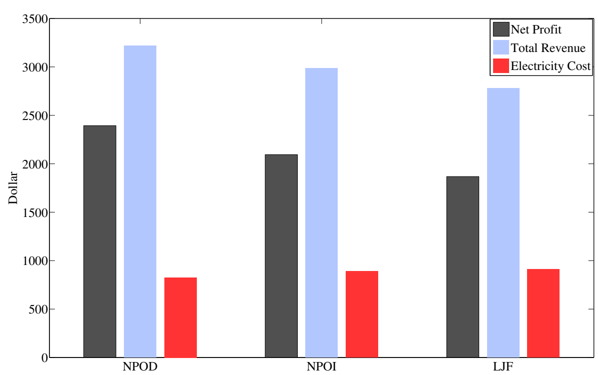


Fig. 3. A comparison of optimized net profit of NPOD, NPOI, LJF.

We solve Problem NPOD and NPOI using the proposed algorithms in Fig. 2. Problem NPOD and NPOI use the same job parameters, while Problem NPOD assume that all jobs are divisible, and therefore provides an upper bound for Problem NPOI. To provide benchmarks for our evaluations, we also implement a greedy algorithm that sequentially schedules all jobs with a *Largest Job First (LJF)* policy. Fig. 3 compares the optimized net profit of NPOD, NPOI, and LJF algorithms. When jobs are indivisible, our NPOI algorithm improves the net profit by $12\%$ over the baseline LJF algorithm (from \$1,868 to \$2,095), while an additional $16\%$ increment (to \$2394) can be achieved by NPOD if all jobs are divisible. We also notice that our NPOI algorithm is able to simultaneously improve total revenue and cut down electricity cost, compared to the baseline LJF algorithm. This is achieved by the joint optimization over job completion time $d_i$ and scheduling decisions $x_{i,j}(t)$.

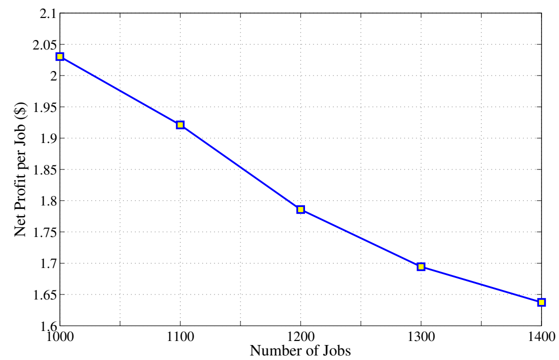


Fig. 4. Plot net profit per job for different workload in NPOI.

Fig. 4 studies the impact of changing data center workload when jobs are indivisible. We run the NPOI algorithm and plot the average profit per job by varying the number of jobs from $K = 1000$ to $K = 1400$. Increasing the workload results in smaller net profit per job, not only because it makes the average electricity price to go up, but also due to the result of higher congestion in data centers, causing job completion times to be pushed behind. This is not to discourage data center operators to run their services at low utilization. It is also worth noticing that while net profit per job goes down, the total net profit monotonically increases from \$2,030 for $K = 1000$ jobs to \$2,292 for $K = 1400$ jobs.

## VII. Conclusion

This paper studies data center net profit optimization with deadline dependent pricing, by a joint maximization of revenue and minimization of electricity costs. Making use of a new approximation for job completion time, we develop two distributed algorithms for the net profit optimization. The efficacy of the proposed algorithms as well as the total electricity cost reduction are demonstrated through numerical evaluations. As a next step work, we plan to extend the proposed method to capture dynamic job arrival/departures.

## REFERENCES


[1] KPMG International, "Clarity in the Cloud", *Online technical report at* $http://www.kpmg.com/$, Nov. 2011.

[2] X. Fan, W. Weber, L. A. Barroso, "Power provisioning for a warehousesized computer, *in Proceedings of the 34th annual international symposium on Computer architecture*, 2007.

[3] S. Nedevschi, L. Popal, G. Iannaccone, S. Ratnasamy, D. Wetherall, "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation, *In Proceedings of the 5th USENIX Symposium on Networked Systems Design & Implementations (NSDI)*, 2008.

[4] L. Parolini, B. Sinopoli, and B. Krogh, "Reducing Data Center Energy Consumption via Coordinated Cooling and Load Management," *In Proceedings of ACM Workshop on Power Aware Computing and Systems*, 2008.

[5] L. Rao, X. Liu, L. Xie, W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity Market Environment", *In Procedings of IEEE Infocom*, 2010.

[6] P. Patel, A. Ranabahu, and A. Sheth, "Service Level Agreement in Cloud Computing, *in Proceedings of the Workshop on Best Practices in Cloud Computing,* 2009.

[7] C. Wilson, H. Ballani, T. Karagiannis, A. Rowstron, "Better Never than Late, Meeting Deadlines in Datacenter Networks", *in Proceedings of Sigcomm*, 2011.

[8] S. Littlechild,"Wholesale Spot Price Pass-through," *Journal of Regulatory Economics*, vol. 23, no. 1, pp. 61-91, 2003.

[9] S. Borenstein, "The Long-Run Efficiency of Real-Time Electricity Pricing," *The Energy Journal*, vol. 26, no. 3, pp. 93-116, 2005.

[10] K. Herter, "Residential Implementation of Critical-peak Pricing of Electricity," *Energy Policy*, vol. 35, no. 4, pp. 2121-2130, 2007.

[11] A. Faruqui, R. Hledik, and S. Sergici, "Piloting the Smart Grid," *The Electricity Journal*, vol. 22, no. 7, pp. 55-69, 2009.

[12] H. Chao, "Peak Load Pricing and Capacity Planning with Demand and Supply Uncertainty," The Bell Journal of Economics, vol. 14, no. 1, pp. 179-190, 1983.

[13] G. Brunekreeft, "Price Capping and Peak-Load-Pricing in Network Industries," *Diskussionsbeitrage des Instituts fur Verkehrswissenschaft und Regionalpolitik*, Universitat Freiburg, vol. 73, 2000.

[14] Carlee Joe-Wong, Sangtae Ha, Soumya Sen, and Mung Chiang, "Pricing by Timing: Innovating Broadband Data Plans", *In Procedings of ICDCS*, 2011.

[15] P. Skantze, M.D. Ilic and J. Chapman, "Stochastic modeling of electric power prices in a mult-market environment, *IEEE Power Engineering Society Winter Meeting*, 2000.

[16] S. Fleten and J. Lemming, "Constructing Forward Price Curves in Electricity Markets," *Energy Economics*, vol. 25, pp. 409-424, 2007.

[17] Woo CK, Horowitz I, Hoang K. "Cross hedging and value at risk: wholesale electricity forward contracts." *Advances in Investment Analysis and Portfolio Management*, vol. 8, pp. 283-301, 2001.

[18] B. Veeravalli and W. H. Min, "Scheduling divisible loads on heterogeneous linear daisy chain networks with arbitrary processor release times", *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, pp. 273-288, 2010.

[19] A. Neumaier and O. Shcherbina, "Safe bounds in linear and mixed-integer linear programming", *Mathematical Programming*, vol. 99, pp. 283-296, 2004.

[20] S. Boyd and L. Vandenberghe, "Convex Optimization", *Cambridge University Press*, 2004.

[21] W. Wang, P. Zhang, T. Lan, V. Aggarwal, "Data Center Net Profit Optimization with Deadline Dependent Pricing", *Online technical report available at* $www.seas.gwu.edu/\sim tlan/papers/CISS\_2012\_Data\_Center.pdf$, 2012.

[22] R. Weber, "Cutting the electric bill for Internet-scale systems", pp. 123-134, *in Proceedings of Sigcomm* 2009.

[23] B. K. Sriperumbudur and G. R. G. Lanckriet, "On the Convergence of the Concave-Convex Procedure", *Online technical report at* $http://cosmal.ucsd.edu/\sim gert/papers/nips\_cccp.pdf$, 2009.