ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-1167

Efficient Policy Iteration for Periodic Markov Decision Processes

Takayuki Osogami[1] and **Rudy Raymond**[2]

Abstract. We propose a solution to a new problem that is faced by steelworks, who own private thermal power-plants and plan to use batteries to absorb fluctuations in power demand. A major challenge is in controlling both the power generation and the use of batteries under such fluctuations. We formulate a Markov decision process (MDP) and design the states of the MDP so that it has a periodic structure to avoid the explosion of its state space. We then develop a policy iteration algorithm that exploits the periodic structure for computational efficiency. Numerical experiments suggest that the combination of the proposed MDP and the policy iteration allows us to find a control policy that can significantly reduce the electricity cost.

1 INTRODUCTION

For economical and environmental reasons, there has been a significant amount of research for optimizing the control of the generation or the use of electric power. This research is motivated by requirements not only from electric power companies but also from manufactures of steel, automobile, chemical, and other products that require a huge amount of electricity for production. The goal of this paper is to introduce and solve a new problem faced by leading steelworks in Japan who own private power plants and seek to optimize the way they generate *and* use electric power.

The prior work addresses various issues that appear in the generation or the use of electric power. An active area of research for optimally controlling the electric power plants is in unit commitment [12] for determining when to turn on and turn off their turbines. Energy storage devices for flexible and secure unit commitment are studied in [2, 9] to cope with the wind energy fluctuation.

There is also an increasing interest in batteries for balancing the use of electric power. Because there are various factors that can affect their performance, the charging schedule of batteries must be designed carefully. A problem of designing the charging schedule of batteries in electric vehicles from wind energy is studied in [17], and a study in using uninterrupted power supply (UPS) as energy storage instead of just for emergency for data centers is in [18].

The first contribution of this paper is to introduce and formulate a new problem in the steelworks that deals with both aspects of electric power generation and usage by exploiting storage capabilities of batteries. The new problem that we address appears in anonymous steelworks who own private power plants that are capable of generating electric power cheaper than directly buying from electric power companies. However, due to turbine constraints, the steelworks must commit the amount of the electric power it will use for every 30-minute period and do so 15 minutes prior to the beginning of the period. Undercommitment results in buying expensive electricity to cover the power deficit, and overcommitment results in wasting unused electricity. On the other hand, accurate prediction of the amount of power consumption is difficult in practice, because production schedule can change within the last minutes and power consumption depends on uncontrollable external factors such as the temperature. To reduce the cost of electricity that results from over- and undercommitment, the steelworks plan to use batteries to mitigate the uncertainties and fluctuations of the demand for electric power.

The second contribution of this paper is in using MDPs for optimizing the control of the generation and the use of electric power. We design the state space of the MDP so that it has a periodic structure that allows constructing an efficient algorithm.

The third contribution of this paper is a new approach of policy iteration for the MDP whose state space has a periodic structure. Our work follows a trend in the community where the computational complexity of policy iteration for MDPs having particular structures can be significantly reduced [7, 19]. However, unlike previous approaches, our proposed policy iteration avoids matrix inversion and other bottlenecks from matrix operation, and therefore, is faster, as shown from the numerical experiments.

2 A PROBLEM FROM STEELWORKS

Steelworks consume a huge amount of electricity for rolling slabs into coils, electric furnaces, and other manufacturing processes.[3] They thus often have private thermal power plants that can produce electricity at the cost lower than that provided by electric power companies. For efficient operation, power plants often shut down and start up some of their turbines to cope with the changes in power demand. However, due to the delay in starting turbines, the operation of the turbines needs to be planned in advance by taking into account the uncertainty in the power demand. This planning is called unit commitment and has been well studied in the literature. Our work can be considered as a preprocessing step prior to the unit commitment and its output decision can be utilized by a unit commitment solver.

In practice, the steelworks cannot commit the exact amount of power to be used in a period due to the uncertainty in the demand. The use of batteries to store the power surplus during a period of overcommitment to be used later at a period of undercommitment is a solution of the steelworks. Notice that, with the ability to store surplus for later use, it is not necessarily an optimal choice to commit the exact amount of power that is predicted for the next period.

We must take into account various factors to appropriately charge and discharge batteries. First, a nonnegligible fraction of power is

[1] IBM Research - Tokyo, email:osogami@jp.ibm.com
[2] IBM Research - Tokyo, email:raymond@jp.ibm.com

[3] According to Japan Statistical Yearbook 2012, the iron and steel industry in Japan consumed 70.5 TWh of electric power in 2009, which amounts to 18.3% of the total consumption in Japan.

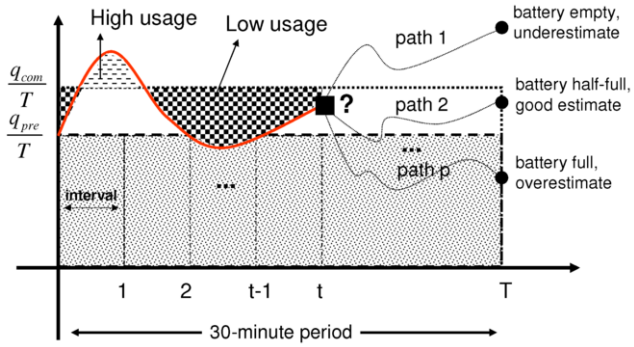


Figure 1. Fluctuation of power demand in a period

lost when batteries are charged and discharged. For example, approximately 10% or more of the power is lost upon charging lead-acid batteries and sodium-sulfur batteries. Second, a careless charging and discharging policy can shorten the life time of batteries. For example, the life time can be shortened far more quickly when they are used at low state of charge (SOC) [14, 13]. Third, the rate at which batteries can be charged and discharged is constrained, and can also affect their life time.

3 MODELING WITH MDP

We design the MDP to optimally decide the amount of power to be generated and the amount of power to be stored to (or released from) batteries. The idea behind the MDP formulation for the steelworks is described in Figure 1. Let $q_{\rm com}$ be the committed amount of power (as bounded by the larger rectangle in the figure), and $q_{\rm pre}$ be the predicted amount of power usage in the 30-minute period (as bounded by the smaller gray rectangle in the figure). Dividing the period into T intervals, one can see that, if the steelworks avoid underestimation, then the average of power usage per interval is at most $q_{\rm com}/T$.

However, the actual power demand fluctuates. The curved line illustrates the *average* power consumed since the beginning of the 30-minute period. The steelworks have opportunities to store the unused power to batteries when the actual average usage is below the average commitment (low usage region in the figure), and to release it when the actual average usage is above the averaged commitment (high usage region in the figure). These opportunities will be weighed against other internal and external factors of batteries. Moreover, when deciding the commitment amount in the next period, the current state of batteries and total amount of committed, predicted and actual power usage can be used to estimate the probabilities of ending the period with empty batteries (underestimate in path 1 in the figure), or fully charged batteries (overestimate in path p in the figure). An MDP is used to guide decisions of the counter actions by modifying the amount of charge/discharge decisions in the rest of the intervals and by committing an appropriate amount of power for the next period.

The MDP is a 4-tuple $\langle \mathcal{S}, \mathcal{A}, P(\cdot \mid \cdot, \cdot), C(\cdot, \cdot) \rangle$, where $\mathcal{S}$ is a set of finite states, $\mathcal{A}$ is a set of finite actions, $P(s_{t+1} \mid s_t, a_t)$ is the transition probability of moving to state s_{t+1} when taking action a_t in state s_t, and $C(s_t, a_t)$ is the cost of taking action a_t in state s_t. We describe each component of the tuple below. For simplicity, we fix the length of period to 30 minutes, and the time to commit the amount of power to 15 minutes prior to the beginning of the period.

We assume that the parameters of the MDP are set in advance. The parameters can be determined based on the electric tariff, the fuel cost, and the specifications of batteries.

The set of finite states $\mathcal{S}$ is one of the key features of our formulation. We divide the period of 30 minutes into (finite) T intervals of length $30/T$ each, and for each interval $t \in \{1, 2, \ldots, T\}$ define a set of possible states $\mathcal{S}_t$. Each state is designed to store the predicted values of power deficit or surplus (i.e., the difference between power consumed and committed) at the end of the current period and in the next period. We refer to these estimated values as the *power balance values* of the current and next period.

Formally, each of $s_t \in \mathcal{S}_t$ is represented by

$$s_t \equiv (t, d_t, d', u_t), \tag{1}$$

where t is the interval number in the current period, d_t is the power balance value of the current period, d' is the power balance value of the next period, and u_t is the state of the battery at t. A major benefit of our definition of states is that it does not explicitly include the amount of power consumption and commitments in the current and succeeding period. Such amount may vary for each period and can do so by orders of magnitude.

Notice that d_t varies with t since we can use the knowledge of the total power consumed up to the beginning of the interval t to refine the power balance value of the current period. Specifically, we use the linear model

$$d_t \equiv \left(q_t + \frac{T-(t-1)}{T} q_{\rm pre} \right) - q_{\rm com}. \tag{2}$$

The first term on the right-hand side is the prediction of the consumed power, where q_t is the total power consumed up to the interval t, and $\frac{T-(t-1)}{T} q_{\rm pre}$ is the estimation of the total power consumed from t to the end of the period (we simply estimate that each interval from t consumes $q_{\rm pre}/T$ power). The second term on the right-hand side is the committed power for the current period (which was decided 15 minutes prior to the beginning of that period).

The set of actions, shown later, use these predicted values of power balance and the battery state u_t to decide either to store or use energy from batteries. Roughly speaking, under power surplus (deficit) and the availability of uncharged (charged) batteries, the action tends to charge (discharge) so that the power deficit or surplus at the end of the period, $d_T = q_T - q_{\rm com}$, becomes close to 0.

The power balance value at the succeeding period, d', is $q'_{\rm pre} - q'_{\rm com}$ for $t \geq T_{\rm com}$, or $\emptyset$ (i.e., not used) for $t < T_{\rm com}$. Here, $q'_{\rm pre}, q'_{\rm com}$ are, respectively, the total amount of predicted and committed power for the next period, and $T_{\rm com}$ is the interval that includes the time deadline to commit the amount of power generated at the succeeding period. Let $\mathcal{D}_t$ be the set of (finite) possible values of d_t for $t = 1, \ldots, T$, and let $\mathcal{D}'$ be that of d'. Let $\mathcal{U}_t$ be the set of (finite) possible values of u_t for $t = 1, \ldots, T$.

The set of actions, $\mathcal{A}$, consists of possible action sets $\mathcal{A}_t$ of charging or discharging batteries at the beginning of interval $t \in \{1, 2, \ldots, T\}$, and committing to generate $q'_{\rm com}$ energy in the succeeding 30-minute period at $t = T_{\rm com}$. Each action $a_t \in \mathcal{A}_t$ takes the following form

$$a_t \equiv (r_t, d'), \tag{3}$$

where $|r_t|$ is the amount of power charged to (or, discharged from) batteries during the t-th interval if $r_t \geq 0$ (or, $r_t < 0$), and d' is the power balance value of the succeeding period. Notice that $d' = \emptyset$ (i.e., ignored) if $t \neq T_{\rm com}$, and the amount of committed energy is obtained from $q'_{\rm pre} - d'$ otherwise. Clearly, the possible values of all

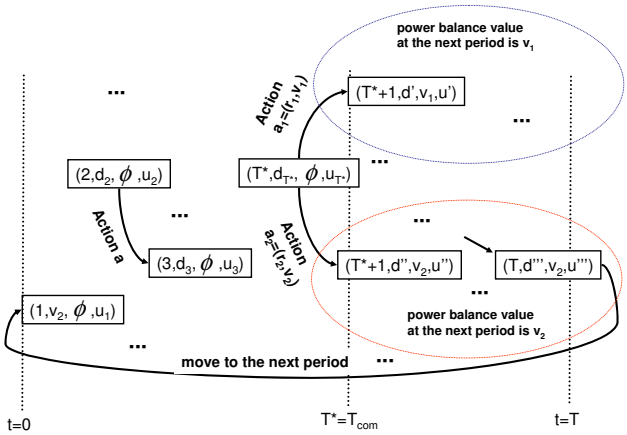


Figure 2. State transition

d' in a_t's are $\mathcal{D}'$. Let $\mathcal{R}$ denote the possible values of r_t, which is primarily determined by the maximum speed at which the batteries can be charged or discharged during an interval in the period.

The combination of states and actions defines the transition probability $P(s_{t+1} \mid s_t, a_t)$, which is the probability of moving to state $s_{t+1} \in \mathcal{S}_{t+1}$ when performing $a_t \in \mathcal{A}_t$ at state $s_t \in \mathcal{S}_t$. For example, taking an action $a_t = (r_t, d'')$ in state $s_t = (t, d_t, d', u_t)$ at interval $t \notin \{T_{\rm com}, T\}$, we transition to $(t+1, D_{t+1}, d', U_{t+1})$, where D_{t+1} and U_{t+1} are the random variables for power balance values and battery states at $t+1$. If the amount of consumption were exactly $q_{\rm pre}/T$ at t, we would have $D_{t+1} = d_t + r_t$ surely. However, in practice the power balance value at t is inaccurate due to the fluctuation of q_t and q_{t+1}. We model these inaccuracies and fluctuations as random transitions by estimating the probability distribution of $D_{t+1} - (d_t + r_t)$ using the data about past predictions and actual usage of electric power.

Notice that from the periodicity, we have $t = T+1$ is equal to $t = 1$, and the non-zero transition probability is possible only from $s_T = (T, d_T, d', u_T)$ to $s_1 = (1, d_1, \emptyset, u_1)$ if $d_1 = d'$ (i.e., setting the power balance value at the beginning of the succeeding period). This and other representative transitions are illustrated in Figure 2. The non-zero transition probability at $t = T_{\rm com}$ is possible only from $s_t = (t, d_t, \emptyset, u_t)$ to $s_{t+1} = (t+1, d_{t+1}, d', u_{t+1})$ when the action is $a_t = (r_t, d')$. Moreover, since we cannot change the commitment made at $t = T_{\rm com}$, the non-zero transition probabilities at $t > T_{\rm com}$ are only between states with the same power balance value of the next period, i.e., $s_t = (t, d_t, d', u_t)$ to $s_{t+1} = (t+1, d_{t+1}, d', u_{t+1})$.

We consider four types of costs to take into account short and long term benefits of actions. The first type of cost is determined by the amount of electricity purchased from the power company when the commitment was an underestimate. The second is by the amount of electricity that the private power plant generated but not used. The third is the cost associated with the electric power that is lost upon charging or discharging batteries. The fourth is the cost associated with shortening the life time of the batteries.

The first two types of costs are incurred at the end of the period (i.e., at the end of the interval T), depending on a_T and s_T. We denote these costs as $C_P(a_t, S_t)$ which is zero if $t \neq T$. Notice that the first cost is approximately twice as high as the second, and therefore, an optimal sequence of actions will guide to actions of generating power from the private plant. However, since the second cost imposes limiting the amount of wasted power, unexpected fluctuation of power consumption can occasionally cause underestimation.

The third type of cost is denoted by $C_E(r_t)$ and depends on the efficiency of charging and discharging batteries by the amount of r_t. For simplicity, we can assume that the power is lost only at the time of charging. Suppose that the fraction of the power that is lost from charging to discharging is α. Then, assuming that the unit cost of power from the private plant is γ, we set $C_E(r_t) = \frac{\alpha}{1-\alpha} r_t \gamma$ for $r_t > 0$. On the contrary, we set $C_E(r_t) = 0$ when $r_t \leq 0$.

The fourth type of cost is denoted by $C_L(s_t, r_t)$ and depends on the characteristics of the batteries under consideration and the rate of (dis)charge, r_t, of action a_t. For example, being at a particular SOC or (dis)charging at high rate can shorten the life time of batteries.

In summary, we define, $C(s_t, a_t)$, the cost of taking action $a_t = (r_t, d')$ at state s_t as

$$C(s_t, (r_t, d')) \equiv C_P(s_t, r_t) + C_E(r_t) + C_L(s_t, r_t). \quad (4)$$

4 EFFICIENT POLICY ITERATION

We seek to obtain the optimal policy that assigns an action to each state in such a way that the average expected cost is minimized. Because our MDP is periodic, the average expected cost is defined with the Cesaro limit,

$$\lim_{n\to\infty} \frac{\mathsf{E}[C_1] + \mathsf{E}[C_2] + \cdots + \mathsf{E}[C_n]}{n}, \quad (5)$$

where C_k denotes the cost incurred at the k-th transition of our MDP. We do not discount the future cost, because our MDP has a short time step such that a few hours ahead would be quickly ignored even with a moderate discount factor of 0.99. The approaches whose convergence relies on the discount factor converge very slowly when the discount factor is very close to 1. Not using a discount factor is essentially equivalent to using the discount factor very close to 1.

4.1 Policy iteration

Standard approaches for finding optimal policy that minimizes the average expected cost of an MDP include value iteration, linear programming, and policy iteration (see Chapter 8 from [15]). The prior work on policy iteration that exploits the structure of the MDP for computational efficiency [19, 7] motivates us to investigate policy iteration for our MDP that has a particular structure resulting from the 30-minute period. Policy iteration consists of four steps: (Step 1) Select an initial policy, (Step 2) Evaluate the policy, (Step 3) Update the policy, (Step 4) End if there is no update; return to Step 2 otherwise.

A bottleneck of policy iteration is in policy evaluation, which requires to solve the system of equations

$$\mathbf{c} = g\,\mathbf{1} + (\mathbf{I} - \mathbf{P})\,\mathbf{h}, \quad (6)$$

where we assume that the MDP is unichain (see Corollary 8.2.7 from [15]). For readability, we omit the description for multichain.

In (6), g and $\mathbf{h}$ are variables that represent the performance of a given policy, π, that we are evaluating; $\mathbf{c}$ and $\mathbf{P}$ are given by the parameters of the MDP and π; $\mathbf{1}$ denotes the vector whose elements are all 1, and $\mathbf{I}$ denotes an identity matrix. Specifically, g is referred to as gain in the literature and denotes, in our context, the average cost incurred per transition when we follow π. Also, $\mathbf{h}$ is called bias. An element of $\mathbf{h}$ is associated with a state and denotes $\lim_{n\to\infty} \sum_{k=1}^{n} (\mathsf{E}[C_k] - g) + \beta$, where β is an arbitrary constant, and C_k denotes the cost associated with the k-th transition, starting

from that state following π. Notice that (6) does not determine the solution uniquely. If $(g^\star, \mathbf{h}^\star)$ is a solution of (6), then $(g^\star, \mathbf{h}^\star + \beta\,\mathbf{1})$ is also a solution of (6) for any β. Finding any solution of (6) would suffice for the purpose of policy iteration.

4.2 Policy evaluation for periodic MDPs

We say that an MDP, $\langle \mathcal{S}, \mathcal{A}, P(\cdot \mid \cdot,\cdot), C(\cdot,\cdot)\rangle$, is periodic, if $\mathcal{S}$ can be partitioned into into $\{\mathcal{S}_t\}_{t=1,\ldots,T}$ for $T \geq 2$ such that $P(s' \mid s,a) = 0, \forall (s',s,a) \in \mathcal{S}_{t'} \times \mathcal{S}_t \times \mathcal{A}$ such that $t' - 1 = t \bmod T$. Note that a periodic MDP has an infinite horizon and is different from a finite-horizon MDP. We will solve the infinite-horizon MDP without approximation by exploiting the periodic structure.

Given π, an MDP is reduced to a Markov reward process. Let $\mathbf{P}$ be the transition probability matrix for a π. For a periodic MDP, we can arrange the states so that $\mathbf{P}$ has the following structure:

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{P}_{1,2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{2,3} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & & & \ddots & \mathbf{P}_{T-1,T} \\ \mathbf{P}_{T,1} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \end{pmatrix} \tag{7}$$

Here, an element of $\mathbf{P}_{t,t+1}$ denotes the probability of transitioning to a state in $\mathcal{S}_{t+1}$ from a state in $\mathcal{S}_t$. An element of $\mathbf{c}$ denotes the expected cost incurred immediately from a state when an action is selected based on π. Analogously to (7), let

$$\mathbf{c}^{\mathrm{T}} = (\ \mathbf{c}_1^{\mathrm{T}} \quad \mathbf{c}_2^{\mathrm{T}} \quad \cdots \quad \mathbf{c}_T^{\mathrm{T}}\ ) \tag{8}$$

$$\mathbf{h}^{\mathrm{T}} = (\ \mathbf{h}_1^{\mathrm{T}} \quad \mathbf{h}_2^{\mathrm{T}} \quad \cdots \quad \mathbf{h}_T^{\mathrm{T}}\ ). \tag{9}$$

The key idea in our approach is to first evaluate the bias for a subset of states. We start by evaluating the bias for the subset having the minimum number of states. Without loss of generality, let $\mathcal{S}_1$ be that subset and be referred to as a set of core states.

To evaluate the bias for $\mathcal{S}_1$, we calculate

$$\mathbf{Q} = \mathbf{P}_{1,2}\,\mathbf{P}_{2,3}\cdots\mathbf{P}_{T-1,T}\,\mathbf{P}_{T,1} \tag{10}$$

$$\mathbf{b} = \mathbf{c}_1 + \mathbf{R}_{1,2}\,\mathbf{c}_2 + \mathbf{R}_{1,3}\,\mathbf{c}_3 + \cdots + \mathbf{R}_{1,T}\,\mathbf{c}_T, \tag{11}$$

where $\mathbf{R}_{1,k} \equiv \mathbf{P}_{1,2}\,\mathbf{P}_{2,3}\cdots\mathbf{P}_{k-1,k}$ is the transition probability matrix from states in $\mathcal{S}_1$ to states in $\mathcal{S}_k$. The (i,j) element of $\mathbf{Q}$ is the probability of transitioning to the j-th core state given that the Markov reward process transitions from the i-th core state in the preceding cycle. Then the i-th element of $\mathbf{b}$ denotes the expected cumulative cost from when the Markov reward process transitions out of the i-th core state to when it transitions back to a core state.

Observe that $\mathbf{Q}$ and $\mathbf{b}$ define a Markov reward process on $\mathcal{S}_1$. An element of $\mathbf{Q}$ denotes the probability of transitioning from a core state to another core state (via states in $\mathcal{S}_2, \ldots, \mathcal{S}_T$). An element of $\mathbf{b}$ denotes the expected cost to be incurred from a core state (until coming back to a core state via states in $\mathcal{S}_2, \ldots, \mathcal{S}_T$). Analogously to (6), we thus need to solve the following system of equations to obtain g and $\mathbf{h}_1$:

$$\mathbf{b} = T\,g\,\mathbf{1} + (\mathbf{I} - \mathbf{Q})\,\mathbf{h}_1 \tag{12}$$

We can solve (12) using an iterative algorithm such as conjugate gradient [16]. Although neither (12) nor (6) determines the solution uniquely, it suffices to find one from the iterative algorithm.

Once a pair of g and $\mathbf{h}_1$ is obtained as a solution of (12), we can obtain $\mathbf{h}_t$ for $t = 2, \ldots, T$ recursively from (with $\mathbf{P}_{T+1} = \mathbf{P}_1$ and $\mathbf{h}_{T+1} = \mathbf{h}_1$),

$$\mathbf{h}_t \quad = \quad \mathbf{c}_t - g\,\mathbf{1} + \mathbf{P}_{t,t+1}\,\mathbf{h}_{t+1}. \tag{13}$$

It is straightforward to verify that the solution, $(g, \mathbf{h})$, satisfies (6).

5 NUMERICAL EXPERIMENTS

We now evaluate the effectiveness of the proposed policy iteration with experiments.

We divide a 30-minute period into intervals of length ΔT minutes each, and vary ΔT such that ΔT divides 30. The number of intervals in a period is $30/\Delta T$ and each is labelled as $t = 1, 2, \ldots, 30/\Delta T$. At $t = t_c \equiv \lceil 15/\Delta T \rceil$, the amount of power to be generated during the succeeding period is determined.

We consider a battery having the capacity of $C = 4$ MWh and the power of $P_{\mathrm{discharge}} = 18$ MW, which corresponds to a 100 t lead-acid battery of 0.04 MWh/t and 0.18 MW/t. The charging rate is set $P_{\mathrm{charge}} = 4$ MW. For simplicity, we assume that, for each interval, the battery is either charged at the full rate of P_{charge}, or discharged at the full rate of $P_{\mathrm{discharge}}$, or is left unchanged.

Recall that at the beginning of an interval t, the state s_t of the MDP is in the form (t, d_t, d'_t, u_t), as in Eq. (1), and an action a_t of the MDP is in the form (r_t, d''_t). When $d'_t \neq \emptyset$, the state s_t is a state that commits to generate $q'_{\mathrm{pre}} - d'_t$ MWh, where q'_{pre} is the given predicted power consumption, and d'_t is the power balance value of the next period whose value is chosen between $-C/2$ and $C/2$.

We discretize the power into the unit of ΔP MW and vary ΔP in the following experiments. The internal state (SOC) of a battery is correlated to the amount of electricity charged. Thus, u_t can be represented by an integer in the range $\left[0, \left\lceil \frac{60\,C}{\Delta T\,\Delta P} \right\rceil\right]$. Analogously, d'_t, d''_t, and r_t are integers in the range $\left[-\left\lceil \frac{30\,C}{\Delta T\,\Delta P} \right\rceil, \left\lceil \frac{30\,C}{\Delta T\,\Delta P} \right\rceil\right]$. Notice that although batteries are charged or discharged at full speed, the charge (discharge) is terminated when the batteries become full (empty). Hence, the possible values of r_t vary between states depending on the SOC (or, u_t).

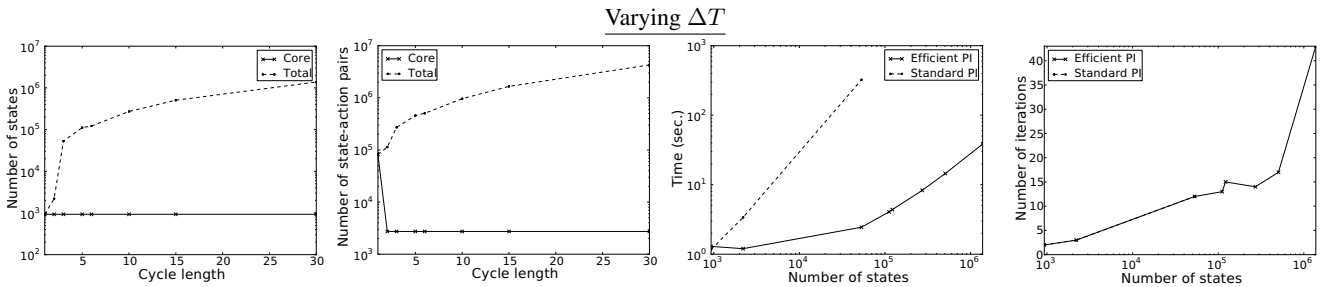
As shown in the previous section and Fig. 2, when we take an action $a_t = (r_t, d''_t)$ at the state $s_t = (t, d_t, d'_t, u_t)$ for $t < T/\Delta$, the non-zero transition probabilities are only to states of the form $(t+1, D_{t+1}, d'_{t+1}, u_{t+1})$ with $D_{t+1} = d_t + r_t + Y_t$, where Y_t is a random variable representing the fluctuation of the power consumption in the interval t (i.e., the curved line in Fig. 1). Therefore, the power balance value of the current period, d_t, in the state s_t satisfies

$$d_t = d_0 + \sum_{i=1}^{t} r_i + \sum_{i=1}^{t} Y_i, \tag{14}$$

where d_0 is the power balance value at the beginning of the period.

We assume that the amount of power consumption in a period, $\sum_i Y_i$, follows a Brownian motion. We adjust the parameters of the Brownian motion so that at the end of the period its mean is the predicted amount of power consumption (q_{pre}), and its standard deviation is $\Sigma = 12$ MWh. This implies that the standard deviation of the power consumption at each interval is $\sigma = \frac{\Sigma}{\Delta P}\sqrt{\frac{\Delta T}{T}}$. When taking action (r_t, d''_t) at state (t, d_t, d'_t, u_t), the probability of transitioning into state $(t+1, d_t + r_t + Y_t, d'_{t+1}, u_{t+1})$ can be approximated from

$$\Phi\left(\frac{Y_t + 0.5}{\sigma}\right) - \Phi\left(\frac{Y_t - 0.5}{\sigma}\right) \tag{15}$$



(a) States (b) State-action pairs (c) Time per iteration (d) Number of iterations

Figure 3. The left two figures show how (a) the number of states and (b) the number of state-action pairs grows as functions of cycle length, $T/\Delta T$. The right two figures show the performance: (c) the running time per iteration and (d) the number of iteration.

by rounding off the small probabilities of 0.005, where Φ denotes the cumulative distribution function of the standard normal distribution.

We consider two types of costs: the inefficiency of batteries and mispredicting the amount of power to be generated. We consider batteries with leakage percentage $\alpha = 13\ \%$. The cost of the power leakage is \$10/MWh. Therefore, the power of 1/0.87 unit is necessary to store one unit in the battery ($1/0.87-1$ is lost per unit stored). Such cost is associated with every action (r_t, d_t''), with $r_t > 0$. Meanwhile, the price of the electricity to cover underestimation is \$200/MWh, and the cost of the unused electricity due to overestimation is \$100/MWh. Such cost is associated with transitions that move states to the next period.

All of the experiments are carried out with Python 2.7.2 on a Windows machine with two Intel Xeon Processor E5503 and 132 GB RAM. For fair comparison, only one node of the CPU is used throughout the experiments. The system of linear equations in (12) is solved with Python implementation of LSMR 1.0.1 [3]. We examine the running time of the efficient policy iteration in two settings related to the key parameter that affect its running time: the number of core states (i.e., $|\mathcal{S}_1|$). We vary the number of intervals (or cycle length), keeping the number of core states unchanged, and vice versa.

We first vary ΔT, keeping $\Delta P = 8$ MW. Figure 3 (a)-(b) shows how the number of states and the number of state-action pairs grow as the cycle length, $T/\Delta T$, increases. The dashed line with dots shows the total number of states in Figure 3 (a) and the total number of state-action pairs in Figure 3 (b). The solid lines with cross marks show the corresponding number of core states in both figures. Observe that the total number of states grows quickly, while the number of core states remains unchanged.

Figure 3 (c)-(d) shows the running time of our efficient policy iteration and standard policy iteration. The dashed line with dots shows the running time of standard policy iteration, and the solid line with cross marks for the efficient policy iteration. Specifically, Figure 3 (c) shows the running time for an iteration (i.e., one policy evaluation and one policy improvement), and Figure 3 (d) shows the number of iterations. Total running time can be obtained by multiplying a value in Figure 3 (c) and a corresponding value in Figure 3 (d). Because the number of iterations does not vary between efficient policy iteration and standard policy iteration, only one curve can be seen in Figure 3 (d). Observe that the running time of efficient policy iteration is shorter than that of standard policy iteration by orders of magnitude when there is a large number of states.

We next vary ΔP, keeping $\Delta T = 15$. Figure 4 (a)-(b) shows how the number of states and the number of state-action pairs grow as $P/\Delta P$ increases, analogously to Figure 3 (a)-(b). Notice that $P/\Delta P$ represents the number of levels of the SOC. Namely, the SOC can be represented by an integer in $[0, \lceil P/\Delta P \rceil]$. Both the total number of states and core states grow with the number of charge levels.

Figure 4 (c)-(d) shows the running time of our efficient policy iteration and its degradation, analogously to Figure 3 (c)-(d). The degraded version evaluates the system of equations (12) by calculating the inverse of a matrix instead of using an iterative approach, LSMR. The running time of the degraded version roughly corresponds to the policy iteration of [7], which exploits the structure of the MDP that is skip-free in one direction. The solid line with cross marks shows the running time of efficient policy iteration, and the dashed line with dots shows that of its degradation. The running time of the degraded version grows far more quickly than that of efficient policy iteration due to the matrix inversion. Also, the inverse of a sparse matrix can be dense, resulting in memory constraints for large core states.

6 RELATED WORK

The existing approaches to optimizing the control of electric power include mathematical programming [17, 2, 9] and MDPs [10, 20]. The solution with MDPs has the desirable flexibility of allowing the action to depend on the state under consideration.

The design of the state space of an MDP largely determines the computational complexity and the quality of its solution. In the prior work, including [10, 20], a state of an MDP includes the information about the demand of electric power at or by the corresponding time, and the state space is defined based on the probability distribution of demand forecast. On the other hand, the key feature of our state space is the information about the differences between the forecasted and actual demand. Although the state space with the demand of electric power does not show a periodicity, the state space with the differences does under mild assumptions. This periodicity also stems from the fact that the steelworks must make a commitment on the amount of electric power for every period.

Popular algorithms for finding optimal policies for MDPs include value iteration, linear programming (LP), and policy iteration [15]. The best algorithm depends on instances and purposes. For example, [4] designs an approximate LP for a subclass of factored MDP with particular graph structures, and policy iteration is used for online planning in large MDPs with Monte-Carlo tree search [1]. Many algorithms can benefit from our definition of the state space, but we further show that policy iteration can exploit the periodic structure of the state space for computational efficiency.

White [19] studies policy iteration for the MDP whose state transitions follow a quasi-birth-and-death (QBD) process for any policy. Our periodic MDP does not have the QBD structure. Lambert

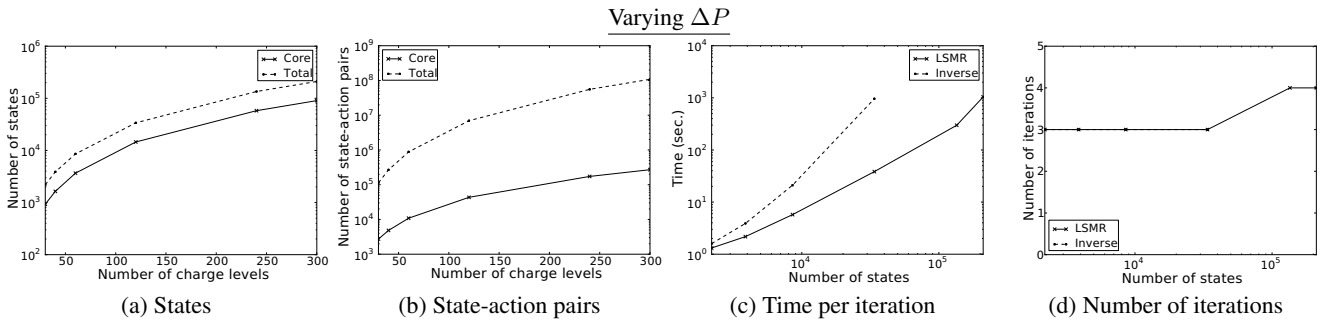


Figure 4. The left two figures show how (a) the number of states and (b) the number of state-action pairs grows as functions of cycle length, $P/\Delta P$. The right two figures show the performance: (c) the running time per iteration and (d) the number of iteration.

et al. [7] study policy iteration for the MDP whose state transitions follow a Markov chain that is skip-free in one direction, where the state space can be divided into ordered subsets of states such that "backward" transitions are allowed only between neighboring subsets of the states, while "forward" transitions are allowed between any subsets. The class of the Markov chains that are skip-free in one direction includes a QBD process and our periodic MDP. However, it is unclear if the techniques in [7] work when the future cost is not discounted because they require inversion of matrices that become singular when translated into our formulation. Our policy iteration avoids matrix inversion and runs faster than that of [7].

Jacobson et al. [6] study the periodic MDP with an additional structure that the MDP is time-homogeneous within a period, which we do not assume. Their focus is in giving a guarantee on the quality of approximation that assumes initially stationary policies when a period is long. Our approach does not approximate but efficiently finds the optimal policy by exploiting periodicity. The project management studied in [6] is another application of periodic MDPs.

7 CONCLUSION

We have introduced a problem faced by leading steelworks that requires controlling both power-generation and battery-use when future power demand is uncertain. Our solution is an MDP whose state space has a periodic structure to avoid the explosion of its state space. The optimal policy can be found efficiently with policy iteration due to the MDP's periodic structures. Value iteration does not appear to get much benefit from such structures beyond what can be exploited by existing techniques of compressing the state space [5]. There exists a wide range of techniques for efficiently analyzing structured Markov chains either exactly or approximately [8, 11], and it is an interesting direction to investigate how these techniques can help designing efficient algorithms for structured MDPs.

Acknowledgments

A part of this research was supported by JST, CREST.

REFERENCES

[1] H. Baier and M. H. M. Winands, 'Nested Monte-Carlo tree search for online planning in large MDPs', in *Proceedings of the 20th European Conference on Artificial Intelligence*, pp. 109–114, (2012).

[2] A. Daneshi, M. Khederzadeh, N. Sadrmomtazi, and J. Olamaei, 'Integration of wind power and energy storage in SCUC problem', in *Proceedings of World Non-Grid-Connected Wind Power and Energy Conference*, pp. 1–8, (2010).

[3] D. C.-L. Fong and M. A. Saunders, 'LSMR: An iterative algorithm for sparse least-squares problems', Technical Report Report SOL 2010-2R1, Systems Optimization Laboratory, Stanford University, (2011).

[4] N. Forsell and R. Sabbadin, 'Approximate linear-programming algorithms for graph-based Markov decision processes', in *Proceedings of the 17th European Conference on Artificial Intelligence*, pp. 590–599, (2006).

[5] J. Hoey, R. St-aubin, A. Hu, and C. Boutilier, 'SPUDD: Stochastic planning using decision diagrams', in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 279–288, (1999).

[6] M. Jacobson, N. Shimkin, and A. Shwartz, 'Markov decision processes with slow scale periodic decisions', *Mathematics of Operations Research*, **28**(4), 777–800, (2003).

[7] J. Lambert, B. Van Houdt, and C. Blondia, 'A policy iteration algorithm for Markov decision processes skip-free in one direction', in *Proceedings of the International Workshop on Tools for solving Structured Markov Chains (SMCtools)*, (2007).

[8] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ASA-SIAM, Philadelphia, 1999.

[9] C. A. S. Monroy and R. D. Christie, 'Energy storage effects on day-ahead operation of power systems with high wind penetration', in *Proceedings of North American Power Symposium, IEEE*, pp. 1–7, (2011).

[10] D. Nikovski and W. Zhang, 'Factored Markov decision process models for stochastic unit commitment', in *Proceedings of the IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*, pp. 28–35, (2010).

[11] T. Osogami, *Analysis of multiserver systems via dimensionality reduction of Markov chains*, Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, June 2005.

[12] N. P. Padhy, 'Unit commitment — A bibliographical survey', *IEEE Transactions on Power Systems*, **19**(2), 1196–1205, (2004).

[13] V. Pop, H. J. Bergveld, D. Danilov, P. P. L. Regtien, and P. H. L. Notten, *Battery Management Systems: Accurate State-of-Charge Indication for Battery-Powered Applications*, Springer, 2002.

[14] V. Pop, H. J. Bergveld, P. H. L. Notten, and P. P. L. Regtien, 'State-of-the-art of battery state-of-charge determination', *Measurement Science and Technology*, **16**(12), 93–110, (2005).

[15] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, 2005.

[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, second edn., 2003.

[17] O. Sundstroem and C. Binding, 'Optimization methods to plan the charging of electric vehicle fleets', in *Proceedings of the 1st International Conference on Control, Communication, and Power Engineering*, pp. 323–328, (2010).

[18] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramanian, 'Optimal power cost management using stored energy in data centers', in *Proceedings of the ACM SIGMETRICS 2011*, pp. 221–232, (2011).

[19] L. B. White, 'A new policy evaluation algorithm for Markov decision processes with quasi birth-death structure', *Stochastic Models*, **21**(2-3), 785–797, (2005).

[20] W. Zhang and D. Nikovski, 'State-space approximate dynamic programming for stochastic unit commitment', in *Proceedings of North American Power Symposium, IEEE*, pp. 1–7, (2011).