Minimizing the Electricity Bill of Cooperative Users under a Quasi-Dynamic Pricing Model

Safar Hatami and Massoud Pedram
University of Southern California
Department of Electrical Engineering - Systems
{*shatami,pedram*}@*usc.edu*

Abstract—**Dynamic energy pricing is a promising development that addresses the concern of finding an environmentally friendly solution to meeting energy needs of customers while minimizing their electrical energy bill. In this paper, we mathematically formulate the electrical energy bill minimization problem for cooperative networked consumers who have a single energy bill, such as those working in a commercial/industrial building. The idea is to schedule user requests for appliance use at different times during a fixed interval based on dynamic energy prices during that interval. Two different methods are presented to minimize the energy cost of such users under non-interruptible or interruptible jobs. The methods relay on a quasi-dynamic pricing function for unit of energy consumed, which comprises of a base price and a penalty term. The methods minimize the energy cost of the users while meeting all the scheduling constraints and heeding the pricing function. The proposed methods result in significant savings in the energy bill under different usage pricing, and scheduling constraints.**

Keywords: *energy, cost, optimization, dynamic pricing, non-linear programming, dynamic programming.*

I. INTRODUCTION

With the current emphasis on environmentally friendly solutions alongside the concerns of a recovering economy, dynamic energy pricing may be exploited as an effective means of utilizing green energies while reducing the electricity costs by a significant amount (i.e., by an average of 20%) [1]. Consumers need access to dynamic electricity pricing to reduce greenhouse gas (GHG) emissions and save money on their bills [2]. The Association of Home Appliance Manufacturers (AHAM) released a white paper strongly advocating that "residential electricity prices must be based on time of use" to fully enable smart grid technology [2].

Energy pricing may be classified as two major types: real-time/dynamic pricing and time-of-use (TOU) pricing. An economic view of real-time and TOU energy pricing has been presented in [3] where it is shown that dynamic pricing is the ideal method to capture the true cost of producing energy [3]. Also, dynamic changes in energy prices provide an incentive for the customer to reduce their energy consumption during "peak" energy-use hours [3]. Since dynamic energy pricing results in a time shift of consumption from peak time to off-peak time, the grid power capacity requirement reduces, which can result in around 10% gain for the whole energy economy [3].

By transitioning to dynamic energy pricing and by providing relevant information to the consumers (e.g., energy consumption comparison with similar households/facilities or the mix of green vs. brown energy sources that are being used by the utility companies to produce electricity at a given time), there will be strong incentives to reduce the overall energy use to reduce cost or to change the energy usage profile to make it more environmentally friendly.

It is clear that a time-based pricing is useful when there is a significant difference between usage of peak and off-peak times. This is often the case as indicated, for example, by studies published by the Demand Response Research Center on Automated Critical Peak Pricing [4], which emphasizes the difference in peak, off peak, and "needle peak" energy demands. The time-based pricing model that we use in this paper is a *quasi-dynamic pricing model* as detailed next. The price of one unit of energy consumption comprises of two parts: (i) A *TOU-dependent base price*, which is specified in advance, and captures the slow dynamics of energy usage; an example is the hourly price of a unit of energy consumption in the current day provided the day before, and (ii) A *penalty term*, which penalizes the users when their peak power consumption over some recent window of time goes above a predetermined TOU-dependent threshold.

A smart grid, which delivers electricity from suppliers to consumers, controls appliances at consumers' homes, and provides the required data and applications to optimize energy costs, is a very large and complex system with a lot of sensing/metering, data fusion and information processing as well as intelligent control and decision making built into it. It comprises of at least a physical power generation facility, power distribution, and power delivery infrastructure.

To achieve end user energy savings, it is necessary to install power meters, sensors/actuators for appliances and other power consuming resources in a facility (e.g., air conditioning, heating and ventilation, lighting, and machine rooms) to track and control energy consumption. This by itself requires significant upfront investment, planning, and installation work. Hence, only a select number of areas in the country have implemented consumer-level energy optimization based on dynamic energy pricing. However, the potential payoff is so large that we expect this monitoring/sensing/actuation infrastructure will be put in place rather rapidly. For example, Power Smart Pricing, a program from the Ameren Illinois Utilities, provides the customer with the billing price for electricity as it varies – hourly – based on the actual market price [1]. Participants in the program "saved an average of 20 percent compared with what they would have paid on the standard fixed rate pricing scheme (based on billing results for December 2007 through December 2009.)"

In this paper we propose two different methods, a deadline-driven continuous-variable method and a timeslot-based method to optimize the energy cost of a networked community of cooperative users.

The deadline-driven continuous-variable method is considered for two classes of energy cost optimizations: (a) atomic task execution with a bi-level penalty term in the pricing function and (b) interruptible task with a multi-level penalty term in the pricing function. In this method, continuous-variable, nonlinear programming is utilized for energy cost optimization. The method is appropriate for energy cost optimization for tasks with zero or very few interrupts. In addition the length of each fragment of an interruptible task is pre-specified.

The timeslot-based method itself uses two mathematical frameworks, one based on nonlinear programming, the other based on dynamic programming, for minimizing the overall energy cost to a community of cooperative networked users. The time slot-based method is able to handle tasks with multiple interrupts (suspend and resume operation later); the method automatically specifies the length of each fragment of an interrupted task (duration of the suspend mode). For the time slot-based method, the nonlinear programming approach may get trapped into a local minimum because of the non-convexity of the problem. However, when the problem is solved for different initial values, the global minimum may be found. The dynamic programming approach is much more efficient in obtaining the optimum solution however its memory requirement is much higher than the nonlinear programming.

Intuitively by scheduling energy consuming tasks at different time intervals over a specified time frame, our proposed methods are capable of minimizing the cost of energy consumed by a collection of cooperative users. An example scenario for such users would be office workers in an office building owned by a single owner who pays the full cost of electrical energy consumed by the office workers in the building.

The remainder of this paper is organized as follows. In the next section, we present two methods for optimizing the cooperative users' energy cost. Section 3 reports the simulation results. The paper is concluded in Section 4.

II. Cost Optimization

Two different methods are presented for optimizing the energy cost of a community of cooperative users in a (shared) facility. As also stated above, we assume that there is a single owner who pays the full energy cost of a community of cooperative users in a shared work space, e.g. a research laboratory, a warehouse, or an office space.

A. Problem Definition

This section defines the problem for energy cost optimizations. In the case of using a TOU schema in energy pricing, the customer is charged a specific rate during "peak hours" (when energy usage and cost is relatively high) and a lower price during "off-peak" hours (when the demand for energy decreases and energy costs are low). With this schema, the customer and our application have an accurate calculation of the price of energy over the day as it changes per unit of time (in the case analyzed in this paper, per hour), allowing for an optimal schedule to take advantage of low cost hours for high energy appliances.

To provide optimal scheduling for the customer and give them an incentive for them to minimize their energy cost and use during peak hours, the following constraints should be considered:

- Scheduling restrictions such as scheduling all requested jobs in the given time period (day, hour, etc.), and
- Power/ current usage should be less than the maximum allowed power /current usage threshold (no overload).

The importance of scheduling the tasks and providing accurate calculations of the overall savings for the customer (over short or prolonged periods of time) should be emphasized for the industrial customer who needs to schedule high energy-consuming tasks throughout the day while considering the cost of energy for the current hour, as well as the cost of creating a "needle peak" and accumulating a penalty for energy over-use for the given hour.

Notice that although the TOU-dependent base pricing does not provide an accurate reflection of the exact cost of providing energy and the rapid (real-time) fluctuation in price of producing energy [3], it allows the customer to be aware and responsive to the cost of energy use during peak hours. It is certainly a useful first step in the direction of full accounting of dynamic energy prices.

Definition 1: *Cooperative users* $U = \{U_n\}$ for $n = 1,\ldots,N$ want to carry out their tasks while minimizing the total cost of energy consumption. Each user, U_n, schedules the set of tasks $\{T_{nj}\}$ for $j=1,\ldots,K_n$, where K_n is the total number of tasks requested by user U_n. The set of all tasks U is defined as T, $T = \cup \{T_{nj}\} = \{T_k\}$ for $k = 1,\ldots,K$ where $K = \sum_{1}^{N} K_n$. Each task, T_k, is associated with an electrical appliance $A(T_k)$ that has m power modes. The power usage, $P_{k,m}$, of appliance $A(T_k)$

reflects its power mode (m). An appliance can only perform one task at any given time. The required time for performing task T_k in power mode m of appliance $A(T_k)$ is $I_{k,m}$. The power mode m of appliance A refers to different stages of the operation that an appliance goes thru when performing a function; For example, an appliance may begin with a start-job mode, normal operation mode, sleep mode, normal operation mode, and finish-job mode. The power mode m of appliances, their duration, and sequence are pre-specified and known. Note that here mode of an appliance does *not* refer to the existence of multiple power-performance modes (e.g., power saving, normal, and high performance) that a user may choose from.

For simplicity, the solutions given below are presented for applications with only two power-mode appliances. Obviously the methods can be easily extended to handle applications with any number of power-modes.

Definition 2: *Price function* $C(t)$ gives the price of one unit of energy (kWh) at time t. Note that price is given in discrete fragments of time; for example, price is 0.20 US dollars per kWh in time interval 0:00-1:00 PM. In order to give a general formulization which is simple in representation, we assume $C(t)$ is a function of continuous variable time t, with some discontinuities; For example $C(t)$, is discontinuous at time 1:00 PM, if it changes from \$0.20 to \$0.30 per kWh at that time. Continuous representation of price (integration of cost over t) will be used for the deadline-driven continuous-variable method while discrete representation of price (summation of cost over t) will be utilized for the timeslot-based method.

Definition 3: *Penalty function*, $R(p(t))$, is a multiplicative factor of the energy usage which penalizes the total energy cost of U based on its peak power usage, $p(t)$, over the last Δt time-units. The penalty function can dynamically penalize users when the users' consumption goes above some predetermined thresholds. Figure 1 shows a bi-level penalty function which penalizes the cost by a factor of $1+\alpha$ when the peak power usage in window Δt is more than p_0.

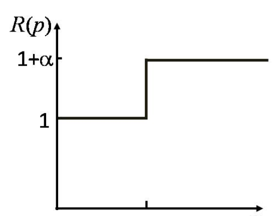

Figure 1: A bi-level penalty function.

The value of $p(t)$ is a function of time t and is calculated as follows:

$$p(t) = \max_{t-\Delta t \le x \le t} \sum_{k=1}^{K} d_k(x) P_{k,m} \quad (1)$$

where

$$d_k(x) = \begin{cases} 1 & A(T_k) \text{ is turned on} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Note that by changing the value of Δt, the presented methods move from a TOU-based pricing scheme (in which the energy cost is predetermined over the next day, week, etc.) toward a dynamic pricing scheme, in which the cost of energy changes according to the amount of power used at the given time. For example, when Δt=0, the penalty function depends on the peak power usage at time t.

Definition 4: *Energy Cost,* $G(a, b, k, m)$, is the cost of energy associated with appliance $A(T_k)$ which works from $t = a$ to $t = b$ at mode m and is calculated as follows:

$$\text{Energy Cost} = G(a,b,k,m) = \int_a^b P_{k,m} C(t) R(p(t))\,dt \quad (3)$$

Problem 1: *Cooperative Energy Cost Optimization is the problem of scheduling all tasks T of all users U so as to minimize the total energy cost while meeting all the scheduling constraints:*

$$Min\{Total\ Cost\} = Min\left\{ \sum_{k=1}^{K} \int_{I_{k,m}} P_{k,m} C(t) R(p(t))\,dt \right\} \quad (4)$$

B. Deadline-Driven Continuous-Variable Energy Cost Minimization

Consider *Problem 1* for a collection of cooperative-users, *U.* In this formulation, we suppose each appliance has only two power modes, 0 (start up mode) and 1 (normal mode). Each appliance $A(T_k)$ works at mode 0 for the first $I_{k,0}$ time units (start up time) and then goes to normal mode, 1. The proposed approaches below solve *Problem 1* by finding the start time for scheduling each task T_k for k= 1,…,K.

Atomic Task Execution and Bi-level Penalty Function

Consider the two-mode appliance version of *Problem 1* and suppose each task T_k is completed —without interruption— i.e., once it begins its operation on an appliance, it runs to completion.

$$Min\{Total\ Cost\} = Min\left\{ \sum_{k=1}^{K} \left(\begin{aligned} &\int_{a_k}^{a_k+I_{k,0}} P_{k,0} C(t) R(p(t))\,dt \\ &+\int_{a_k+I_{k,0}}^{a_k+I_{k,0}+I_{k,1}} P_{k,1} C(t) R(p(t))\,dt \end{aligned} \right) \right\} \quad (5)$$

A penalty function $R(p(t))$ which is a bi-level function of $p(t)$ can be defined as follows:

$$R(p(t)) = \begin{cases} 1+\alpha & p(t) \ge p_0 \\ 1 & p(t) < p_0 \end{cases} \quad (6)$$

Suppose in equation (1) that Δt is equal to zero, which means the penalty function depends on the peak power usage of U at time t. The function $p(t)$ can then be formulated as follows:

$$p(t) = p(t, \vec{a}, \overrightarrow{I^0}, \overrightarrow{I^1}) = \sum_{k=1}^{K} \begin{pmatrix} P_{k,0} f\left(t, a_k,\ a_k + I_{k,0}\right) + \\ P_{k,1} f\left(t, a_k + I_{k,0},\ a_k + I_{k,0} + I_{k,1}\right) \end{pmatrix} \quad (7)$$

where $\vec{a} = (a_1, ..., a_K)$, $\overrightarrow{I^0} = (I_{1,0}, ..., I_{K,0})$, $\overrightarrow{I^1} = (I_{1,1}, ..., I_{K,1})$. $f(t,a,b)$ is a pulse function and can be written as follows:

$$f(t,a,b) = u(t-a) - u(t-b) \quad (8)$$

where $u(t\text{-}\zeta)$ is a step function and can be given by:

$$u(t-\zeta) = \left\lfloor \left| \frac{\left\lfloor \frac{t}{\zeta} \right\rfloor + 2}{\left\lfloor \frac{t}{\zeta} \right\rfloor + 1} \right| - 2 \right| \quad (9)$$

Here $\lfloor\ \rfloor$ denotes the floor operation and a is greater than zero.

The optimization problem given in (5) is subject to the following constraints.

1- Scheduling start-times and deadlines of tasks T, enforce that each task T_k cannot start earlier than starting-time s_k and cannot go beyond deadline e_k:

$$\begin{cases} s_k \le a_k \\ a_k + I_{k,0} + I_{k,1} \le e_k \end{cases} \quad (10)$$

2- The power usage of U at time t cannot go beyond its power capacity ($P_{\max}$):

$$p(t) < P_{\max} \quad (11)$$

The nonlinear optimization problem given in (5) can be solved by using Sequential Quadratic Programming (SQP) [6] or a heuristic method such as simulated annealing. We utilized SQP to solve (5).

Interruptible task and Multi-level Penalty Function

In this section, we generalize the problem given in (5). The first extension is that each task T_k can be divided across L_k time-intervals. Although the required time for performing task T_k at power mode 1 of appliance $A(T_k)$ is $I_{k,1}$, appliance $A(T_k)$ can complete this task in L_k distinct, non-overlapping time intervals (task fragments). Whenever an appliance resumes working on a task, it must spend $I_{k,2} \le I_{k,0}$ time in restart mode. This revised problem may be stated as follows.

$$Min\{Total\,Cost\} = Min\left\{ \sum_{k=1}^{K} \sum_{l=1}^{L_k} \begin{pmatrix} \int_{a_{k,l}}^{a_{k,l}+I_{k,2}} P_{k,0} C(t) R(p(t))\,dt \\ + \int_{a_{k,l}+I_{k,2}}^{a_{k,l}+I_{k,2}+J_{k,l}} P_{k,1} C(t) R(p(t))\,dt \end{pmatrix} \right\} \quad (12)$$

where $a_{k,l}$ is the start time for the l^{th} fragment of the task and $J_{k,l}$ is the length of the task fragment (determined by user U_k). The following condition must hold true:

$$\sum_{l=1}^{L_k} J_{k,l} = I_{k,1} \quad (13)$$

As another extension, it is assumed

$$R(p(t)) = 1 + \alpha_h \qquad p_{h-1} \le p(t) < p_h \quad (14)$$

for h=1,…, H. where $p_0 = 0$ and $p_H = \infty$. The new form of $p(t)$ is:

$$p(t) = p(t, \vec{a}, \overrightarrow{I^0}, \overrightarrow{I^1}) = \sum_{k=1}^{K} \sum_{l=1}^{L_k} \begin{pmatrix} P_{k,0} f\left(t,\ a_{k,j},\ a_{k,l} + I_{k,2}\right) \\ + P_{k,1} f\left(t,\ a_{k,l} + I_{k,2},\ a_{k,l} + I_{k,2} + J_{k,l}\right) \end{pmatrix} \quad (15)$$

This revised optimization problem given in (12) is subject to the following constraints.

1- Scheduling start-times and deadlines of tasks T enforce that the first fragment for each task T_k cannot start earlier than the task's starting-time s_k, and the last task fragment cannot go beyond the deadline e_k:

$$\begin{cases} s_k \le a_{k,1} \\ a_{k,L_k} + I_{k,2} + J_{k,L_k} \le e_k \end{cases} \quad (16)$$

2- For each task fragment:

$$a_{k,l} + J_{k,l} + I_{k,2} < a_{k,l+1} \quad (17)$$

3- The power usage of U at time t cannot go beyond its power capacity ($P_{\max}$):

$$p(t) < P_{\max} \quad (18)$$

The optimization problem given by (12) is a nonlinear problem similar to (5), which can be solved by SQP or a heuristic method.

C. Time Slot-Based Energy Cost Minimization

Reconsider *Problem 1* for the cooperative-users U. Each appliance has a single power mode, say mode 1. The proposed slot-based methodology solves *Problem 1* by finding a time slots in which an appliance $A(T_k)$ is performing a task, for k= 1,…,K.

Definition 5: *Time Resolution*, τ, is the minimum time resolution (i.e., the greatest common divisor) used to determine the timing function and constraints in a *Cooperative Energy Cost Optimization* problem. For example, if the discrete price function is changing each hour, the task durations are a multiple of half of an hour, and the task deadlines are a multiple of 15-minute periods, then the time resolution will be a quarter of an hour (τ= 15 minutes). Hence, the energy cost can be measured in each *time slot* (with the length of τ units of time). For the new problem formulation, t denotes the index of time slot, and s_k, e_k and functions $C(t)$,

$R(p(t))$ and $p(t)$ are determined according to the slot index, t. The slot based optimization problem is:

$$Min\{Total\,Cost\} = Min\left\{\sum_{t=1}^{O}\sum_{k=1}^{K} d_k(t)P_{k,1}C(t)R(p(t))\tau\right\} \quad (19)$$

where O is the total number of time slots in the scheduling interval and

$$d_k(t) = \begin{cases} 1 & A(T_k)\text{ is turned on} \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

$R(p(t))$ and $p(t)$ are in turn calculated as follows:

$$R(p(t)) = 1 + \alpha_h \qquad p_{h-1} \le p(t) < p_h \quad (21)$$

$$p(t) = \sum_{k=1}^{M} d_k(t)P_{k,1} \quad (22)$$

The constraints of optimization problem (19) are the following.

1-Scheduling start-times and deadlines of tasks T, enforce that each task T_k cannot start earlier than starting-time s_k and cannot go beyond deadline e_k:

$$s_k < t < e_k \text{ where } d_k(t) = 1 \quad (23)$$

2- Each task T_k needs $I_{k,1}$ time unit to be done. Therefore,

$$I_{k,1} = \sum_{t=1}^{O} d_k(t)\tau$$

3- The power usage of U at time t cannot go beyond its power capacity ($P_{\max}$):

$$p(t) < P_{\max} \quad (24)$$

The optimization problem given in (19) is another nonlinear optimization problem. Fortunately this version of the energy cost minimization is amenable to dynamic programming as explained next.

We use a dynamic programming method to calculate the value of $d_k(t)$ in the optimization problem (19).

Consider the recursive function:

$$\begin{aligned} Q(S, Y_z) &= Q([a, b], Y_z) \\ &= Min\left\{Q([a, \tfrac{a+b}{2}], Y^l_{z-1}) + Q([\tfrac{a+b}{2}, b], Y^r_{z-1})\right\} \end{aligned} \quad (25)$$

For $z = 1,\ldots, Z$ where Z is the number of recursion levels and $Z = \log(b-a)$ where Y^z is the set of all *I's* (scheduling intervals) at recursion level z, which should be optimally scheduled between time interval $S=[a\ b]$. For the highest recursion level Z, Y^z is the set of all I_k associated with task T_k for $k = 1,\ldots, K$. Function Q schedules tasks Y^z between a and b to obtain the *minimum energy cost*. Optimum scheduling of Y in time-interval $[a\ b]$ is equivalent to optimum scheduling of Y^l_{z-1} and Y^r_{z-1} in time-intervals $[a, (a+b)/2]$, and $[(a+b)/2, b]$, respectively, where

$$\begin{aligned} Y^l_{z-1} \cup Y^r_{z-1} &= Y_z \\ Y^l_{z-1} \cap Y^r_{z-1} &= \varnothing \end{aligned} \quad (26)$$

To solve the time slot-based energy cost optimization problem, we propose *Slot-Based-DP Algorithm*, which starts from the bottom level of recursion where $b - a = 1$ and builds up the optimum solution to the up level. Therefore, in the bottommost level, *Slot-Based-DP algorithm* operates/sweeps on all the time slots one-by-one.

Slot-Based-DP Algorithm:

1- The number of slots in the bottommost level is $W = b - a$.

2- For the bottommost level, z=1, the value of Q is evaluated for slots $i = 1,\ldots,W$ and for possible scheduling $V^z_n \in Y^z$, which is a vector with K elements and each element is an integer value less than or equal to $2^{(z-1)}$, where $n = 1,\ldots,N$, N is the number the vectors and $N = (2^{(z-1)}+1)^K$. For example for K=3, z=1, there are 2^3 vectors:

$$(0,0,0),(0,0,1),(0,1,0),\ldots,(1,1,1) \quad (27)$$

The value of Q for a slot S_i and a possible scheduling V_n in slot S_i is:

$$Q(S_i, V^z_n) = \left(C(a_i) - C(b_i)\right)R\left(p\left(a_i, V^z_n\right)\right) \quad (28)$$

where $S_i = [a_i\ b_i]$, b_i-a_i = 1 and

$$\begin{aligned} \cup S_i &= [a\,b] \\ \cap S_i &= \varnothing \end{aligned} \quad (29)$$

and function p can be calculated as follows:

$$p(a_i, V^z_n) = \sum_{k=1}^{K} V^z_n(k)P_{k,1} \quad (30)$$

Here $V^z_n(k)$ refers to the k^{th} element of vector V^z_n whereas $R(p)$ denotes a multi-level penalty function.

3-For the level z=2 to Z, build up optimum Q from Q of level z-1.

$$Q(S_i, V^z_n) = \underset{f,g}{Min}\left\{Q(S^l_i, Y^{z-1}_f) + Q(S^r_i, V^{z-1}_g)\right\} \quad (31)$$

where

$$S_i = S^l_i \cup S^r_i \quad (32)$$

and the S_l and S_r contains $2^{(z-1)}$ slots, and

$$V^z_n = Y^{z-1}_f + V^{z-1}_g \quad (33)$$

The constraints of optimization problem are:

1-Scheduling start-times and deadlines of tasks T, enforce that each task T_k cannot start earlier than starting-time s_k and cannot go beyond deadline e_k:

$$s_k < t < e_k \text{ where } d_k(t) = 1 \tag{34}$$

2- Each task T_k needs I_k^1 time unit to be done. Therefore,

$$I_k^1 = \sum_{t=1}^{O} d_k(t)\tau$$

3- The power usage of U at time t cannot go beyond its power capacity (Pmax):

$$p(t) < P_{\max} \tag{35}$$

III. SIMULATION RESULTS

The proposed algorithms have been implemented and tested for different cost functions and different scheduling tasks and constraints. The table below shows the performance of each method compared to a baseline greedy optimization method.

Simulation results show that the performance of the proposed methods respect to the baseline method is highly dependent on the shape of the base price and penalty function, scheduling parameters, power values and other parameters. For example, if the price is increasing from the beginning of the day to the end of the day, then most of the jobs will be scheduled at the beginning of the day. As another example, if the price function is constant, the proposed methods do not have too much gain over the baseline scheduling algorithm (a job is scheduled as soon as it is requested by a user). In order to assess the quality of the proposed methods, a variety of the base price and penalty functions are utilized for the simulations. Time granularity for the price function is 60 minutes. The durations of jobs/loads are assumed to follow a Gaussian distribution.

In the first experiment given in TABLE I , 100 jobs are scheduled by using the deadline-driven continuous-variable method. The cost function has low price at the beginning of the day and the end of the day. The penalty function is a bi-level function. In the second experiment, the price of energy is low at the beginning of the day and goes up at end of the day and 100 jobs are scheduled. In the third experiment, 50 jobs are scheduled while using a three-level penalty function. The price functions of the second and third experiments are similar. As given in TABLE I, deadline-driven method gives around two to 14 times energy cost reduction.

TABLE I Deadline-Driven Continuous-Variable Energy Cost Minimization

	Methods		K	Cost reduction factor of the proposed method
	Deadline-Driven (cents)	Baseline (cents)		
Expr. 1	580.4	8038.7	100	13.9
Expr. 2	1078	2052.3	100	1.9
Expr. 3	575	1098.4	50	1.91

TABLE II shows simulation results for the time slot-based cost minimization. The price functions of experiments 1 and 2 in TABLE II are similar to price functions of experiment 1 and 2 in TABLE I. However the scheduling tasks for two tables are different. Therefore the optimization results of them are not comparable.

TABLE II Time Slot-Based Energy Cost Minimization

	Methods		K	Cost reduction factor of the proposed method
	Time Slot-Based (cents)	Baseline (cents)		
Expr. 1	620.1	5105	100	8.2
Expr. 2	803	1020	200	1.27

IV. CONCLUSION

First, a formularization of cooperative energy cost optimization was presented. Next two different methods, deadline-driven continuous-variable and time slot-based, to optimize the energy cost of a networked community of cooperative users were described. The former method is appropriate for an energy cost optimization with less-interruptible tasks while the latter is appropriate for more-interruptible tasks. The method were implemented and tested for different test schemes. The results were compared to a baseline method. The results demonstrated huge cost savings under a number of different scenarios.

REFERENCES

[1] "How it works" Power Smart Pricing. http://www.powersmartpricing.org/how-it-works/

[2] Malina, J. "Market-Based Dynamic Pricing Integral to Smart Grid Implementation, Climate Change Mitigation" Compete: Electricity Competition is the Public Interest. http://www.competecoalition.com/blog/2009/12/market-based-dynamic-pricing-integral-to-smart-grid-implementation-climate-change-mitigation/

[3] Borenstein, S. "Policy and Economic Issues in Dynamic Electricity Pricing" http://www.youtube.com/watch?v=LdD4sYvDa08

[4] "Automated Critical Peak Pricing (Auto-CPP) Pilot for Large Commercial Facilities" PIER Demand Response Research Center. http://drrc.lbl.gov/drrc-4.html

[5] Borenstein, S. "Customer Risk from Real-Time Retail Electricity Pricing: Bill Volatility and Hedgeability" http://www.ucei.berkeley.edu/PDF/csemwp155.pdf

[6] "MathWorks - Accelerating the pace of engineering and science" http://www.mathworks.com/