Dual-pricing Policy for Controller-side Strategies in Demand Side Management

Siyu Yue, Jiong Chen, Yuantao Gu
Department of Electronic Engineering
Tsinghua University
Beijing, China
ysy07@mails.tsinghua.edu.cn, {chenjiong05, gyt}@tsinghua.edu.cn

Chenyue Wu
Institute for Interdisciplinary Information Sciences
Tsinghua University
Beijing, China
wucy05@mails.thu.edu.cn

Yiyu Shi
Electrical and Computer Engineering Department
Missouri University of Science and Technology
Rolla, MO, United States
yshi@mst.edu

Abstract—Nowadays, smart grid is drawing great emphasis in the U.S., Europe and China with the increasing awareness of the notion – green world. A whole bunch of new smart grid related technologies and projects are booming throughout the world. These technologies fundamentally change the way how demand side management is done in the grid. In this paper, we propose a demand side management model, where appliances are categorized into non-interruptible tasks and interruptible tasks with parameters that are easy to implement in smart meters. Within our model, we adopt a pricing policy which leads to an optimal user-side strategy and analyze several strategies for the grid controller to set the future prices for the power grid in order to flatten the aggregated profile. A novel dual-pricing policy which sets different prices for different types of tasks is also proposed. Simulation results show that our designed mechanism can help reduce the peak-to-average ratio and coefficient of variation of the aggregated profile very well.

I. INTRODUCTION

Demand side management (DSM) is a concept that has been proposed for more than three decades [1]. It refers to the modification of consumer demand for electricity through various methods. The goal of DSM is usually to match instant electricity consumption with generation, and to reduce peak loads. As current power generation and distribution infrastructure are designed to accommodate peak demands [2], a reduction in peak demand will greatly alleviate the required capacity of electricity generation [3].

DSM is often done by providing economic incentives for users to reduce or shift their loads from peak hours [2]. For example, nowadays in Beijing, peak-hour prices, off-peak prices and seasonal prices are applied to large customers [4]. In traditional power system, DSM only works in reactive and static ways. Neither energy providers nor energy consumers are able to respond actively to each other in real time. However, in smart grid, with various technologies, such as the underlying information infrastructure, available, DSM can be done in an active and automatic manner.

A lot of pioneer research has been done regarding DSM in smart grid but only within the last few years. For example, Allcott, in [5] and Chen, et al. in [6] performed economic analysis of electricity market and lay the foundation of using economic incentives for management. Mohsenian-Rad, et al. in [7] and Conejo, et al. in [8] modeled the energy consumption of one single user as optimization problems. The successful solving of these problems relies on prediction of future electricity prices. In [9], Caron et al. considered a system consisting of many users, and electricity price is set to be a non-decreasing function of instant aggregated load of all the users. Based on this model, analysis and several algorithms are given for different knowledge settings. In [10], Mohsenian-Rad et al. proposed a game theoretic model, where each user only cares about his/her own interests instead of global costs. By devising a good pricing policy, the Nash equilibrium of this game is proved to be unique and optimal both for each user and for the whole grid. In [11], Bakker et al. designed a three-step control strategy to optimize the overall energy efficiency and to increase the amount of generation based on renewable resources. In [12], He et al. considered the balancing of instant load and generation with the integration of wind energy, which introduces much uncertainty in generation.

In this paper, we aim to design a practical mechanism with an easy-to-implement user-side strategy. Our contributions can be summarized as follows:

- **Novel DSM model**: we propose a novel DSM model which categorizes tasks according to whether they are interruptible. Combined with our dual-pricing policy, it best flattens the total energy consumption profile.
- **Dual-pricing policy**: We introduce a dual pricing policy, which sets different prices for different type of tasks.
- **Controller-side strategies**: A family of controller-side strategies is analyzed and their performance based on this DSM model and pricing policy is compared.

The remainder of the paper is organized as follows. In Section II, we give detailed descriptions of our model and the pricing policy we will use. User-side strategy and controller-side strategies are demonstrated in Section III and Section IV, respectively. In Section V, we introduce the novel dual-pricing policy and improve the controller-side strategies. Simulation results of strategies from Section IV and V are shown and
assessed in Section VI. The concluding remarks and future work are addressed in Section VII.

II. MODEL DESCRIPTION

In our DSM model, we consider a 24-hour period, and this period is divided into \(H\) time slots. The system consists of an energy provider, a central controller, and many end users as demonstrated in Fig. 1.

![Architecture of the system](image)

Figure 1. Architecture of the system

Define the set of all users as \(N\). Define the set of all tasks of user \(n\) as \(M_n\). For each task \(m\) in \(M_n\), we define five parameters for it as follows:

- \(P_m\): Power for task \(m\)
- \(D_m\): Duration for task \(m\)
- \(S_m\): Earliest allowable start time for task \(m\)
- \(E_m\): Latest allowable end time for task \(m\)
- \(T_m\): Type of task \(m\)

\(P_m\) multiplied by \(D_m\) gives the total energy consumption of that task. \(S_m\) and \(E_m\) serve as the limitation on when the task should be completed. All the tasks are categorized into two types: non-interruptible and interruptible.

- **Non-interruptible tasks**, are those tasks that as long as they start, they cannot pause until finish, e.g., the micro oven, and cloth washer. Currently, most of the tasks are non-interruptible.

- **Interruptible tasks**, on the other hand, are the tasks that can pause or resume at any time, e.g. water heaters and battery chargers, as they are often not required to be done in a continuous period of time.

For task \(m\) of either type, if

\[
D_m < E_m - S_m, \quad (1)
\]

then there will be some flexibility in arranging task \(m\). For non-interruptible task, the actual start time \(a_m\) can be selected within the range:

\[
S_m \leq a_m \leq E_m - D_m. \quad (2)
\]

For interruptible task, let \(A\) donate the set of time slots in which the task will be carried out. Then it must follow:

\[
\left| A_m \right| = D_m, \quad (3)
\]

and

\[
\forall a \in A_m, S_m \leq a < E_m. \quad (4)
\]

Benefiting from the underlying communication network and the technology of smart appliances and smart meters, we are able to control the tasks automatically. Different pricing policies can also be adopted to provide incentives for the users to participate in demand side management actively through a smart software agent implemented in smart meters which represent their own interests.

In our model, instead of using real-time pricing (RTP) policy and time-of-use pricing (ToUP) policy [13], we adopt a day-ahead electricity pricing policy, as in day-ahead markets [14], where the grid operator will announce the day-ahead electricity price every day. As we will see in the next section, this pricing policy results in a very simple strategy on the users’ side.

III. USER-SIDE STRATEGY

As a vision in the future smart grid, the architecture of the end user is shown in Fig. 2. All the smart appliances of each user are connected to the grid through their smart meter. The power, duration and type of each task are directly communicated between appliances and the smart meter. All that the users have to do is to set the earliest allowable start time and the latest allowable end time for each task.

![Architecture of an end user](image)

Figure 2. Architecture of an end user

Given that the objective of each user is to minimize his cost, with the future prices available, this cost is minimized if and only if the cost of each single task is minimized. Therefore, the algorithm implemented in each smart meter is localized and simple. Let \(f(t)\) donate the electricity price at time \(t\). For non-interruptible task \(m\), the actual start time \(a_m\) which minimizes cost will be:

\[
a_m = \arg \min_{S_m \leq a_m < E_m} F_{T_m}(a_m), \quad (5)
\]

where

\[
F_{T_m}(a_m) = \sum_{t=0}^{T_m-1} f(t'). \quad (6)
\]

For interruptible task \(m\), the smart meter can simply choose \(D_m\) slots with the lowest prices out of \(S_m, S_m+1, \ldots, E_m-1\).

Note that we have assumed that each user will only shift his tasks rather than abandoning them due to high price. If users are forced to abandon their tasks due to high price, it will...
becomes meaningless to study the aggregated profile and rate different controller-side strategies.

IV. CONTROLLER-SIDE STRATEGY

On the controller’s side, the goal is to achieve a desirable aggregated profile. Generally, we want the profile to be as flat as possible. In our paper we use peak-to-average ratio (PAR) and coefficient of variation (CV) to measure the desirability of a profile. Note that CV is defined as the standard deviation divided by mean. PAR and CV are good indicators as PAR measures the max deviation to a perfectly flat profile and CV measures the average deviation.

In this section, we will analyze three controller-side strategies. The first two strategies require a-priori knowledge of the accurate task information. Furthermore, the first strategy requires the controller to have absolute control over all the tasks, while in the second and the third strategies management is done through pricing policy.

A. Dictator Strategy

In the dictator strategy, we assume the central controller is able to gather all the tasks’ information before the first time slot. And it is also able to arrange all the tasks. Under this setting, the central controller will only be able to achieve the optimal profile with unbounded computational power. Therefore, we propose a greedy algorithm as Algorithm 1, instead of enumeration. As will be shown in Section VI, this algorithm is able to achieve a good, if not optimal, profile with low PAR and CV. The basic idea of this greedy method is to add one task to the aggregated profile at a time, and each time find the arrangement of the task which minimizes the CV of the current profile.

Algorithm 1: Greedy method for the dictator

Input: \((P, D, S, E, T)\) of all the tasks
Output: Aggregated profile

1: Sort all the non-interruptible tasks according to their power in descending order.
2: For each non-interruptible task \(m\)
3: For each \(a_m\) from \(S_m\) to \(E_m-D_m\)
4: Calculate the CV of the current profile if task \(m\) starts at time \(a_m\).
5: End
6: Find \(a_m\) which minimizes the standard deviation.
7: Add task \(m\) to the current profile.
8: End
9: Sort all the interruptible tasks according to their power in descending order.
10: For each interruptible task \(m\)
11: Find \(D_m\) slots with lowest aggregated power out of the \(S_m, S_m+1, \ldots, E_m-1\) slots of the current profile.
12: Add task \(m\) to the current profile.
13: End

In the future smart grid, it is highly unlikely that users will authorize the grid to control their tasks due to security and privacy concerns. It renders the dictator strategy unpractical. Nonetheless, the dictator strategy can give us insight on what a good profile will be like, and it is meaningful to use this method as a benchmark.

B. Prophet Strategy

In the prophet strategy, we assume the central controller is able to gather accurate task information. Unlike dictator strategy, this central controller cannot directly arrange the tasks. It can only guide the users to arrange their tasks by setting different electricity prices at different time. The strategy adopted by each user is the one proposed in Section III, as it minimizes each user’s own cost.

Note that as the arrangement of each task is solely determined by the prices of different time slots, we can use enumeration to find the best price setting which minimizes CV of the aggregated profile. Genetic algorithms can also be used to reduce computational complexity. One merit of the enumeration and genetic algorithm is that computational complexity grows linearly with the number of tasks.

Applying enumeration, the prophet strategy is able to yield the best profile that can be achieved by pricing policy.

C. Statistician Strategy

In a large and real grid, it is often impossible for the controller to get accurate future task information of all the users. However, it is reasonable to expect that all the tasks of each day correspond to a certain stochastic model. From the controller side, let the number of tasks which arrive at time slot \(s\) (earliest allowable start time \(S\) equals \(s\)) be \(k_s\). We assume \(k_s\) follows a Poisson distribution with different expectation \(\lambda_s\) in different time slots:

\[
p_s (k_s) = \frac{\lambda_s^{k_s} e^{-\lambda_s}}{k_s!}.
\]

Of the tasks that arrive at time \(S\), we assume the distribution of \(P\) and \(D\) obeys a mixed Gaussian distribution, in which each Gaussian stands for one kind of appliance, e.g. air-conditioner, cloth washer or laptop. Only the weight \(\alpha_i\) of each Gaussian varies through time. \(E-(D+S)\), which measures the flexibility of arranging tasks, obeys an exponential distribution. Thus, the joint distribution of \(P\), \(D\) and \(E\) is:

\[
p_s (P, D, E) = \sum \alpha_i (S) \cdot p_{\text{Gauss}} (P, D) \cdot p_{\text{exp}} (E - (D + S)).
\]

Two stochastic processes with independent parameters will be used for each type of tasks.
In this statistician strategy, we assume the controller is only able to determine the parameters of this stochastic model. It must set the future prices without knowing the exact information of each task.

The algorithm we propose here is to randomly generate several sample cases according to the stochastic model. For each given price, we calculate the indicator (for example, CV) of the resulting profile of each sample case. Let $c_j$ denote the indicator of the profile of sample case $j$. We can use enumeration or genetic algorithm to find the best price setting that minimizes mean + 3 standard deviations of $c_j$.

The key to this strategy is to determine all the parameters of the stochastic model (7) and (8). Past data on price and profile alone are not sufficient, as they only give statistical information of the aggregated profile. Extra effort must be put to get detailed statistical information of tasks. For example, we can conduct a survey on the smart appliances. With the permission of users, anonymous information can also be collected from smart meters.

The above three strategies are summarized in Table I.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Information setting</th>
<th>Control method</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictator strategy</td>
<td>Exact information</td>
<td>Direct control</td>
<td>Greedy algorithm</td>
</tr>
<tr>
<td>Prophet strategy</td>
<td>Exact information</td>
<td>Price-based control</td>
<td>Enumeration / Genetic algorithm</td>
</tr>
<tr>
<td>Statistician strategy</td>
<td>Statistical information</td>
<td>Price-based control</td>
<td>Enumeration / Genetic algorithm</td>
</tr>
</tbody>
</table>

V. IMPROVEMENT BY DUAL-PRICING POLICY

The performance of uniform-pricing policy based strategies is limited by the fact that during peak hours, though the peak of the original profile is removed, new peaks will be created at the troughs of prices and troughs be created at the peaks of prices. It results in big fluctuations in the aggregated profile, as shown in Fig. 3 and Fig. 4 of Section VI.

This phenomenon is hard to avoid in all uniform-pricing policy based strategies as everyone tries to move their energy consumption from high-price periods to low-price periods.

In [11], Bakker et al. proposed a strategy to use different price settings for different users. Inspired by their work, we propose a dual-pricing policy, which sets different prices for non-interruptible tasks and interruptible tasks. Intuitively, under this dual-pricing policy, the peak of interruptible tasks will cancel the trough of non-interruptible tasks. And our policy is more feasible than using different price settings for different users as the type of each task is predetermined and it will not create any sort of disparity among users.

Based on the dual-pricing policy, we revise the prophet strategy and statistician strategy proposed in Section IV.

A. Dual-Pricing Prophet Strategy

In this strategy, we use a genetic algorithm to search for the best dual-price setting. With each price setting for non-interruptible tasks given, we determine the aggregated profile of all non-interruptible tasks. Then we use a price setting which is proportional to that aggregated profile as the price for interruptible tasks. Finally, the indicator of the aggregated profile of both types of tasks is calculated. The best dual-price setting will be determined based on this indicator.

B. Dual-Pricing Statistician Strategy

In this strategy, similar to that in Section IV, we use the stochastic model to generate several test cases, and apply genetic algorithm to search for the best dual-price setting that minimizes mean + 3 standard deviations of the indicator of all test cases.

It is important to note that in Section III and IV, the decision of each user is only determined by the relative value of the prices. Therefore, the prices yielded in the second and third strategies are also relative values. The absolute value of prices matters only when the profit of the energy provider is taken into consideration.

VI. SIMULATION RESULTS

In all the following simulations, we set the number of time slots $H$ to be 48, which means the price changes per 0.5 hour. For each test case, we generate around 5,000 tasks according to the stochastic model (7) and (8). For (8), we derive the parameters of the mixed Gaussian distribution of power and duration from an extensive survey on electrical appliances, mostly from the U.S. Department of Energy [15]. And we manually set the expectation of the Poisson distribution within the range $[0.2,0.4]$. Interruptible tasks account for around 10% of all tasks. In all the above strategies, CV is used as the indicator for optimization. In the prophet and the statistician strategies, genetic algorithms are used to reduce computational complexity. In the statistician strategy of both Section IV and Section V, we randomly generated 10 sample cases to train the best price setting.

As a comparison, we will also show the original profile in simulation results. The original profile is calculated as $a_i$ equals $S_i$ for each non-interruptible task and the first $D_i$ slots selected for each interruptible task.

A. Simulation Results of Section IV

The simulation results of the original profile and profiles achieved by different strategies for an example test case are shown in Fig. 3. The long dashed line depicts the original profile, which has high peak energy consumption. The dictator strategy achieves a flat profile in peak hours as shown by the solid line. Both uniform-pricing policy based strategies result in large fluctuations in peak hours, which is undesirable.
PAR and CV of the profiles in Fig. 3 are shown in Table II. In this test case, all three strategies reduce PAR by more than 15% and CV by more than 30%.

**TABLE II. PAR AND CV OF TEST CASE ONE**

<table>
<thead>
<tr>
<th></th>
<th>PAR</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original profile</td>
<td>1.9188(1)</td>
<td>0.5818(1)</td>
</tr>
<tr>
<td>Dictator strategy</td>
<td>1.3564(-29%)</td>
<td>0.3865(-34%)</td>
</tr>
<tr>
<td>Prophet strategy</td>
<td>1.5593(-19%)</td>
<td>0.3959(-32%)</td>
</tr>
<tr>
<td>Statistician strategy</td>
<td>1.5873(-17%)</td>
<td>0.4035(-31%)</td>
</tr>
</tbody>
</table>

Fig. 4 shows the simulations results of the prophet strategy with different price granularity, i.e., the profiles achieved by prophet strategy with price changes every one, two or three hours other than half an hour. As shown in this figure, the peak energy consumption becomes higher when price changes slower.

**TABLE III. PAR AND CV OF PROFILES IN FIG. 4**

<table>
<thead>
<tr>
<th>Price changes</th>
<th>PAR</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half an hour</td>
<td>1.5593(1)</td>
<td>0.3959(1)</td>
</tr>
<tr>
<td>One hour</td>
<td>1.5885(+2%)</td>
<td>0.3939(-1%)</td>
</tr>
<tr>
<td>Two hours</td>
<td>1.7873(+15%)</td>
<td>0.4075(+3%)</td>
</tr>
<tr>
<td>Three hours</td>
<td>1.8062(+16%)</td>
<td>0.4257(+8%)</td>
</tr>
</tbody>
</table>

**B. Simulation Results of Section V**

The simulation results of original profile and profiles achieved by prophet strategy and dual-price prophet strategy for the example test case are shown in Fig. 5.

Simulation results of profiles achieved by statistician strategy and dual-price statistician strategy for the example test case are shown in Fig. 6.

**TABLE IV. PAR AND CV OF PROFILES IN FIG. 5 AND FIG. 6**

<table>
<thead>
<tr>
<th></th>
<th>PAR</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original profile</td>
<td>1.9188(1)</td>
<td>0.5818(1)</td>
</tr>
<tr>
<td>Prophet strategy</td>
<td>1.5593(-19%)</td>
<td>0.3959(-32%)</td>
</tr>
<tr>
<td>Dual-pricing prophet strategy</td>
<td>1.4328(-25%)</td>
<td>0.3837(-34%)</td>
</tr>
<tr>
<td>Statistician strategy</td>
<td>1.5873(-17%)</td>
<td>0.4035(-31%)</td>
</tr>
<tr>
<td>Dual-pricing statistician strategy</td>
<td>1.5545(-19%)</td>
<td>0.3946(-32%)</td>
</tr>
</tbody>
</table>

PAR and CV of the profiles in Fig. 5 and Fig. 6 are shown in Table IV. Compared with uniform-pricing based strategies, dual-pricing based strategies achieve better profiles with lower PAR, CV and smaller fluctuations.

Below are the figures of PAR and CV of the profiles achieved by different strategies of five other test cases. These test cases are randomly generated according to the same model parameters. Therefore, the same price settings are used for all five test cases under statistician strategy and dual-pricing statistician strategy. As shown in Fig. 7 and Fig. 8, dual-price prophet strategy has the best performance on both PAR and CV.
among all price-based strategies. It even beats the dictator strategy on CV in some cases. In most of the cases, dual-pricing statistician strategy outperforms prophet strategy.

![Figure 7. PAR of profiles achieved by different strategies](image)

![Figure 8. CV of profiles achieved by different strategies](image)

VII. CONCLUSION

In this paper we propose a DSM model based on some prior work of other researchers. In our model we characterize the usage of smart appliances into two types of tasks, non-interruptible ones and interruptible ones. The type of each task is predetermined by the physical properties of the appliances. Information regarding the power, duration and type of each task is directly communicated between smart appliances and smart meters. Users need only set the time span in which the task must be finished. As shown in Section III, with future prices announced in advance, the smart meter can adopt a very easy algorithm to decide the arrangement of all the tasks which minimizes cost. Then we focus on the controller side and analyze several strategies for the central controller to manage all the tasks or to set the future prices under different assumptions. Furthermore, we propose an improvement on price-based strategies by setting different prices for non-interruptible tasks and interruptible tasks. Simulation results show that all these strategies reduced PAR by more than 15% and CV by more than 30% of the aggregated profile. The adoption of dual-price policy further enhances the performance of corresponding strategies. The fact that prophet strategy achieves profiles with lower PAR and CV than that of statistician strategy also informs us that the central controller should try to incent the users to have their future demands reported to the grid.

A lot of improvement work can be done on our DSM model. For example, we can study the performance of all the strategies if indicators other than CV are used to measure the aggregated profile. We can try to develop new algorithms which are less complex than genetic algorithms for price-based strategies. Our stochastic model of tasks can be polished to make it match reality better. We can even categorize the tasks further into smaller groups, which have different price settings each, and study how flat the aggregated profile will be.

REFERENCES


